

Analýza elektronických obvodů programem OrCAD PSpice

Doc. Dr. Ing. Zdeněk Kolka

Obsah

1	ÚVOD	2
2	NÁVRHOVÝ SYSTÉM PSPICE	3
2.1	UKÁZKOVÝ PŘÍKLAD	3
2.2	MODULÁRNÍ STRUKTURA	3
2.3	ZÁKLADY JAZYKA (P)SPICE	5
3	EDITOR CAPTURE	9
3.1	PRÁCE S PROJEKTY	9
3.2	KRESLENÍ SCHÉMATU	11
3.2.1	<i>Editace</i>	11
3.2.2	<i>Nastavení parametrů součástek</i>	12
3.2.3	<i>Konektory</i>	14
3.2.4	<i>Zásady pro kreslení schémat</i>	14
4	POSTPROCEDURE PROBE	15
4.1	POPIS OVLÁDÁNÍ	15
4.1.1	<i>Spolupráce s editorem schématu</i>	15
4.1.2	<i>Práce s grafy</i>	15
4.2	ZPRACOVÁNÍ VÝSLEDKŮ	17
4.2.1	<i>Makra</i>	17
4.2.2	<i>Fourierova transformace</i>	17
4.2.3	<i>Hlášení o průběhu digitální simulace</i>	17
4.2.4	<i>Vícenásobná analýza (Performance Analysis)</i>	18
5	ZÁKLADNÍ METODY ANALÝZY	25
5.1	OBEČNÉ NASTAVENÍ	25
5.1.1	<i>Spuštění simulace a hledání případných chyb</i>	26
5.1.2	<i>Nastavení globálních podmínek simulace</i>	27
5.2	STEJNOSMĚRNÝ PRACOVNÍ BOD (BIAS POINT)	28
5.2.1	<i>Stejnosemřná přenosová funkce (.TF)</i>	28
5.2.2	<i>Citlivostní analýza</i>	29
5.3	STEJNOSMĚRNÉ ROZMÍTÁNÍ (DC SWEEP)	29
5.4	STRÍDAVÁ ANALÝZA (AC SWEEP)	32
5.4.1	<i>Lineární šumová analýza</i>	34
5.5	ANALÝZA V ČASOVÉ OBLASTI (TRANSIENT)	36
5.5.1	<i>Spektrální analýza (Fourier Analysis)</i>	37
5.6	DOPLŇKOVÉ ANALÝZY	38
5.6.1	<i>Krokování parametru (Parametric Sweep)</i>	38
5.6.2	<i>Teplotní analýza (Temperature Sweep)</i>	39
5.6.3	<i>Statistická a toleranční analýza (Monte Carlo/Worst Case)</i>	40
5.7	SIMULACE ČÍSLICOVÝCH A SMÍŠENÝCH OBVODŮ	48
5.7.1	<i>Základní principy</i>	48
5.7.2	<i>Číslíkové prvky a jejich parametry</i>	49
5.7.3	<i>Programování číslíkových zdrojů</i>	51
5.7.4	<i>Hlášení o chybových stavech během simulace</i>	55
5.7.5	<i>Analýza číslíkového obvodu v režimu Worst Case</i>	56

1 Úvod

Předmět Počítačové řešení elektronických obvodů (REO) je zařazen do třetího semestru bakalářského studia. Jeho cílem je poskytnout znalosti v oblasti počítačových metod řešení obvodů a modelování elektronických prvků. Simulace hraje velmi významnou roli při návrhu elektronických obvodů. Aby výsledky byly prakticky použitelné, musí uživatel znát principy a omezení různých metod pro řešení obvodů, stejně jako možnosti, které poskytují modely především polovodičových prvků. I přes značný vývoj v oblasti obvodové simulace v žádném případě nelze bez jakékoliv znalosti věci nakreslit obvod a stiskem tlačítka dostat výsledky. Je jenom na uživateli, aby zvolil adekvátní metodu řešení a adekvátní model pro danou situaci.

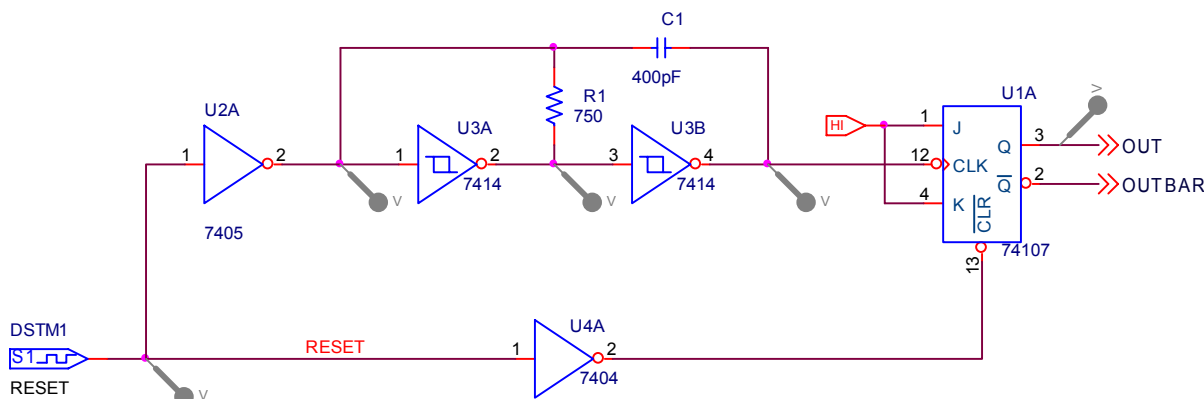
Kromě toho, že v rámci předmětu REO získá student rutinní znalosti ovládání programů, které bude využívat během celého dalšího studia, získá i přehled metod řešení a seznámí se s problematikou modelování elektronických prvků. Náplň navazuje na výuku základní teorie obvodů a některých partií elektronických prvků a numerické matematiky. Získané znalosti budou použitelné všude tam, kde se při řešení nebo návrhu obvodů používá počítačová simulace.

2 Návrhový systém PSpice

2.1 Ukázkový příklad

Součástí demonstrační i plně verze programu PSpice jsou ukázkové příklady. Mezi nimi je příklad oscilátoru pro smíšenou simulaci (obvod obsahuje analogové i číslicové prvky). Nachází se v adresáři *OrCAD\tools\pspice\capture_samples\mixsim\osc* nebo pod jménem *osc.opj*. Nejdříve spustíme editor schématu pomocí ikony **Capture**. Volbou **File/Open** otevřeme demonstrační příklad. Na levé straně pracovní plochy je okno projektu. Rozbalíme položky *\osc.dsn\Osc\Page1* a dvojitým poklepáním na **Page1** se otevře schéma. Všechny parametry součástek i nastavení analýz jsou již přichystány. Dvojitým kliknutím na součástku si můžeme prohlédnout okno pro nastavení jejích parametrů (žádné hodnoty neměnit!). Bude se provádět analýza v časové oblasti. Její nastavení se provádí volbou **PSpice/Edit Simulation Profile**. Ve schématu jsou speciální součástky - markery $\leftarrow \bigcirc$, které způsobí že se po ukončení simulace automaticky zobrazí průběh napětí nebo proudu v místech kde jsou připojeny.

Analýzu obvodu spustíme volbou **PSpice/Run (F11)**. Výsledky se automaticky zobrazí v postprocesoru **Probe**. Kdyby ve schématu nebyly markery, tak dostaneme pouze prázdný graf. Číslicové a analogové signály jsou v samostatných grafech. Pro prohlížení grafů je možné volbou **Trace/Cursor/Display** zobrazit dva kurzory, které se ovládají levým a pravým tlačítkem myši.



2.2 Modulární struktura

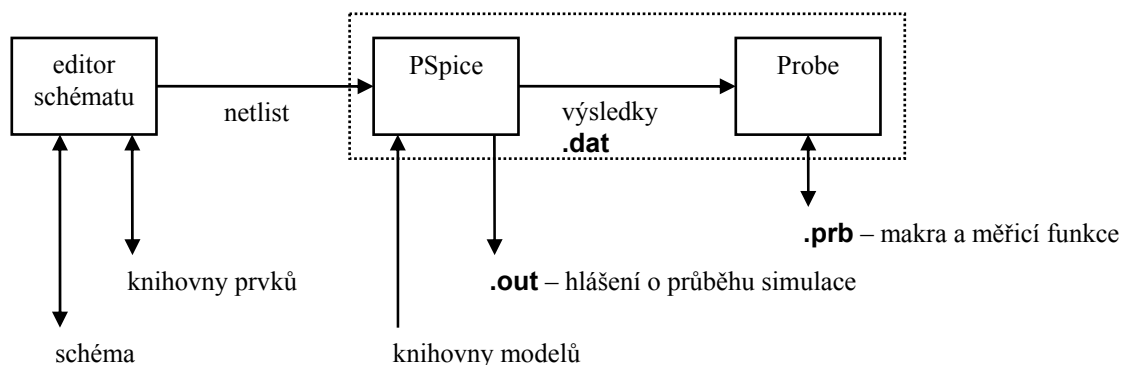
Jeden z nejúspěšnějších simulátorů pro PC - **PSpice** byl představen v roce 1984. Program byl původně určen pro DOS, od verze 5.4 pracuje pod MS Windows. PSpice nyní pokrývá celou oblast simulací elektronických obvodů, tj. obvody analogové, digitální i smíšené (mixed - mode).

PSpice je tvořen několika samostatnými programy, které tvoří kompaktní celek. Od verze 9 je program dodáván s novým editorem **Capture** vyvinutým firmou Orcad.

Editor schématu umožňuje vytvořit jak samostatně použitelné schéma, tak zejména připravit popis obvodu v jazyku Spice, tzv. netlist, pro následnou analýzu. Dovoluje vytvářet hierarchicky strukturované obvodové bloky a opakovaně je používat. Dodávány jsou knihovny pro více jak 10000 analogových a číslicových prvků. Pro zjednodušení práce s kompletním systémem je zaintegrovaný přechod do ostatních modulů jako: vyvolání editací modelů prvků, dialogový výběr typu analýzy a spuštění simulace. Tento modul tedy představuje výchozí ovládací článek pro celý systém. Součástí základního balíku programů pro simulaci obvodů jsou dále:

- PSpice** Modul pro simulaci smíšených (mixed - mode) obvodů, který dal jméno celému systému. Jeho základem byl program SPICE2 vyvinutý na univerzitě v Berkeley. Simulátor je od verze 9 integrovaný s postprocesorem **Probe** pro kompletní grafické vyhodnocení výsledků.
- Model Editor** Program pro vytváření modelů součástek z naměřených nebo katalogových údajů. Výstupem jsou knihovny modelů.
- Stimulus Editor** Program pro grafické vytváření analogových a číslicových zdrojů s komplikovaným časovým průběhem.
- Optimizer** Modul pro optimalizaci obvodů. Pracuje v interaktivním nebo automatickém režimu.
- Advanced Analysis** Modul nové generace pro provádění citlivostní, toleranční a zátěžové analýzy a optimalizace.

Organizace souborů a vzájemná komunikace mezi jednotlivými moduly je poněkud komplikovaná. Nicméně proces simulace zachovává standardně používané schéma, Obr. 2.1.



Obr. 2.1: Průběh simulace z hlediska výměny dat mezi moduly

V editoru schématu se kreslí schéma, definují parametry součástek a nastavují analýzy. Schematické značky jsou uloženy v knihovnách. Po spuštění simulace z prostředí editoru nejdříve proběhne kontrola schématu (ERC). Pokud nejsou chyby, tak se vygenerují soubory s popisem obvodu, které se předají simulátoru. Jedná se o textové soubory v jazyku PSpice (rozšíření původního Spice). Tyto soubory neobsahují modely součástek (např. tranzistorů). Ty jsou uloženy ve zvláštních knihovnách – textových souborech s příponou *.lib*.

Popis obvodu je načten simulátorem, který pak vyhledá příslušné modely v knihovnách (*.lib*). Při simulaci jsou generovány dva soubory. Textový soubor *<jméno>.out* obsahuje hlášení o průběhu simulace, chybách a výsledky některých analýz. Tento soubor lze otevřít v

editoru schématu nebo samozřejmě jakýmkoliv jiným textovým editorem. Výsledky analýz pro grafické zpracování jsou ukládány do binárního souboru <jméno>.dat. Po skončení simulace je automaticky spuštěn grafický postprocesor Probe, který zobrazí výsledky. Pokud při vyhodnocování grafů používáme makra a měřicí funkce, pak jsou tyto uloženy v souborech s příponou .prb. Simulátor i postprocesor je možné spustit samozřejmě ručně a příslušné soubory otevřít standardní volbou File/Open.

2.3 Základy jazyka (P)Spice

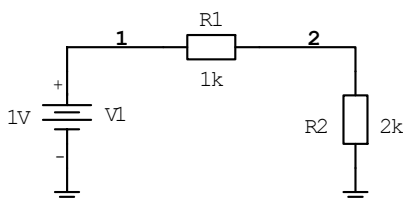
Syntaktická pravidla pro soubor popisující strukturu obvodu a požadované analýzy, často nazývaná jazyk, jsou založena na původním jazyku pro simulátor SPICE2. PSpice zavádí některé nové příkazy a vylepšení, samozřejmě je zachována zpětná kompatibilita Spice→PSpice. Vstupní soubor je přímým vstupem simulátoru PSpice. Je to textový soubor obsahující popis obvodu (propojení součástek a jejich parametry) a požadované analýzy. Samotný popis obvodu se nazývá „netlist“. Při práci v grafickém prostředí je generování vstupního souboru automatické. Existuje však výjimka, kdy jsme nuceni přímo pracovat s těmito soubory. Výrobci elektronických prvků totiž poskytují modely součástek z důvodů univerzality v textové podobě většinou v jazyce Spice. Uživatel by měl proto porozumět použití příkazu .MODEL a principu tvorby a volání makroobvodu (příkaz .SUBCKT).

Popis syntaxe některých příkazů jazyka PSpice je uveden v příloze. Automaticky generovaný vstupní soubor se skládá ze dvou hlavních částí:

- popisu obvodu (netlist),
- příkazové části.

První řádek souboru se neinterpretuje, slouží jako nadpis. Pokud řádek začíná „*“ tak je celý uvažován jako komentářový. Středník slouží k zápisu komentáře do příkazových řádků. Text od středníku do konce se neuvažuje. Jestliže se nějaký příkaz nevejde na jeden řádek, tak může pokračovat na dalším, ale prvním znakem pokračovacího řádku musí být „+“. V PSpice může být jméno uzlu jakýkoliv řetězec, v Spice pouze čísla. Uzel číslo 0 je vyhrazen jako referenční, každý obvod jej musí obsahovat. Malá a velká písmena se nerozlišují.

Příklad 2.1 *Výpočet pracovního bodu odporového děliče pomocí ručně vytvořeného vstupního souboru.*



Vstupní soubor má tvar:

Odporový dělič	první řádek slouží jako nadpis
* netlist	komentář na celý řádek
R1 1 2 1k ; rezistor R1 mezi uzly 1 a 2	použití komentáře se středníkem
R2 2 0 2k	
V1 1 0 DC 1	zdroj o napětí 1Vss

* definice analýz
 .OP
 .END

prázdné řádky se ignorují
 výpočet pracovního bodu
 soubor musí končit příkazem .END

Vytvoříme jej např. v editoru Notepad a uložíme pod jménem *delic.cir*. Spustíme PSpice (ikona PSpice) a volbou File/Open (maska Circuit Files *.cir) načteme soubor. Po skončení simulace jsou výsledky ve výstupním souboru *delic.out* (zkráceno):

```

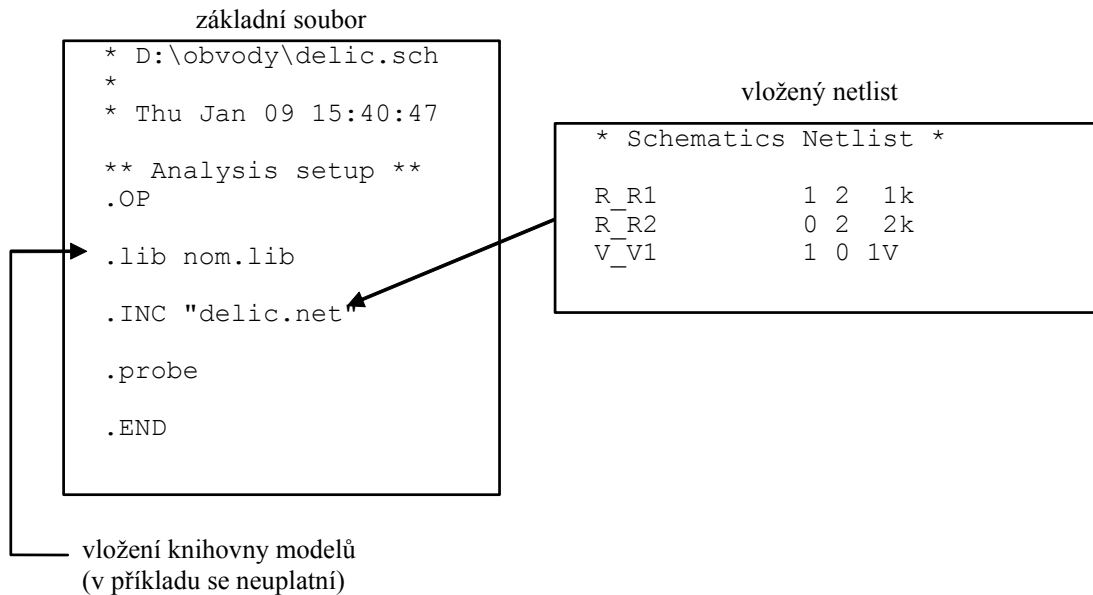
****      SMALL SIGNAL BIAS SOLUTION      TEMPERATURE = 27.000 DEG C
NODE      VOLTAGE      NODE      VOLTAGE      NODE      VOLTAGE      NODE      VOLTAGE
(  1)      1.0000      (  2)      .6667

VOLTAGE SOURCE CURRENTS
NAME      CURRENT
V1          -3.333E-04

TOTAL POWER DISSIPATION 3.33E-04 WATTS

```

Pro zajímavost uvedme vstupní soubor vytvořený automaticky editorem schématu pro stejný příklad:



Příkaz .MODEL

Odkaz na model je volitelnou nebo povinnou součástí definice součástky. Např. prvky R, C, L nemusí mít model, pokud vystačíme s definicí hodnoty. Jestliže ale požadujeme zadání teplotního koeficientu nebo tolerance je nutné model použít. U polovodičových prvků je model povinnou součástí definice.

Například:

R1 1 2 1k

definice rezistoru bez použití modelu

R2 3 4 Rmod 2k definice odkazem na model Rmod

Definice modelu v knihovně:

.MODEL Rmod RES (TCE = 0.1) nastavení teplotního koeficientu v modelu

Modely jsou uloženy v knihovnách (v PSpice mají příponu *.lib*). Vložení knihoven se provádí příkazem *.LIB*. Standardně definiční soubor obsahuje pouze odkaz na hlavní knihovnu *nom.lib* (příkaz *.LIB nom.lib* ve výpisu k příkladu), která teprve obsahuje odkazy na knihovny modelů, většinou strukturované podle výrobců. Pokud uživatel vytváří vlastní modely, doporučuje se vytvořit uživatelskou knihovnu, která se bude volat v *nom.lib* a nezasahovat do stávajících modelů.

Makroobvod (příkazy .SUBCKT, .ENDS)

PSpice může simulovat obvody, které mají hierarchickou strukturu. Tuto možnost dává příkaz *.SUBCKT*, který dovoluje formálně popsat dílčí blok obvodu, na který se pak můžeme vícenásobně odkazovat. Našli bychom zde analogii s použitím podprogramu ve vyšších programovacích jazycích. Ve formě makroobvodů jsou např. definovány operační zesilovače nebo logické prvky.

Definice makroobvodu:

.SUBCKT <name> [node]* [PARAMS: <<name> = <value>>]*]

..... *definice těla makroobvodu*

.ENDS

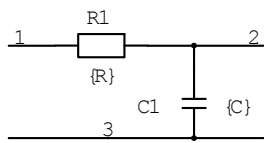
<name> jméno makroobvodu;
[node]* formální označení vývodů;
PARAMS: definice a přiřazení implicitních hodnot parametrů makroobvodu.

Uvnitř bloku se mohou použít všechny součástky (včetně volání jiných makroobvodů). V popisu obvodu mají jména (čísla) uzlů lokální charakter. Výjimkou je referenční uzel „0“ a uzly se jmény začínající na „\$G_“ (globální uzly). Rekurzní volání není dovoleno.

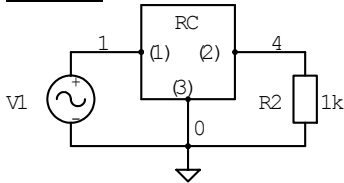
Volání makroobvodu:

X<name> [node]* <name> [PARAMS: <<name> = <value>>]*]

X<name> označení bloku začínající na X;
[node]* skutečné uzly;
PARAMS: definice parametrů. Pokud některý není uveden platí hodnota definovaná v deklaraci.

Příklad 2.2 *Definice a volání makroobvodu - článku RC s parametry R a C.*Definice:

```
.SUBCKT RCobv 1 2 3 PARAMS: R = 1k C = 1n
R1 1 2 {R} ; pokud je hodnota prvku určena
výrazem,
C1 2 3 {C} ; musí se použít složené závorky
.ENDS
```

Volání:

```
V1 1 0 AC 1 ; zdroj V1 pro AC analýzu
X1 1 2 0 RCobv PARAM: R = 2k
R2 2 4 1k
```

Jména vnějších uzlů jsou při volání nahrazena skutečnými jmény, pro vnitřní uzly se vytvoří jedinečná jména. Makroobvody (např. modely součástek od výrobců) mohou být uloženy v knihovnách *.lib* podobně jako modely polovodičů.

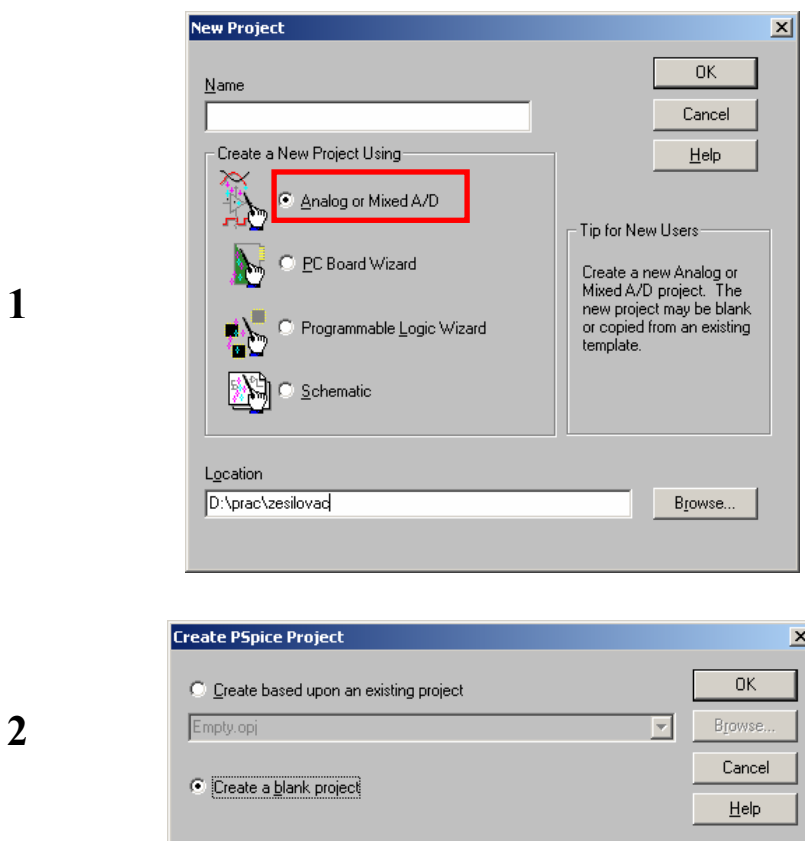
3 Editor Capture

U editoru Capture závisí nabídka hlavní lišty na tom, které okno pracovní plochy je aktivované.

V dalším textu budou popsány jen ty položky, které se bezprostředně dotýkají práce se simulátorem PSpice. Další možnosti (vytváření plošných spojů, syntéza číslicových obvodů) nalezne čtenář v elektronické referenční příručce, která je standardní součástí instalace.

3.1 Práce s projekty

Základem práce s editorem je tzv. projekt, který v sobě zahrnuje všechny soubory vztahující se k analyzovanému obvodu (schéma, simulační profily, lokální knihovny). Každému projektu by měl odpovídat samostatný adresář na disku. Po spuštění editoru se otevře prázdná pracovní plocha. Nový projekt vytvoříme volbou **File/New/Project**. Otevře dialogové okno s parametry nového projektu.

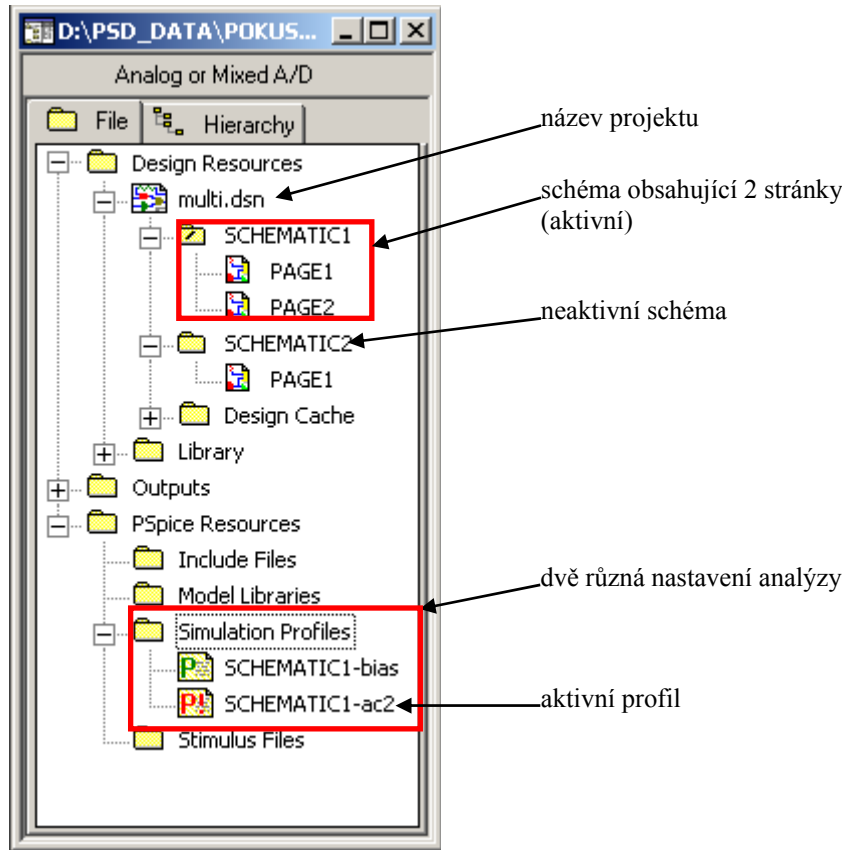


Obr. 3.1: Vytvoření nového projektu

V tomto okně se do pole **Name** vyplní jméno nového projektu. Jako typ zvolíme **Analog or Mixed A/D**. Do políčka označeného **Location** uvedeme adresář projektu. Výchozí hodnotou je zde adresář, ve kterém máme umístěné projekty, např. **D:\prac**. Do pole dopíšeme proto jen název adresáře samotného projektu (**zesilovac**), který bude automaticky vytvořen. Po stisku

OK zvolíme **Create a blank project**. Pokročilí uživatelé mohou vytvořit projekt na základě některé z přednastavených šablon. Po potvrzení se otevře prázdná stránka schématu.

Na pracovní ploše se objeví okno projektu, kde je možné spravovat jednotlivé soubory, které tvoří jeho součást, Obr. 3.2.



Obr. 3.2: Okno správce projektu

Tabulka 3.1: Příkazy pro správu projektu

přidání stránky do schématu	aktivovat jméno schématu + menu: Design/New Schematic Page nebo pravé tlačítko myši: New Page
přidání schématu do projektu	aktivovat jméno projektu+ menu: Design/New Schematic nebo pravé tlačítko myši: New Schematic
aktivace schématu	aktivovat jméno schématu + menu: Design/Make Root nebo pravé tlačítko myši: Make Root
aktivace profilu	aktivovat jméno profilu + menu: PSpice/Make Active nebo pravé tlačítko myši: Make Active

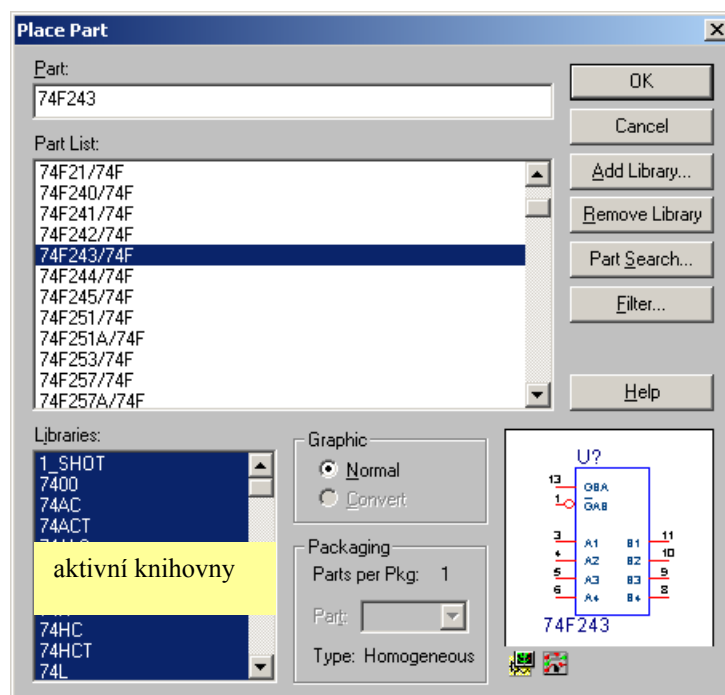
Každý projekt sestává z jednoho nebo více „schémat“ a každé schéma z jedné nebo více „stránek“. Schéma představuje samostatný elektrický obvod, jehož zapojení je nakresleno na více stránkách. Pro spojení uzlů mezi stránkami se používají *off-page* konektory (viz dále). V projektu může být aktivní vždy jen jedno schéma. To je pak označeno lomítkem vedle názvu (SCHEMATIC1 na Obr. 3.2). Pro nastavení aktivního schématu použijeme pravé tlačítko nad jeho jménem a volbu **Make Root**.

Nastavení požadované analýzy, tzv. simulační profil, se vždy vztahuje k aktivnímu schématu. V projektu můžeme definovat více profilů, Obr. 3.2. Aktivní profil lze nastavit volbou **Make Active**. Simulace je vždy prováděna s aktivním schématem a podle nastavení aktivního profilu. Neaktivní prvky nemají na výsledek vliv.

3.2 Kreslení schématu

3.2.1 Editace






Knihovnu prvků otevřeme nabídkou **Place/Part** (klávesa **P**), Obr. 3.3. Je vhodné aktivovat všechny knihovny. Tlačítkem **Add Library** otevřeme dialogové okno pro výběr knihoven (souborů *.olb*). Vybereme všechny (označíme první soubor a pak při stisknutí klávese **Shift** poslední soubor). Aktivní knihovny se pak objeví v poli **Libraries**. Tuto akci je nutné provést pouze jednou pro daný projekt. Do pole **Part** napíšeme jméno požadovaného prvku a stiskneme **Enter** nebo **OK**. Při psaní jména prvku automaticky probíhá hledání v seznamu. Je možné použít znak ***** a **?** podobně jako při práci se soubory. Např. **BC*** vyhledá všechny tranzistory, jejichž jméno začíná na **BC**. Obnovení seznamu se dosáhne zadáním znaku *****.



Obr. 3.3: Okno pro výběr prvku

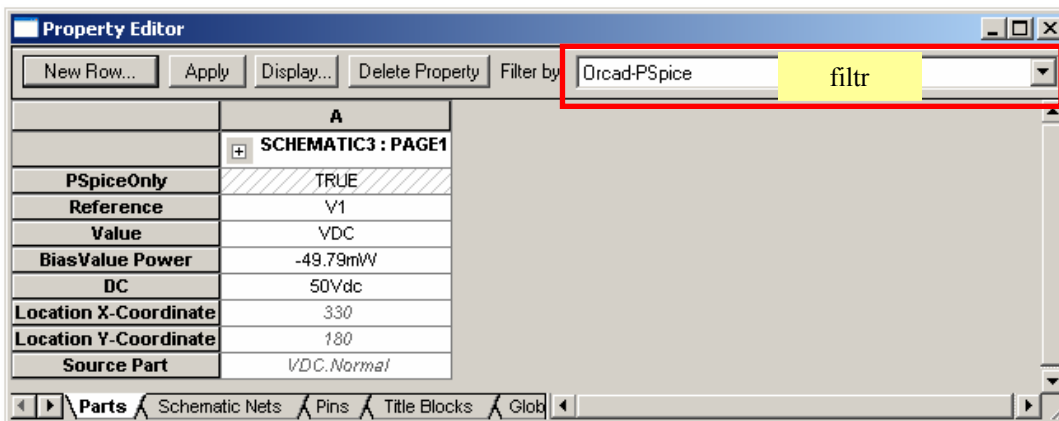
Následující tabulka uvádí základní příkazy pro kreslení schématu. Lišta s ikonami se standardně nachází na pravé straně okna programu.

Tabulka 3.2: Příkazy pro kreslení schématu

příkaz	menu	kl. zkratka	ikona
otevření knihovny	Place/Part	P	
kreslení vodiče	Place/Wire	W	
pojmenování vodiče (uzlu)	Place/Net Alias	N	
propojení křížujících se vodičů	Place/Junction	J	
rotace prvku	Edit/Rotate	R	
zrcadlení kolem svislé osy	Edit/Mirror/Horizontally	H	
zrcadlení kolem vodorovné osy	Edit/Mirror/Vertically	V	
vložení referenčního uzlu (vybrat prvek 0/SOURCE)	Place/Ground	G	

3.2.2 Nastavení parametrů součástek

Dvojitým kliknutím levého tlačítka na součástku se otevře okno pro zadávání parametrů, Obr. 3.4. Parametry jsou organizovány do tabulky (podobně jako u Excelu). Z důvodu zvýšení přehlednosti je vhodné pomocí pravého tlačítka myši nad tabulkou a volbou Pivot otočit tabulku tak, aby jednotlivé parametry tvořily řádky. Filtr nastavíme na hodnotu *Orcad-Pspice*. Tím se zobrazí jen parametry, které mají souvislost se simulací.



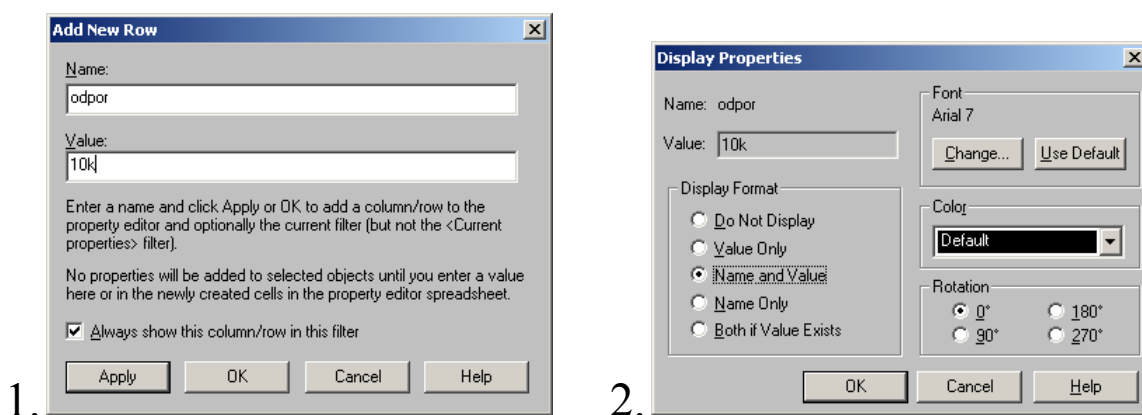
Obr. 3.4: Nastavení parametrů prvků

V tabulce je možné nastavovat hodnoty parametrů a eventuelně jejich viditelnost. Pokud je některý parametr viditelný ve schématu lze jeho hodnotu změnit přímo dvojitým kliknutím levého tlačítka. Pro zadání číselných hodnot je možné použít přípony (velikost písmen nerozhoduje). Za číslo resp. příponu je možné připsat označení fyzikální jednotky.

F	P	N	U	M	K	MEG	G	T
10^{-15}	10^{-12}	10^{-9}	10^{-6}	10^{-3}	10^3	10^6	10^9	10^{12}
femto	piko	nano	mikro	mili	kilo	mega	giga	tera

Příklad: 2k, 1.1uF, 1kHz, 1H

Kromě numerické hodnoty může být parametr určen výrazem uzavřeným do složených závorek { }. Povolené operátory a funkce jsou v příloze. S výjimkou funkčních zdrojů musí být hodnota výrazu známa před začátkem simulace. Není možné např. definovat odpor rezistoru závislý na napětí. Ve výrazu je možné použít parametr, který lze případně rozmítat nebo krokovat (kap. 5.6.1). Jeho deklarace se provádí pomocí pseudosoučástky PARAM, kterou umístíme do schématu (k ničemu se nepřipojuje). Otevřeme parametry prvku a pomocí tlačítka New Row nebo New Column (podle orientace tabulky) zadáme jméno proměnné a její hodnotu.

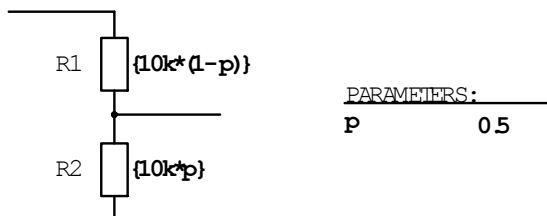


Obr. 3.5: Přidání globální proměnné do pseudosoučástky Param

Pomocí tlačítka Display v okně pro nastavování parametrů nastavíme Name and Value, aby se proměnná zobrazila přímo ve schématu.


Příklad 3.1 *Vytvoření modelu potenciometru pomocí parametru.*


Parametr p představuje relativní natočení potenciometru v intervalu (0, 1). Hodnota rezistoru R1 je $\{10k * p\}$, rezistoru R2 pak $\{10k * (1-p)\}$. Při deklaraci byl p nastaven na 0,5. Při případném rozmítání nesmí nikdy nastat případ, že hodnota některého rezistoru bude nulová.



3.2.3 Konektory

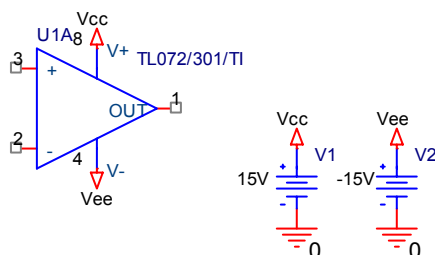
Konektory jsou speciální součástky umožňující „bezdrátové“ propojení. **Ke spojení dojde mezi konektory se stejným jménem.** Typickým příkladem použití je rozvod napájecího napětí u složitějších obvodů. Editor nabízí v paletě nástrojů tyto možnosti:

Place/Power (F)  konektory pro rozvod napájecího napětí. Vodiče označené konektory se stejným jménem se vzájemně propojí. Na zvoleném tvaru značky nezáleží, propojení se uskuteční na základě stejného jména.


Place/Ground (G)  zemní konektory. Mají stejnou funkci jako napájecí. Výjimku tvoří zemní konektor se jménem “0”, který představuje referenční uzel pro simulátor.

Place/Off-Page Connector konektory pro spojení mezi stránkami schématu.

Příklad 3.2 *Použití konektoru VCC_ARROW pro rozvod napájecího napětí operačních zesilovačů.*



3.2.4 Zásady pro kreslení schémat

- Všechny zdroje (napájecí i budící) se musí definovat ve schématu jakoby se jednalo o skutečné zapojení pro měření v laboratoři.
- PSpice požaduje, aby každý uzel měl definovanou stejnosměrnou cestu k referenčnímu uzlu 0. Z toho vyplývá, že alespoň jeden uzel obvodu musí být referenční, definovaný značkou  0 (důležité je jméno “0”).
- Volně ponechaný vývod součástky způsobí obvykle chybu.
- Častou příčinou chyb je několik součástek umístěných nad sebou tak, že se překrývají.
- Významné uzly obvodu (vstup, výstupy,...) bychom měli pojmenovat. V postprocesoru pak máme k dispozici napětí uzlu jako proměnnou $v(\text{jméno uzlu})$.

Součástí pracovní plochy je okno **Session Log**, kde lze hledat různá chybová hlášení při vytváření netlistu. Lze je zobrazit např. volbou **Window/Session Log**.

4 Postprocesor Probe

4.1 Popis ovládání

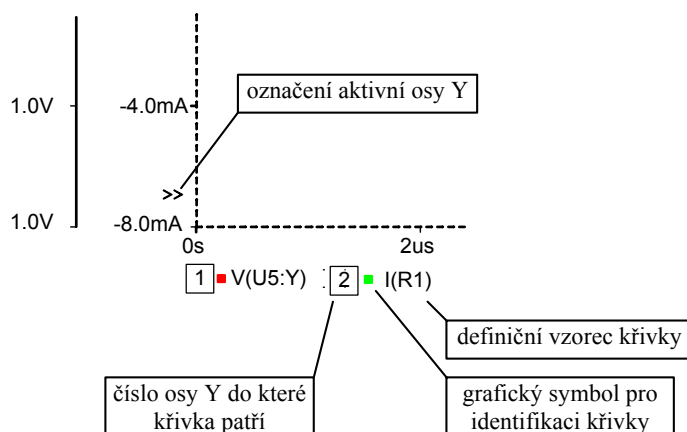
4.1.1 Spolupráce s editorem schématu

Pokud schéma obsahuje markery, tak po skončení simulace jsou příslušné veličiny. Nevýhodou je, že všechny průběhy mají společnou osu Y, tj. společné měřítko. Proto se nedoporučuje používat markery pro současné zobrazení veličin s řádově rozdílnými číselnými hodnotami. Pokud schéma neobsahuje markery, tak je po skončení simulace pouze načten datový soubor a zobrazí se prázdný graf. Při ručním spuštění Probe je data možné otevřít volbou File/Open a uzavřít File/Close. Volbou File/Append je možné současně načíst více datových souborů do jednoho grafu. Podmínkou je, že tyto soubory musí vzniknout simulací stejného obvodu.

Jestliže datový soubor obsahuje více analýz, tak při jeho otevření musíme zvolit typ analýzy. V průběhu práce jde totéž udělat volbou Plot/AC|DC|Transient.

4.1.2 Práce s grafy

Okno grafu je rozděleno na analogovou a digitální část. Pokud datový soubor neobsahuje číselná data, tak digitální část není přístupná a naopak. Každý graf může mít jednu až tři osy Y, přičemž každá má samostatné měřítko. V případě existence více os je aktivní označena symbolem „>>“, který lze přesouvat myší. Do této osy se přidává nová křivka volbou Trace/Add Trace (Insert).



Obr. 4.1: Levý spodní okraj grafu v Probe

Pod grafem je uveden seznam vzorců pro jednotlivé kreslené křivky. Vedle vzorce je barevný grafický symbol, v případě vícenásobné analýzy soubor symbolů pro každou dílčí křivku. Dvojitým poklepáním na barevný symbol (ne na vzorec) se zobrazí informace o křivce.

Zadávání a editace křivek

Volbou Trace/Add (Insert) se přidá nová křivka do aktivní osy. V okně pro zadávání křivek jsou zobrazeny proměnné (tj. napětí a proudy) a též dostupné funkce, které můžeme

použít ve výrazech. Do pole Trace Expression (resp. Trace Command u starších verzí) napíšeme výraz, který se má kreslit. Povolené funkce a operátory pro analogovou a digitální část jsou v příloze.

Pro editaci objektů slouží menu Edit. Křivku vybereme pomocí kliknutí levým tlačítkem na vzorec pod grafem, ostatní objekty kliknutím na ně.

Změna nastavení os a grafů

Standardně se otevře jeden graf, tj. jedna osa X . Volbou Plot/Add Plot to Window přidáme nový graf (novou osu X). Nově vytvořené grafy mají stejnou osu X jako předchozí. Pokud požadujeme jinou osu, tak musíme nejdříve volbou Plot/Unsynchronize X Axis povolit rozdílné osy X . Volbou Plot/Delete Plot smažeme aktivní graf, tj. ten, jehož některá osa Y je aktivní (symbol „>“). Novou osu Y přidáme volbou Plot/Add Y Axis. Tato osa je automaticky označena jako aktivní. Mazání aktivní osy se provádí volbou Plot/Delete Y Axis.

Volbou Plot/Axis Settings nebo dvojitým kliknutím na některou osu (vně grafu) se otevře okno pro nastavení parametrů rozdělené na záložky.

Nastavení osy X :

Data Range	Přepínání mezi automatickým a ručním nastavením rozsahu osy.
Scale	Přepínání mezi lineárním a logaritmickým měřítkem.
Use Data	Nastavení intervalu osy X pro integrační funkce: FFT, S(X), AVG(X), RMS(X), MIN(X) a MAX(X).
Processing Options	<input checked="" type="checkbox"/> Fourier - Provedení Fourierovy transformace dat grafu, <input checked="" type="checkbox"/> Performance Analysis - přepnutí vícenásobné analýzy, kap.4.2.4
Axis Variable	Nastavení proměnné osy X . Používá stejné dialogové okno jako při zadávání nové křivky (Trace/Add). Standardně je nastaveno: AC - frekvence, DC - rozmítaná veličina, Transient - čas.

Nastavení osy Y :

Y Axis Number	Volba čísla osy pokud jich je více.
Data Range	Přepínání mezi automatickým a ručním nastavením rozsahu osy.
Scale	Přepínání mezi lineárním a logaritmickým měřítkem.
Axis Title	Nastavení názvu osy (používá se pro tisk).

V položce Tools/Options jdou nastavit různé parametry Probe. Jejich popis je v nápovědě (F1) nebo v referenčním manuálu.

Kurzory

V grafech lze použít dva kurzory, které se zapínají volbou Tools/Cursor/Display (Ctrl-Shift-C). Kurzor číslo 1 se ovládá levým tlačítkem myši, číslo 2 pravým tlačítkem. V grafu se zobrazí tabulka udávající polohu obou kurzorů a rozdíl mezi nimi. Křivka, po které se kurzor pohybuje se volí pomocí kliknutí příslušného tlačítka na barevný symbol odpovídající křivce v levém spodním rohu nebo na název digitální křivky. Probe umožňuje hledat význačné body

na křivce. Příkaz pro hledání se vždy týká kurzoru, se kterým se naposledy pohnulo, hledání se provádí ve směru předchozího pohybu. Příkazy jsou přístupné pomocí položek menu Trace/Cursor nebo ikonami:

Peak	lokální maximum,
Trough	lokální minimum,
Slope	inflexní bod (maximální sklon křivky),
Min	globální minimum,
Max	globální maximum,
Point	přesun kurzoru na další datový bod,
Next/Previous Transition	- přesun kurzoru na následující číslicový přechod (událost).

4.2 Zpracování výsledků

4.2.1 Makra

Kromě standardních funkcí můžeme pro definici křivek použít vlastní funkce - makra. Volbou Trace/Macros otevřeme editor pro jejich definici.

Formát: $\langle name \rangle [(arg [,arg] *)] = \langle definition \rangle$

Příklad: $ADD(A, B) = A+B$

Volání: $ADD(v(1),v(2))$

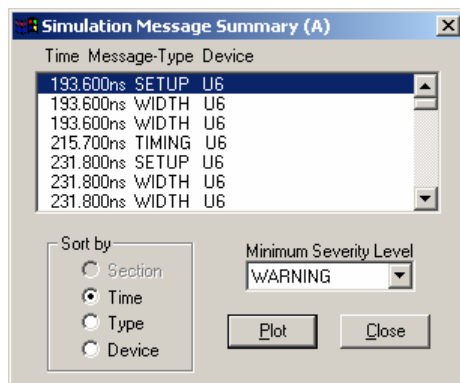
V definici můžeme použít standardní funkce a operátory a již existující makra. Rekurní definice není dovolena. Před vyhodnocením makra se provádí jen textová expanze.

4.2.2 Fourierova transformace

Zobrazení grafu lze přepnout do Fourierovy transformace nebo zpět pomocí volby Trace/Fourier. Pokud je na ose X čas, změní se na frekvenci a naopak. Před transformací dojde k převzorkování průběhu. Počet vzorků je roven nejbližší mocnině 2 od aktuálního počtu bodů, používá se kvadratická interpolace. Abychom dostali korektní výsledky pro periodické průběhy, je třeba pro transformaci použít celistvý násobek periody v ustáleném stavu. To lze dosáhnout omezením výstupu v nastavení časové analýzy (Analysis/Setup/Transient) nebo omezit interval pro transformaci v X Axis Settings/Data Range. Výsledkem FFT je komplexní spektrum, standardně je zobrazena amplituda. Pokud máme časový graf přepnutý do FFT pak můžeme použít funkce pro komplexní čísla, např. $R(v(1))$ zobrazí reálnou část spektra napětí V_1 .

4.2.3 Hlášení o průběhu digitální simulace

Volbou View/Simulation Messages jde zobrazit okno s hlášením o chybách vzniklých v průběhu simulace digitálního nebo smíšeného obvodu. Hlášení lze filtrovat podle úrovně důležitosti. Volba Plot zobrazí detail průběhu s popisem chyby.

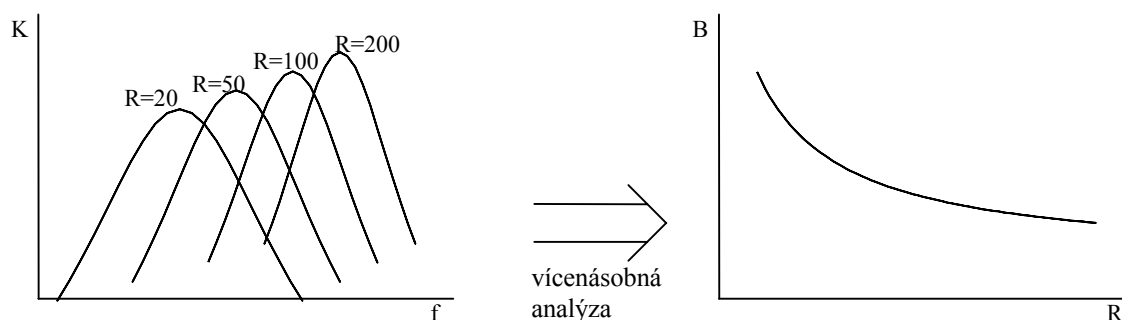


Obr. 4.2: Hlášení o chybách v číslicové části obvodu

4.2.4 Vícenásobná analýza (Performance Analysis)

Vícenásobná analýza je prostředek pro kvalitativní zpracování výsledků krokování parametru nebo analýzy Monte Carlo. Princip spočívá v odvození jediné hodnoty z každého dílčího běhu a zobrazení těchto hodnot do grafu v závislosti na krokovaném parametru. Pro analýzu Monte Carlo se zobrazí histogram.

Předpokládejme například, že je krokován parametr nějakého prvku pásmové propusti (rezistor R). Mění se jak šířka pásma, tak střední frekvence.



Obr. 4.3: Princip vícenásobné analýzy

Pomocí vícenásobné analýzy můžeme zobrazit závislost šířky pásma na krokovaném rezistoru R . Základem je **měřicí funkce** (Measurement), jejímž vstupem je celý průběh (křivka) a výstupem jediná hodnota, v našem případě šířka pásma.

Vícenásobnou analýzu zvolíme pomocí Trace/Performance Analysis. Zobrazí se prázdný graf nebo histogram (pro Monte Carlo), kde musíme pomocí Trace/Add (Insert) zadat příslušnou měřicí funkci. Druhou možností představuje spuštění „průvodce“ volbou Trace/Performance Analysis/Wizard, který uživatele provede celým nastavením. Jednotlivé kroky jsou bohatě okomentovány.



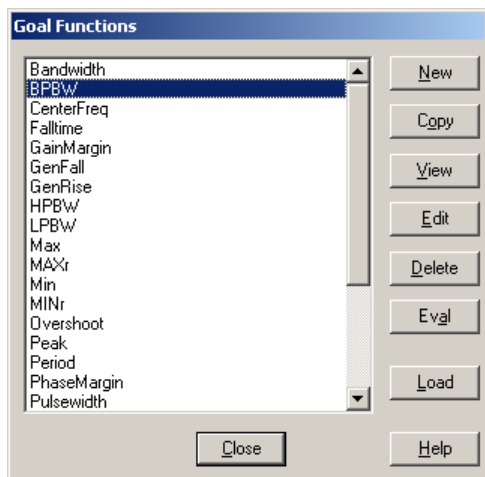
Pokud je použita jednobodová analýza jako vnitřní smyčka krokování, pak se parametrická analýza spustí automaticky a není nutné zadávat měřicí funkce protože výsledkem dílčího běhu je jenom jeden bod.

Definice měřicích funkcí

Textová definice měřicích funkcí může být uložena na třech místech:

1. Globální soubor (standardně *pspice.prb*, jméno je určeno položkou **PRBFILE** v souboru *pspice.ini*), který je přístupný pro všechna schémata. Obsahuje řadu předdeklarovaných maker a funkcí včetně jejich popisu. Uživatel by měl přednostně ukládat svoje funkce do tohoto souboru.
2. Lokální soubor *<jméno>.prb* který soubor obsahuje nastavení Probe pro daný datový soubor (tj. schéma) a lokální definice maker a měřicích funkcí.
3. Uživatelský soubor. Při opětovném spuštění Probe není tento soubor automaticky načten. Uživatel musí použít volbu **Trace/Measurements/Load**. Doporučuje se používat příponu *.prb* pro jméno souboru.

Prostředek ke správě měřicích funkcí otevřeme volbou **Trace/Measurements**. V seznamu jsou uvedené měřicí funkce načtené z globálního a lokálního souboru.



Obr. 4.4: Editor měřicích funkcí

New Definice nové měřicí funkce. Po zvolení této položky musíme určit soubor kam bude funkce uložena a otevře se textový editor, kde doplníme definici těla funkce a návratové hodnoty. Při uzavírání editoru se kontroluje správnost definice. Popis syntaxe je uveden dále v této kapitole.

Copy Kopíruje zvolenou měřicí funkci pod novým jménem do jiného souboru.

View Zobrazí definici zvolené měřicí funkce.

Edit Editování již existující funkce.

Delete Odstranění funkce.

Eval Nástroj pro ověření činnosti zvolené vyhledávací funkce. Musíme zadat argumenty funkce a Probe zobrazí nalezené body a návratovou hodnotu. Pokud je zobrazen svazek křivek, tak se vyhodnocení provádí pouze na první. O případných chybách během vyhodnocení je podána informace.

Load Načtení definic měřicích funkcí z uživatelského souboru. Globální a lokální soubor je načítán automaticky.



Je důležité se vždy přesvědčit, že měřicí funkce pracuje správně, tj. zkoumaná křivka má průběh, se kterým se počítalo při vytváření funkce. K tomu účelu je nejlépe použít volbu **Trace/Measurements/Eval**.

Zobrazení výsledku měřicích funkcí pro jednu křivku

Volbou **Trace/Evaluate Measurement** máme možnost zobrazit výsledek měřicí funkce pod oknem grafu.

Syntaxe měřicích funkcí

Měřicí funkce vrací hodnotu výrazu získaného z funkčních hodnot grafu v bodech, které byly nalezeny a označeny pomocí vyhledávacích příkazů v těle měřicí funkce. Tyto body jsou označeny jako 1, 2,, Příslušné x -ové souřadnice jsou x_1, x_2, \dots a y -ové souřadnice jsou y_1, y_2, \dots

```
goal_function_name(1, 2, ..., n, subarg1, subarg2, ..., subargm) = marked_point_expression
{
  1| search_commands_and_marked_points_for_expression_1;
  2| search_commands_and_marked_points_for_expression_2;
  .
  n| search_commands_and_marked_points_for_expression_n;
}
```

(1, 2, ..., n) Tyto argumenty jsou při volání funkce nahrazeny křivkami (tj. výrazy z proměnných v obvodu) na které se bude aplikovat vyhledávání. Každé číslo 1, 2, ... n odpovídá skupině vyhledávacích a značkovacích příkazů označených pomocí návěští 1|, 2|, atd. v těle funkce.

(subarg1, subarg2, ..., subargm) - Volitelné argumenty pomocí kterých můžeme v měřicí funkci použít uživatelem specifikované konstanty.

marked_point_expression - Výraz pro výpočet návratové hodnoty měřicí funkce. Má stejný formát jako standardní výrazy v Probe s následujícími výjimkami:

1. Označené body (x_1, y_3, \dots) jsou použity jako proměnné namísto napětí a proudů.
2. Je k dispozici nová funkce MPAVG která počítá průměrnou hodnotu z mezi dvěma označenými body. $MPAVG(x_1, x_2[,fraction])$
[fraction] - relativní centrovaná část intervalu mezi x_1 a x_2 .
3. Vícebodové funkce d(), s(), AVG(), RMS(), MIN() a MAX() nejsou povoleny.
4. Komplexní funkce M(), P(), R(), IMG() a GD() nejsou povoleny. Pokud je argument funkce komplexní číslo, automaticky se uvažuje jeho modul.

search_commands_and_marked_points_for_expression_n - Příkazy pro vyhledání a označení bodů na zkoumané křivce. Jsou rozděleny do sekcí 1|, 2|, atd. podle křivky. Za posledním příkazem sekce musí být středník, uvnitř ne.

Formát vyhledávacího příkazu:

search [direction] [/start_point/] [#consecutive_points#] [(range x [,range y])] [for] [repeat:]<condition>

[direction] Určení směru kterým se bude hledat. Možné směry jsou: FORWARD nebo BACKWARD. Standardní hodnota je FORWARD.

[/start_point/] Bod ze kterého se začíná hledat. Pokud se neuvede tak při prvním hledání se začíná od počátku a další hledání začíná od posledně nalezeného bodu. Možnosti:

- ^ nebo BEGIN počáteční bod prohledávaného intervalu,
- \$ nebo END koncový bod prohledávaného intervalu,

xn výraz z již označených bodů (např. $(x1+x2)/2$).

[#consecutive_points#] Počet následných bodů aby byla splněna vyhledávací podmínka. Význam závisí na konkrétní podmínce. Např. pokud se hledá maximum tak tento parametr udává počet bodů napravo a nalevo které musí mít menší hodnotu než bod maxima. Standardní hodnota je 1.

[(range x [,range y])] Specifikace intervalu na ose X a Y ve kterém se bude provádět hledání (pokud se neuvede tak se prohledává plný rozsah).

možnosti:

„nic“ pro plný rozsah	(, 10), (, , , 1)
numerické hodnoty	(1, 2, 0, 10.4)
procenta plného rozsahu	(1, 50%)
výraz ze značených bodů	(x1+x2, 90%)

[repeat:] Specifikace počtu splnění <condition> pro nalezení bodu (např. 3:).

<condition> Podmínka pro nalezení bodu. Pokud není nalezen žádný vyhovující bod, tak měřicí funkce nevrátí žádnou hodnotu a je ohlášena chyba.

LEvel<(value[,posneg])> - Nalezne první datový bod, jehož souřadnice y překročí <value>. Provádí interpolaci. Uplatní se #consecutive_points#.

<value>:

- číslo
- procenta plného rozsahu osy Y
- výraz ze značených bodů
- hodnota v decibelech vzhledem k maximum nebo minimum
 - max-3 db nebo -3 db => 3 db pod max. hodnotou
 - min+3 db nebo 3 db => 3 db nad min. hodnotou
- hodnota vzhledem k maximum nebo minimum
 - max-3 => maximum -3
 - min+3 => minimum +3
- hodnota v decibelech vzhledem k posledně nelezenému bodu
 - .-3 db=> 3 db pod souřadnicí y předchozího bodu
 - +.3 db => 3 db nad souřadnicí y předchozího bodu
- hodnota vzhledem k posledně nelezenému bodu
 - .-3 => předchozí bod -3
 - +.3 => předchozí bod +3

“.” představuje souřadnici y posledního nalezeného bodu.

[,(posneg)] Určení směru kterým křivka protne definovanou úroveň. Kladný směr = P, záporný = N, oba = (B). Standardně je nastaveno B. Směr se uvažuje relativně vzhledem ke směru prohledávání.

SLope[(posneg)] - Najde bod s největší směrnicí v zadaném směru. Provádí interpolaci. Uplatní se #consecutive_points#.

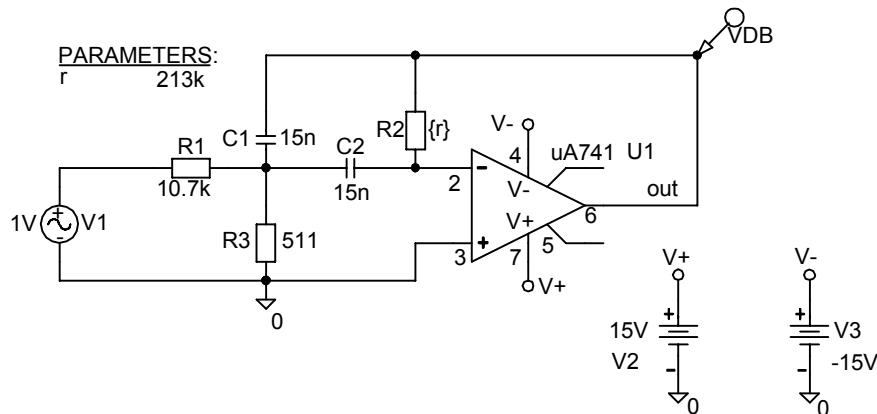
[(*posneg*)] Určení požadovaného znaménka derivace. Kladné = P, záporné = N, obě = (B). Standardně je nastaveno P.

- PEak** Najde nejbližší lokální maximum v daném směru. Uplatní se *#consecutive_points#*.
- TRough** Najde nejbližší lokální minimum v daném směru.
- MAx** Najde bod s maximální hodnotou souřadnice y na celém intervalu.
- Min** Najde bod s minimální hodnotou souřadnice y na celém intervalu.
- POint** Přesune se na další datový bod v daném směru.
- XValue**<(value)>- Najde první bod jehož souřadnice x je rovna specifikované hodnotě. <(value)> - stejná definice jak v příkazu LLevel.

Označení nalezeného bodu:

Příkazem *!n* se označí bod. Číslo *n* je libovolné a slouží pouze k identifikaci bodu. Označení může být umístěno za jakýmkoliv kompletním vyhledávacím příkazem.

Příklad 4.1 *Analýza aktivní pásmové propusti s operačním zesilovačem.*



A) Určení závislosti činitele jakosti Q na hodnotě rezistoru $R_2 \in [100\text{k}\Omega, 300\text{k}\Omega]$.

Závislost Q na R_2 určíme pomocí parametrické analýzy. Základem bude analýza AC v rozsahu 500Hz až 3kHz s krokováním R_2 v rozsahu 100k Ω až 300k Ω .

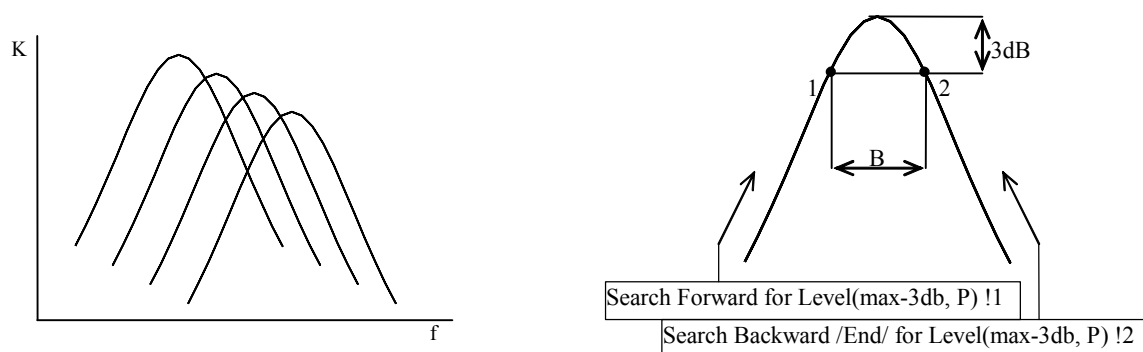
AC Sweep: •Decade Pts/Decade: 200
Start Freq.: 500
End Freq.: 3k

Parametric: • Global Parametr Name: r
• Linear Start Value: 100k
End Value: 300k
Increment: 10k

Výstupní uzel filtru označíme jako *out* a navíc na něj umístíme decibelový marker (PSpice/Markers/Advanced/). Napětí vstupního zdroje V_1 (VAC) je 1V, tedy přenosová funkce K je číselně rovna napětí na výstupu $v(out)$.

Před vytvářením měřicí funkce je nutné zhodnotit průběh všech dílčích křivek abychom měli zaručeno že navržené vyhledávací příkazy vždy najdou správný bod. Příkazem

PSpice/Run (F11) spustíme simulaci. Výsledek v Probe je zhruba zachycen na obrázku. K výpočtu Q budeme potřebovat určit šířku pásma pro pokles o 3dB. Frekvenční rozmítání musí být nastavené tak, aby všechny křivky s rezervou umožnily určit pokles o 3dB.



Obr. 4.5: Konstrukce měřicí funkce

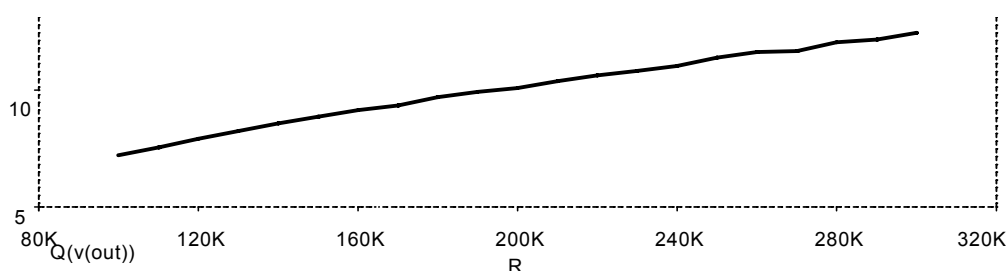
Pomocí vyhledávacích příkazů najdeme body 1 a 2 ve kterých přenos poklesne o 3dB oproti maximu. Hledání bodu č. 1 začíná od počátku, bod č. 2 je hledán od konce intervalu. V příkazech Level by nemusela být specifikace směru průsečíku (P - *posneg*). Je to jistá pojistka pro případ že prohledávaná křivka nebude mít předpokládaný tvar. V tom případě je pravděpodobnější vznik chyby. Pro určení Q použijeme přibližný vzorec $Q = \frac{f_0}{B}$, $f_0 = \sqrt{f_1 f_2}$.

Při výpočtu návratové hodnoty musíme samozřejmě místo f_1 a f_2 použít x_1 a x_2 . Měřicí funkce Q bude mít konečný tvar:

```
Q(1)=SQRT(x1*x2)/(x2-x1)
{
  1| Search Forward for Level(max-3db,P) !1
  Search Backward /End/ for Level(max-3db,P) !2;
}
```

Parametr 1 se při volání nahradí přenosovou funkcí. Protože $V_1 = 1V$ stačí uvést výstupní napětí $v(out)$. Jestliže je argument měřicí funkce komplexní automaticky se uvnitř převádí na absolutní hodnotu. Přesnost výpočtu Q závisí na správném určení maxima přenosu, proto je v AC analýze použito 200 bodů na dekádu.

Měřicí funkci zapíšeme do lokálního nebo globálního souboru v Probe volbou Trace/Measurements/New. Ověření správného fungování lze provést v Trace/Measurements/Eval. Jako argument zadáme $v(out)$, tj. přenosovou funkci. Srovnávací analýzu zapneme volbou Trace/Performance Analysis/OK a kreslení měřicí funkce pomocí Trace/Add (Trace Expression = $Q(v(out))$) nebo můžeme použít průvodce (Trace/Measurements/Wizard).



Obr. 4.6: Závislost Q na parametru R

Jestliže přidáme další osu do srovnávacího grafu (volbou Ctrl-Y) můžeme zobrazit např. frekvenci f_0 pásmové propusti pomocí předdeklarované měřicí funkce `CenterFreq(db(v(out)),3)`. První argument je modul přenosu v decibelech, parametr 3 udává, že se f_0 bude určovat jako průměr frekvencí dvou bodů pro pokles o 3dB (podobně jako u Q).

B) Určení tolerance f_0 pásmové propusti jestliže tolerance rezistorů je 1% a kapacitorů 5%.

Rezistor R2 má nominální hodnotu 213k Ω . Předpokládáme, že se parametry operačního zesilovače nemění.

Princip bude podobný jako v předchozí části příkladu. Použijeme předdeklarovanou měřicí funkci `CenterFreq` (doporučuji si prohlédnout její definici). Nastavení základní analýzy AC zůstane zachováno. Místo Parametric použijeme Monte Carlo pro 400 běhů. Rozložení pravděpodobnosti bude rovnoměrné.

AC Sweep: •Decade	Pts/Decade:	200
	Start Freq.:	500
	End Freq.:	3k

Monte Carlo:	Analysis:	•Monte Carlo	Numebr of runs:	100
	Output Var:	V(out)	Use distribution:	Uniform
	Function:	YMAX		
	Save data from All runs			

Abychom omezili velikost datového souboru zvolíme ukládání výstupu jen pro uzel s markerem, tj. pro výstup (Pspice/Edit Simulation Profile/Data Collection -At Markers Only). Po ukončení simulace se v Probe zobrazí přenosové funkce pro dílčí běhy analýzy.

Vícenásobnou analýzu zapneme volbou Trace/Performance Analysis/OK a kreslení měřicí funkce pomocí Trace/Add (Trace Expression = `CenterFreq(db(v(out)),3)`) nebo můžeme použít průvodce. První argument měřicí funkce je modul přenosu v decibelech, parametr 3 udává, že se f_0 bude určovat jako průměr frekvencí dvou bodů pro pokles o 3dB. Histogram je vytvořen automaticky, protože Probe pozná, že se jedná o zpracování výsledků Monte Carlo. Pod histogramem čteme:

minimum = 980Hz, maximum = 1080Hz.

Je možné očekávat, že se f_0 bude nacházet v tomto intervalu. Při zvýšení počtu běhů dostaneme velmi podobné mezní hodnoty 980Hz a 1080Hz. Další zvýšení již nepřinese podstatné změny.

5 Základní metody analýzy

5.1 Obecné nastavení

Simulační profil

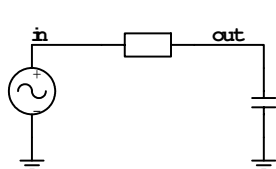
Nastavení prováděných analýz je uloženo v simulačním profilu. Každé schéma v projektu může mít asociován libovolný počet profilů, jen jeden však může být aktivní. Volba aktivního profilu se provádí v okně projektu pomocí pravého tlačítka myši.

Nový profil vytvoříme příkazem PSpice/New Simulation Profile. V profilu může být nastavena jedna ze čtyř základních analýz (pracovní bod, stejnosměrná, střídavá, časová) a případné nadstavby (krokování, Monte Carlo, atd.). Otevření profilu se provede příkazem PSpice/Edit Simulation Profile.

Jména uzlů

Každý uzel simulovaného schématu má jméno, které se používá např. při odkazu na jeho napětí. Standardně jsou jména přidělována automaticky podle šablony N0001, N0002, atd. Nevýhodou tohoto systému je, že nejsme schopni určit vazbu mezi uzly ve schématu a jejich jmény. Je proto výhodné označit sledované uzly uživatelským jménem (label) pomocí příkazu Place/Net Alias (N).

Příklad 5.1 *Označení uzlů ve schématu.*



Napětí v uzlech je $v(in)$ a $v(out)$. Např. modul napět'ového přenosu v decibelech bude $db(v(out)/v(in))$.



Markery

Markery jsou speciální součástky, které způsobí, že se po skončení simulace ihned vykreslí požadovaná veličina v Probe. Při manipulaci s markery není třeba vždy znovu spouštět simulaci. Markery se nacházejí v položce Pspice/Markers hlavního menu.

Typy markerů:

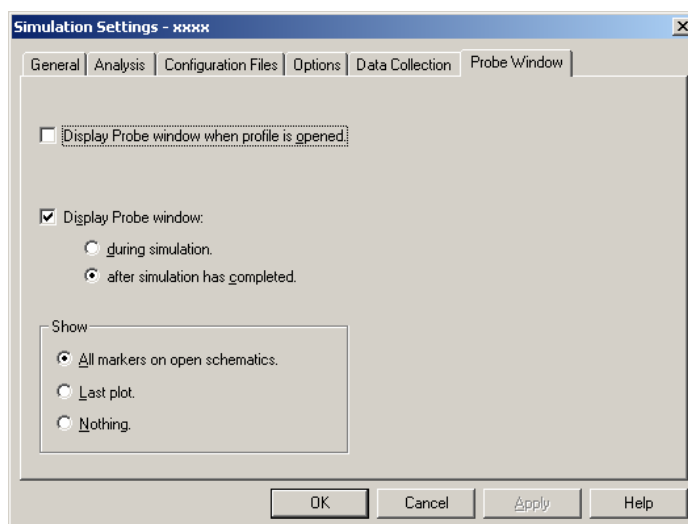
- Voltage Level - zobrazení napětí proti referenčnímu uzlu (AGND nebo EGND).
- Voltage Differential - dvojice markerů (+ a -) pro určení napětí mezi uzly.
- Current Into Pin - měření proudu vývodem součástky.
- Power Dissipation – výkon ztracený v daném prvku.
- Mark Advanced - speciální markery převážně pro analýzu AC.

Zobrazení stejnosměrného pracovního bodu

Proudy a napětí pracovního bodu je možné zobrazit ve schématu pomocí ikon  nebo , anebo volbou Pspice/Bias Points/Enable.

Nastavení postprocesoru Probe

V profilu v záložce Probe Window je možné nastavit chování postprocesoru.



Obr. 5.1: Nastavení postprocesoru Probe

Display Probe Window - Automatické spuštění Probe v průběhu nebo až po ukončení simulace. Při simulaci rozsáhlých systémů je možné spustit postprocesor během výpočtu a monitorovat průběhy. V případě chyby je možné simulaci ukončit.

Show Umožňuje nastavit, co se zobrazí v grafech. Pokud používáme markery, tak musí být aktivní první volba (**All markers...**). Dodatečně definované grafy v Probe budou však při nové simulaci smazány. Výhodnější je proto volba **Last plot**, která zachovává poslední nastavení. Při prvním spuštění však musí uživatel definovat křivky ručně.

V záložce Data Collection je možné nastavit, co všechno se bude ukládat do výstupního souboru. Standardně se ukládají veškeré obvodové veličiny (proudy, napětí, výkony). Selektivní ukládání má smysl jen u velkých obvodů např. při analýze Monte Carlo, kdy by vznikl velmi rozsáhlý výstupní soubor.

5.1.1 Spuštění simulace a hledání případných chyb

Simulace se spouští volbou PSpice/Run (F11). Nejdříve proběhne kontrola schématu (ERC - Electrical Rules Check). Pokud nejsou patřičně nastaveny parametry součástek, jména konektorů apod., zobrazí se okno, kde jsou hlášení o nalezených chybách. Příčinu chyby lze vyčíst z popisu. Pokud ERC proběhne bez chyb, tak se vytvoří soubory s popisem obvodu (netlist) a předají se simulátoru. Jestliže během simulace dojde k chybě, tak se ve stavovém okně simulátoru objeví chybové hlášení. Soubor s hlášením o průběhu simulace a chybách jde

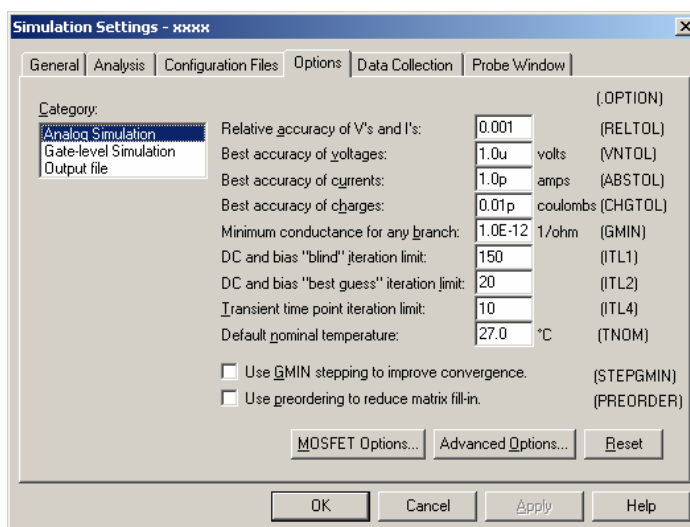
otevřít volbou PSpice/View Output File. Pokud bylo příčinou chyby špatné zadání obvodu (nejčastěji špatné parametry součástek nebo analýz), tak je pod příslušným řádkem nápis ERROR a stručný popis chyby. Uživatel musí rozhodnout o způsobu opravy. Pokud došlo k chybě až během simulace (např. řešení nekonverguje), tak je hlášení až na konci souboru.

Nejčastější příčiny chyb:

- Neuzemněný obvod - ve schématu chybí definice referenčního (zemního) uzlu 0, PSpice hlásí plovoucí (floating) uzly.
- Mezi číselnou hodnotou a příponou je mezera
chybně: 1 kHz ,
správně: 1kHz .
- Logaritmický interval rozmítání obsahuje nulu.
- Hodnota R, L nebo C je nulová - to PSpice nepřipouští.
- Chybějící deklarace globálního parametru (pseudosoučástka PARAM).

5.1.2 Nastavení globálních podmínek simulace

V záložce Options simulačního profilu se nastavují parametry simulace, které určují chování programu PSpice. Výchozí hodnoty jsou nastaveny tak, aby vyhovovaly pro simulaci za běžných podmínek.



Obr. 5.2: Okno pro nastavení analýz

Kompletní popis parametrů je uveden v referenční příručce. V dalším výkladu se omezíme pouze na některé hodnotové parametry.

ABSTOL = 1pA - přesnost výpočtu proudů,
VNTOL = 1μV - přesnost výpočtu napětí,
RELTOL = 0.001 - relativní přesnost výpočtu napětí a proudů.

Tyto tři parametry řídí přesnost výpočtu. Relativní přesnost je určena parametrem RELTOL. Absolutní přesnost však nikdy neklesne pod hodnoty VNTOL resp. ABSTOL.

$$\Delta V = \max(VNTOL, V \cdot RELTOL)$$

$$\Delta I = \max(\text{ABSTOL}, I \cdot \text{RELTOL})$$

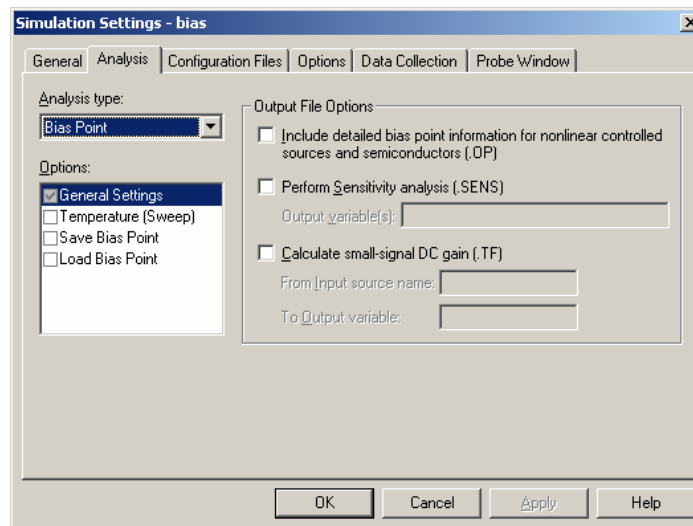
Při simulaci silnoproudých obvodů (řádově kV a kA) se doporučuje zvýšit hodnoty ABSTOL a VNTOL o tři řády.

TNOM = 27°C - nastavení standardní teploty pokud není zvoleno její rozmítání,

GMIN = 1pS - minimální vodivost větve. Uplatní se hlavně pro polovodičové prvky.

5.2 Stejnoseměrný pracovní bod (Bias Point)

Výpočet pracovního bodu je doplněn o citlivostní analýzu a malosignálový stejnosměrný přenos. Výsledky analýzy se ukládají do výstupního textového souboru (PSpice/View Output File) a údaje o pracovním bodu lze zobrazit ve schématu.



Obr. 5.3: Výpočet stejnosměrného pracovního bodu

Include detailed... – detailní výpis malosignálových parametrů nelineárních prvků do textového výstupního souboru.

Sensitivity analysis – citlivostní analýza (viz níže).

Small Signal DC gain - výpočet stejnosměrného přenosu, vstupního a výstupního odporu v obvodu linearizovaném v pracovním bodu.

5.2.1 Stejnoseměrná přenosová funkce (.TF)

Výpočet stejnosměrného malosignálového přenosu, vstupního a výstupního odporu v obvodu linearizovaném v pracovním bodu. Musíme specifikovat jméno budícího zdroje a výstupní veličinu. Pokud je výstupem proud musí se jednat o proud přes napěťový zdroj.

Například pro určení $\beta_d = \frac{\Delta I_C}{\Delta I_B}$ z příkladu v kap. 5.3 bychom nastavili:

Output Variable: I(V1) (pozor na kladný směr proudu - přenosová funkce bude mít opačné znaménko)
 Input Source: I1

Popis specifikace výstupních proměnných je v příloze. Výpočet se provádí v pracovním bodu, takže všechny zdroje musí mít správně nastavené parametry DC. Výstup analýzy je uložen do textového souboru *.out*, který lze otevřít volbou PSpice/View Output File.

<pre> **** SMALL-SIGNAL CHARACTERISTICS I(V_V1)/I_I1 = -1.784E+02 ← musíme otočit znaménko INPUT RESISTANCE AT I_I1 = 2.862E+03 OUTPUT RESISTANCE AT I(V_V1) = 4.816E+04 </pre>

5.2.2 Citlivostní analýza

Výpočet citlivosti stejnosměrného pracovního bodu na parametry rezistorů (R), nezávislých zdrojů (V a I), spínačů (S a W), diod (D) a bipolárních tranzistorů (Q). V nastavovacím okně specifikujeme seznam sledovaných veličin oddělených mezerou. Proud se může specifikovat pouze přes napěťový zdroj. Popis syntaxe je v příloze. Výsledky analýzy jsou ukládány do výstupního textového souboru *.out*, který lze otevřít volbou PSpice/View Output File.

Výsledky jsou uspořádány do čtyř sloupců:

Element name Jméno parametru na který se zjišťuje citlivost.

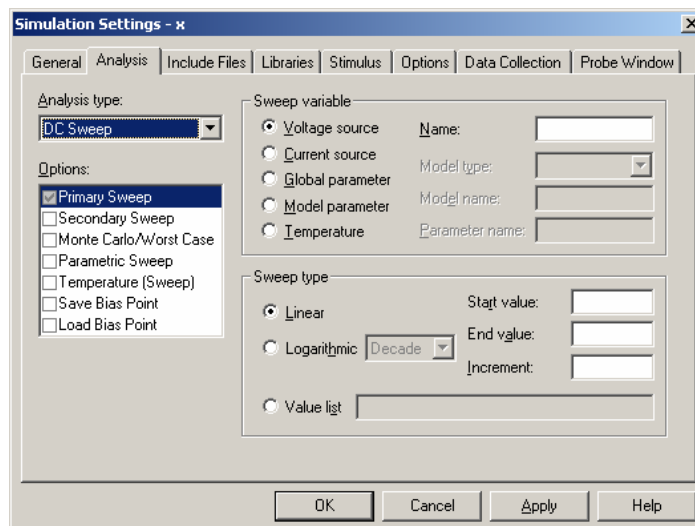
Element value Nominální hodnota parametru v základních jednotkách.

Element sensitivity (Volts/Unit) - Absolutní citlivost $\frac{\Delta F}{\Delta q}$. Je číselně rovna změně sledované veličiny F při změně parametru q o 1. Platí pouze v úzkém rozsahu kolem nominální hodnoty parametru.

Normalised sensitivity (Volts/Percent) - Semirelativní citlivost v procentech $\frac{q}{100} \frac{\Delta F}{\Delta q}$. Je číselně rovna změně sledované veličiny F při změně parametru q o 1% nominální hodnoty. Platí v úzkém okolí kolem nominální hodnoty parametru.

5.3 Stejnosměrné rozmítání (DC Sweep)

Analýza pracovního bodu obvodu v závislosti na rozmítaném parametru. Pro účely DC analýzy se ignorují setrvačné prvky - induktory se nahradí zkratem, kapacitory se neuvažují. Uplatní se všechny zdroje s nastaveným parametrem DC. Pokud chybí nastavení DC a zdroj obsahuje definici časového průběhu, tak je stejnosměrná složka odvozena z něj. Výsledky analýzy se ukládají do datového souboru pro zobrazení v Probe.



Obr. 5.4: Nastavení stejnosměrného rozmítání

Skupina **Sweep Variable** nastavuje typ rozmítané veličiny:

Voltage Source Stejnosměrná složka napěťového zdroje (DC). Hodnota nastavená ve schématu se ignoruje. Do pole **Name** se uvede jméno zdroje.

Current Source Stejnosměrná složka proudového zdroje (DC). Hodnota nastavená ve schématu se ignoruje. Do pole **Name** se uvede jméno zdroje.

Global Parameter - Rozmítání globálního parametru. Do pole **Parameter name** se uvede jeho jméno. Globální parametr je třeba deklarovat pomocí pseudosoučástky **PARAM**, 3.2.2.

Model Parameter Rozmítání parametrů modelu. Použití této položky vyžaduje znalost typu modelu a názvů jeho parametrů.

Příklad:

Rozmítání parametru **BF** (zhruba odpovídá β) bipolárního tranzistoru BC547A. Příslušný model v knihovně má tvar:

```
.MODEL BC547A NPN (..... BF=253 .....
```

Položky nastavíme takto:

Model:	NPN
Model Name:	BC547A
Parameter Name:	BF

Během rozmítání se hodnota **BF** z knihovny ignoruje. Rozmítání má vliv na všechny součástky odkazující se na daný model.

Temperature Teplota. Parametr **TNOM** (globální teplota) se ignoruje během rozmítání.

Pomocí **Sweep Type** se nastavuje typ rozmítání:

Linear Lineární rozmítání. Nastaví se počáteční a koncová hodnota a krok. Krok volíme tak velký, aby na simulační interval připadly řádově stovky bodů.

Logarithmic (Octave/Decade) - Logaritmické rozmítání. Nastaví se počáteční a koncová hodnota a počet bodů na oktávu nebo dekádu. Interval rozmítání nesmí obsahovat nulu.

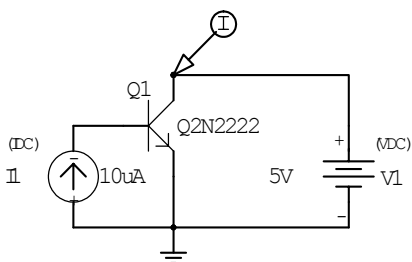
Value List Rozmítání v diskretních hodnotách ze seznamu. Do pole se napíše seznam, jehož položky jsou odděleny mezerou nebo čárkou.

Secondary Sweep umožňuje nastavit vnější smyčku k základnímu DC rozmítání, specifikovaný parametr „krokuje“. Dialogové okno je stejné jako pro základní analýzu. Není možné nastavit rozmítání a krokování stejné proměnné.

Pokud simulátor není schopen určit pracovní bod (řešení nekonverguje), tak mu můžeme „pomoci“ nastavením první iterace Newtonovy metody prostřednictvím pseudosoučástek NODESET.

Příklad 5.2 Výstupní charakteristiky tranzistoru

Vykreslení sady výstupních charakteristik tranzistoru 2N2222 (v knihovně je pod jménem Q2N2222) v zapojení SE, $U_{CE} \in [0, 10V]$ - plynulé rozmítání, $I_B \in [0, 50\mu V]$ - krokování po $10\mu A$ (vnější smyčka).



Na bázi připojíme proudový zdroj, např. IDC a na kolektor napěťový zdroj VDC. Zdroje musíme nastavit na „rozumné“ hodnoty protože se jako první analýza vždy počítá pracovní bod s nominálními hodnotami parametrů. Během analýzy DC se samozřejmě parametry rozmítaných zdrojů ze schématu neuplatní. Čtenář si může vyzkoušet, že pro nastavení $I1 = -10A$ není analyzátor schopen určit pracovní bod protože řešení nekonverguje. Budeme

vynášet závislost $I_C = f(U_{CE})$. Je možné použít proudový marker, takže dojde k automatickému vykreslení výstupních charakteristik.

DC Sweep (vnitřní smyčka):

Sweep Variable:	•Voltage Source	Name:	V1
Sweep Type:	•Linear	Start Value:	0
		End Value:	10V
		Increment:	40mV

Nested Sweep (vnější - krokovací smyčka):

Sweep Variable:	•Current Source	Name:	I1
Sweep Type:	•Linear	Start Value:	0
		End Value:	50uA
		Increment:	10uA

Po nastavení analýz zvolíme PSpice/Run (F11) a díky markeru ve schématu dostaneme v Probe ihned svazek výstupních charakteristik.

5.4 Střídavá analýza (AC Sweep)

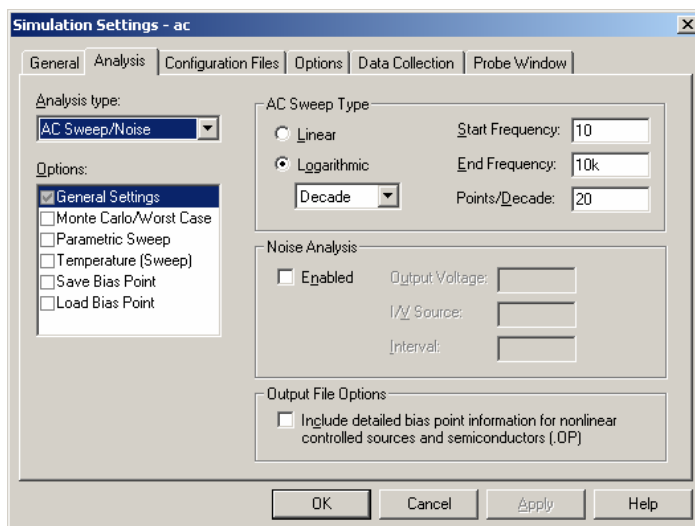
Analýza počítá ustálenou odezvu na malý harmonický signál obvodu linearizovaného v pracovním bodu (tj. přenosy, impedance). Rozmítanou veličinou je kmitočet, výsledky se zobrazují v Probe. Analýza je prováděna ve třech krocích:

1. Výpočet stejnosměrného pracovního bodu. Při výpočtu se uplatňuje stejnosměrná složka všech nezávislých zdrojů, tj. parametr DC, případně je odvozena z časového průběhu, pokud DC chybí.
2. Linearizace obvodu v pracovním bodu. Dále je uvažován již lineární model.
3. Výpočet odezvy na harmonický signál pomocí operátorového počtu ($j\omega$).

Napětí a proudy v obvodu jsou komplexní veličiny. V příloze jsou popsány funkce, které můžeme použít pro komplexní veličiny při vytváření grafů v Probe. Při výpočtu odezvy podle bodu 3 se uplatní AC složky nezávislých zdrojů. U speciálních zdrojů VAC a IAC lze nastavit amplitudu i úhel fázoru napětí respektive proudu pomocí parametrů ACMAG a ACPHASE. U ostatních zdrojů s výjimkou stejnosměrných existuje parametr AC, kterým se nastavuje amplituda fázoru, úhel je 0. Pokud v obvodu působí více zdrojů, pak se jejich účinky sčítají podle principu superpozice.



Ačkoli AC analýza počítá malosignálovou odezvu, tak vlastní výpočet se provádí v linearizovaném modelu. Velikost AC parametrů zdrojů již neovlivní výsledek. Pokud je v obvodu jeden budící zdroj, pak s výhodou volíme $AC = 1$. Přenosová funkce je potom číselně rovna napětí na výstupu.



Obr. 5.5: Nastavení střídavé analýzy

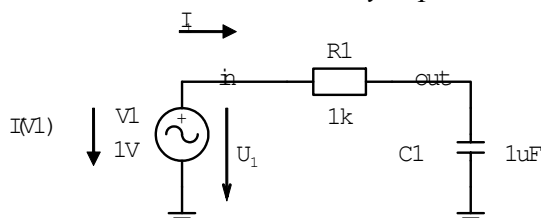
V nastavovacím okně se definuje pouze rozsah frekvenčního rozmítání. Okno je společné i pro šumovou analýzu, která bude popsána dále.

Linear lineární rozmítání. Nastaví se celkový počet bodů, počáteční a koncová frekvence.

Logarithmic (Octave/Decade) - logaritmické rozmítání. Nastaví se počet bodů na oktávu/dekádu, počáteční a koncová frekvence.

Pokud interval rozmítání přesahuje jednu dekádu, je vhodnější nastavit logaritmické rozmítání. Pak dostaneme ekvidistantní dělení intervalu v logaritmickém měřítku, což je obvyklé pro zobrazení frekvenčních charakteristik.

Příklad 5.3 *Analýza přenosu a impedance RC článku.*



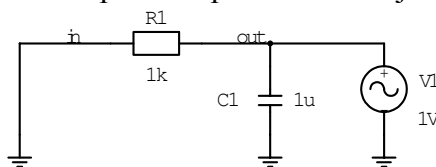
Vstupní a výstupní uzly označíme jako in a out (label). Zdroj V1 je typu VAC s parametry DC = 0, ACMAG = 1V, ACPHASE = 0. Zdroj generuje reálný fázor o velikosti 1V. V AC Sweep nastavíme logaritmické frekvenční rozmítání v intervalu $f \in [100\text{Hz}, 100\text{kHz}]$, 100

bodů na dekádu. Protože ve schématu nejsou žádné markery, tak po proběhnutí simulace dostaneme v Probe prázdný graf. Volbou Trace/Add (Insert) lze zadat vzorec pro kreslení požadované závislosti. Popis syntaxe pro volání proměnných a funkcí je v příloze. Následující odstavce popisují nejčastěji sledované veličiny.

Napěťový přenos $\mathbf{v(out) / v(in)}$, protože $v(in) = 1$ jde číselně (!) psát $K = v(out)$
 modul v decibelech: $\mathbf{db(v(out) / v(in))}$
 fáze ve stupních: $\mathbf{P(v(out) / v(in))}$

Vstupní impedance $\mathbf{v(in) / (-i(V1))}$, vstupní proud I_1 určíme jako proud zdrojem V1. Ten má ale opačný směr, protože všechny součástky v PSpice mají orientaci jako spotřebič, tj. od svorky + k -. Pokud v Probe zadáme kreslení komplexní veličiny, tak se v grafu zobrazí její absolutní hodnota (modul).

Výstupní impedance Pro určení výstupní impedance musíme přesunout budící zdroj. Vyjmutí vstupního napěťového zdroje znamená jeho nahrazení zkratem.

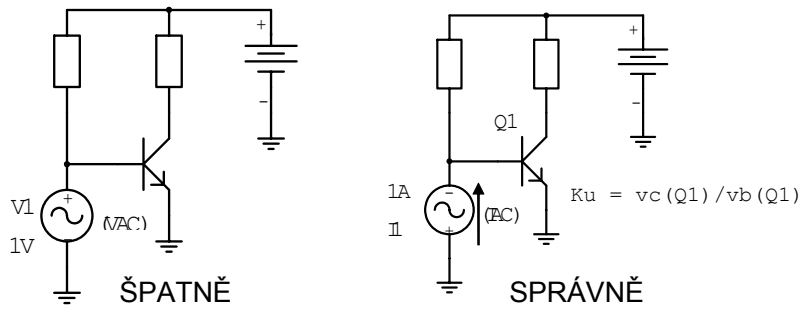


Výstupní impedanci pak určíme jako vstupní impedanci při buzení do výstupu
 $\mathbf{v(out) / (-i(v1))}$.



Pokud do obvodu vkládáme AC napěťové nebo proudové zdroje, je třeba mít na paměti, že se jedná o standardní napěťové a proudové zdroje, které svou přítomností ovlivní stejnosměrný pracovní bod. Pokud je jejich DC složka nulová, pak napěťový zdroj představuje zkrat a proudový zdroj rozpojený obvod. Tedy mezi volné uzly vložíme IAC, do větve pak VAC.

Příklad 5.4 *Připojení střídavého (AC) budícího zdroje k zesilovači SE bez použití oddělovací kapacity.*



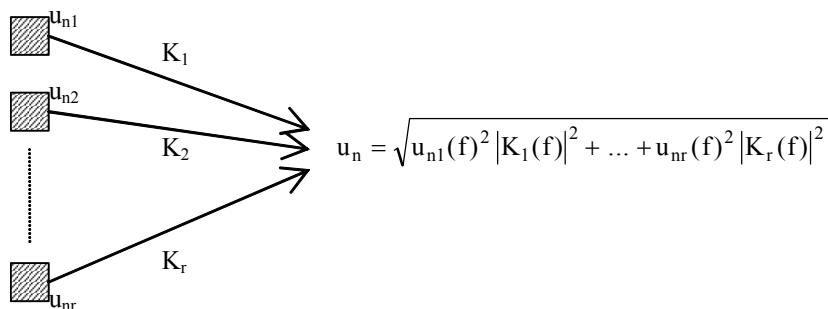
Zdroj IAC je orientován tak, aby kladný směr proudu byl do báze. Jeho DC složka je nulová.

5.4.1 Lineární šumová analýza

Šumová analýza obvodu je prováděna vždy v návaznosti na střídavou analýzu (AC). Analýza počítá výsledný šum na výstupu obvodu ve frekvenční oblasti vzniklý příspěvky jednotlivých součástí (rezistory a polovodiče). Předpokládá se, že vstupní signál (tj. budící zdroj) je bez šumu.

Protože šum je náhodná veličina, nemá smysl vyjadřovat jeho časový průběh. Pro popis se užívají spektrální charakteristiky. V technické praxi se šumový zdroj popisuje spektrální hustotou napětí $v_n(f)$ $\left[\frac{\text{V}}{\sqrt{\text{Hz}}} \right]$ resp. spektrální hustotou proudu $i_n(f)$ $\left[\frac{\text{A}}{\sqrt{\text{Hz}}} \right]$. Pokud například šumové napětí působí na rezistor R má dodávaný výkon hustotu $p_n(f) = \frac{u_n(f)^2}{R} \left[\frac{\text{W}}{\text{Hz}} \right]$ a

celkový šumový výkon v pásmu f_1, f_2 je $P = \int_{f_1}^{f_2} p_n(f) df$. Například rezistor o velikosti R produkuje naprázdno šumové napětí o hustotě $u_n(f) = \sqrt{4kTR}$ nezávislé na frekvenci (jedná se o bílý šum).



Obr. 5.6: Princip výpočtu výsledného šumu na výstupu obvodu

Pokud působí zdroj šumu na vstup obvodu s přenosovou funkcí $K(f)$, tak spektrální hustota napětí na výstupu je $u_{n2}(f) = |K(f)|u_{n1}(f)$. Jestliže v obvodu působí více zdrojů šumu, pak předpokládáme, že jsou nekorelované. Na výstupu se sčítají jejich výkony, tj. kvadráty napětí. Neplatí tedy princip superpozice. Pro přepočítání šumových příspěvků na výstup musíme znát

malosignálovou přenosovou funkci od zdroje šumu. To je důvod proč se šumová analýza provádí společně s AC.

Okno pro nastavení je společné s AC analýzou. Ve spodní části okna povolíme šumovou analýzu (Noise Analysis Enabled).

Output voltage Specifikace výstupního napětí, které definuje výstupní bránu obvodu.

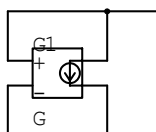
I/V source Jméno zdroje, který definuje vstupní bránu. **Nejedná se o šumový zdroj**, ale pouze o určení místa, kde je obvod buzen. Během analýzy se počítá výstupní šumové napětí **ONoise** a ekvivalentní vstupní šum v místě budícího zdroje. Jeho hodnota je $INoise = \frac{ONoise}{K(f)}$, kde $K(f)$ je přenosová funkce ze vstupní brány (určené zdrojem) na výstup.

Interval Nepovinná položka určující interval detailního výpisu o šumových poměrech do výstupního textového souboru *.out*.

Výstup šumové analýzy je ukládán společně s AC pro zpracování v Probe. Pokud navíc zadáme hodnotu **Interval = n**, pak každý *n*-tý krok dělení frekvenčního intervalu je do výstupního souboru *.out* zapsán detailní výpis šumových příspěvků. V Probe máme k dispozici veličiny $V(ONoise)$, $V(INoise)$, což je **hustota napětí** [V/\sqrt{Hz}] na výstupu a ekvivalentní hustota na vstupu. Ekvivalentní vstupní napětí je napětí, které musíme přivést na vstup nešumícího obvodu, abychom na výstupu dostali stejný šum jako v případě reálného (šumícího) obvodu.

Poznámky:

1. Efektivní hodnotu šumového napětí na výstupu obvodu ve frekvenčním pásmu f_1 , f_2 získáme integrací výstupní hustoty $u_{nef} = \sqrt{\int_{f_1}^{f_2} u_n(f)^2 df}$. Při simulaci nastavíme f_1 a f_2 jako meze AC analýzy a v Probe zadáme **SQRT(S(PWR(V(ONoise), 2)))**. Získáme graf, kde efektivní hodnota je hodnota integrálu na frekvenci f_2 , tj. na konci intervalu frekvenčního rozmítání.
2. Modelované zdroje šumu u jednotlivých součástek jsou popsány v referenčním manuálu. Teplota pro výpočet je určena parametrem **TNOM**, pokud není zvoleno krokování teploty.
3. Výsledek šumové analýzy je vždy optimističtější než skutečnost. Např. u rezistoru se modeluje pouze bílý tepelný šum, ale není modelován vliv materiálů rezistivní vrstvy (např. proudový šum).
4. Pokud potřebujeme vytvořit nešumící rezistor R , můžeme použít zdroj proudu řízený napětím (**G**), který realizuje závislost $I = G U$, kde $G = 1/R$.

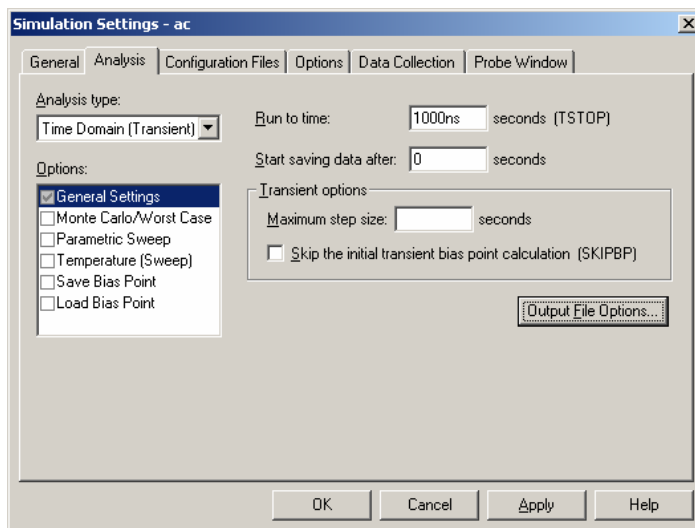


5.5 Analýza v časové oblasti (Transient)

Analýza obvodu v časové oblasti. Výstup je zobrazován v Probe. Pro výpočet se uplatní časové specifikace zdrojů, které mají přednost před nastavením DC parametrů. Pokud časová specifikace chybí (např. u stejnosměrného zdroje - VDC), tak se uvažuje parametr DC.



Parametr DC u zdrojů **nikdy** není stejnosměrná složka časového průběhu.



Obr. 5.7: Nastavení časové analýzy

Run to time Koncový čas simulace. Začátek je vždy v $t = 0$.

Start saving.... Ukládání dat začne až po uplynutí specifikovaného času. Používá se, pokud nás nezajímá přechodný děj na počátku simulace.

Maximum step.... Nastavení maximální délky časového kroku během simulace. Krok je adaptivně řízen podle průběhu veličin v obvodu. O hustotě kroku se můžeme přesvědčit v Probe při zobrazení datových bodů (Tools/Options/Mark Data Points). Pokud se hustota jeví jako nedostatečná, můžeme zkrátit časový krok. Nastavení je nepovinné.

Počáteční podmínky

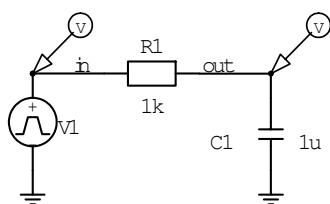
V časové oblasti vycházejí napětí a proudy obvodu v $t = 0$ z daných počátečních podmínek. Standardně se v $t = 0$ spočte pracovní bod a řešení vychází z něj. Jedná se o jiný pracovní bod než se běžně počítá (DC, AC), protože místo DC parametrů zdrojů se uvažuje jejich časová hodnota pro $t = 0$.

U setrvačných součástek (L a C) můžeme nastavit počáteční podmínku parametrem IC. Nevýhodou je, že nepoznáme orientaci podle schematické značky. Druhou možností je použití pseudosoučástky IC1 pro nastavení napětí libovolného uzlu proti referenčnímu uzlu a IC2 pro nastavení napěťového rozdílu mezi uzly. Tyto uživatelské počáteční podmínky jsou uvažovány jako přídavné napěťové a proudové zdroje během výpočtu pracovního bodu pro nastartování časové analýzy. Potom jsou odpojeny a přechodný děj vychází z vypočteného pracovního bodu.



Pomocí volby **Skip the Initial Transient Bias Point Calculation** máme možnost potlačit počítání pracovního bodu před časovou analýzou. Uživatelské počáteční podmínky (IC) stále platí, ostatní veličiny jsou rovny 0. Tato volba je riskantní, protože výpočet pracovního bodu zajišťuje konzistentní startovací podmínky pro integrační proceduru. V některých případech může způsobit, že řešení nekonverguje v prvním časovém kroku.

Příklad 5.5 *Buzení integračního článku obdélníkovým signálem $f = 1\text{kHz}$ se střídou 1:1.*



Parametry budící zdroje VPULSE nastavíme:

TD = 0.499ms, TR = 1us, PW = 0.499ms, TF = 1us, PER = 1ms, V1 = 0V, V2 = 1V.

Přestože požadujeme obdélníkový signál, není vhodné nastavit nulové délky nástupné a sestupné hrany, protože by mohly nastat problémy s konvergencí v časové oblasti (zvláště u obvodů s nelineárními prvky). Simulaci budeme provádět v rozsahu 10 period bez zvláštních požadavků na omezení časového kroku.

Run to Time: 10ms
 Start Saving... 0
 Maximum step size

Na vstup a výstup připojíme napěťový marker, takže dojde k automatickému vykreslení průběhu po skončení simulace.

5.5.1 Spektrální analýza (Fourier Analysis)

Fourierova analýza je prováděna na průběhu získaném časovou simulací. Samostatně tedy nemá smysl. Dialog pro nastavení se otevírá tlačítkem **Output File Options** v okně pro časové analýzy. Musíme zadat frekvenci první harmonické tj. periodu T. Před transformací je navzorkován poslední úsek délky T v časovém průběhu. Vzorkovací interval je určen nastavením **Print Step**, nikdy však nepřekročí 1% z délky simulačního intervalu. Pro správné fungování je nutné, aby čas celkový simulace byl delší než jedna perioda. Fourierova analýza má smysl pouze tehdy, pokud je navzorkována jedna perioda signálu v ustáleném stavu.

Perform Fourier Analysis - povolení provedení analýzy;

Center Frequency - frekvence první harmonické;

Number of Harmonics - počet harmonických, položka je nepovinná. Pokud nezadáme nic, tak se počítá stejnosměrná složka a 1. až 9. harmonická.

Output Variables - seznam napětí a proudů oddělených mezerou, které se budou analyzovat.

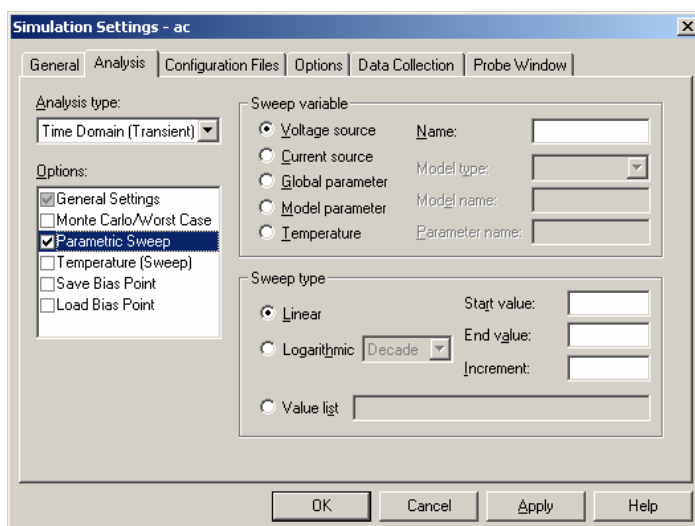
Výstup analýzy je ukládán do textového souboru *.out*. Kromě údajů o harmonických složkách je zde i činitel harmonického zkreslení.

5.6 Doplnkové analýzy

K základním analýzám mohou být aktivovány doplňkové, které využívají jejich výsledků. Např. krokování parametru, Monte Carlo, aj..

5.6.1 Krokování parametru (Parametric Sweep)

Nastavení vícenásobné analýzy, která slouží jako vnější smyčka k AC, DC nebo Transient.



Obr. 5.8: Nastavení krokování

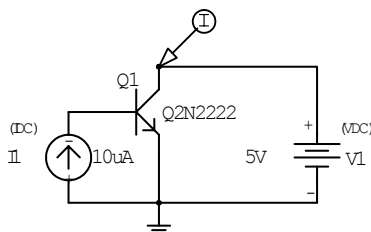
Nastavovací okno je podobné jako pro stejnosměrnou analýzu. Rozdíl spočívá v tom, že nastavená veličina krokuje jako parametr některé základní analýzy.



Není možné současně rozmítat a krokovat stejnou veličinu.

Příklad 5.6 *Zobrazení výstupních charakteristik tranzistoru Q2N2222.*

Příklad je stejný jako v kap. 5.3. Nyní se nepoužije Secondary Sweep (musí být vypnuto), ale vícenásobná analýza.



DC Sweep (vnitřní smyčka):

Sweep Variable:	•Voltage Source	Name:	V1
Sweep Type:	•Linear	Start Value:	0
		End Value:	10V
		Increment:	40mV

Parametric Sweep (vnější - krokovací smyčka):

Sweep Variable:	•Current Source	Name:	I1
Sweep Type:	•Linear	Start Value:	0
		End Value:	50uA
		Increment:	10uA

Simulace se spustí volbou PSpice/Run (F11). Protože je ve schématu marker, dostaneme po skončení simulace ihned graf.

5.6.2 Teplotní analýza (Temperature (Sweep))

Teplota obvodu je určena globální proměnnou TNOM, která se nastavuje v záložce Options simulačního profilu. Rozmítání teploty je možné nastavit v analýze DC (DC Sweep) a vícenásobné (Parametric Sweep). Kromě toho existuje speciální položka Temperature, kde můžeme nastavit seznam teplot, ve kterých se bude provádět zvolená analýza. Posledně jmenovaná možnost je méně komfortnější než vícenásobná analýza s nastavením krokování teploty.

Teplotní závislost je standardně obsažena v modelech polovodičů. Pro ostatní prvky musí uživatel specifikovat teplotní koeficient sám prostřednictvím modelu (příkaz .MODEL). K tomu účelu existuje knihovna značek *breakout* obsahující součástky s prázdnými modely, které jsou určené k uživatelské úpravě (ve jménech těchto prvků je obsaženo „break“, např. Cbreak, Rbreak). V modelech je možné specifikovat teplotní koeficienty, jejich jména a význam je uveden v popisu každého modelu v referenční příručce. Kromě toho je možné ve všech modelech definovat následující parametry:

T_MEASURED= Teplota, při které byly určeny (naměřeny) parametry součástky. Pokud není specifikována, tak $T_MEASURED = TNOM$.

Příklad: hodnota rezistoru je dána vztahem

$$RT = R (1+TC) (TEMP-T_MEASURED)$$

T_ABS= Pokud není tento parametr definován, tak je teplota součástky rovna globální teplotě obvodu (TNOM) nebo jiné hodnotě v případě rozmítání teploty. Pokud je **T_ABS** specifikována, pak určuje konstantní teplotu součástky nezávislou na globální teplotě.

T_REL_GLOBAL= - Udává rozdíl mezi globální teplotou obvodu a teplotou součástky ($T = TEMP+T_REL_GLOBAL$). Pokud není v modelu specifikována, tak $T_REL_GLOBAL = 0$, tj. teplota součástky je rovna globální teplotě.

Příklad 5.7 *Nastavení lineárního teplotního koeficientu kondenzátoru.*

Do schématu vložíme součástku Cbreak. Označíme ji, stiskneme pravé tlačítko myši a zvolíme Edit PSpice Model. Otevře se okno editoru, kde v textové formě doplníme model o položku $TC1 = 2 \cdot 10^{-3} K^{-1}$. Upravený model se vždy ukládá do lokální knihovny příslušející danému projektu.

```
.model Cbreak CAP C=1 TC1=2m ; vložený teplotní koeficient
```

5.6.3 Statistická a toleranční analýza (Monte Carlo/Worst Case)

Parametry součástek, z nichž jsou sestaveny reálné obvody, vždy vykazují jistý rozptyl kolem nominální hodnoty. Během analýzy **Monte Carlo** jsou parametry vybírány náhodně z tolerančního intervalu. Pro každý výběr se stanoví hledaná funkce obvodu (šířka pásma, zisk, doba náběžné hrany atd.). Při dostatečném počtu výběru jsme schopni odhadnout v jakém intervalu bude ležet sledovaná funkce. Analýza **Worst Case** hledá takové rozložení hodnot parametrů součástek v rámci tolerančních intervalů, které znamená nejhorší možnou hodnotu sledované veličiny. V programu PSpice je nutnou podmínkou nalezení nejhoršího případu monotónní závislost sledované funkce na všech parametrech, u kterých je definovaná tolerance.

Specifikace tolerancí

Tolerance mohou být specifikovány pouze v modelu součástky (příkaz `.MODEL`). Za definicí číselné hodnoty parametru se napíše klíčové slovo `LOT` a nebo `DEV` a hodnota relativní (v %) či absolutní tolerance. Přesná syntaxe příkazu `.MODEL` je v příloze. Rozdíl mezi `DEV` a `LOT` spočívá ve způsobu určování hodnot parametrů součástek při náhodném výběru z tolerančního intervalu.

LOT	Pro určení hodnot parametrů všech součástek, které se odkazují na daný model se užije stejné náhodné číslo. Hodnoty jsou 100% korelované.
DEV	Pro každou součástku odkazující se na daný model se použije jiné náhodné číslo. Hodnoty nejsou korelované.

Specifikace `DEV` a `LOT` se mohou kombinovat, výsledná tolerance je pak dána jejich součtem. Pokud se na model odkazuje pouze jediná součástka, pak mezi `DEV` a `LOT` není rozdíl.

Toleranci `DEV` bychom použili pro obvod, který obsahuje např. několik diskretních tranzistorů téhož typu (odkaz na společný model) vybraných z velké série. Zde se můžeme domnívat, že jejich parametry nejsou korelované. Druhým spíše teoretickým extrémem je několik stejných tranzistorů na jednom čipu. Jejich parametry jsou shodné, ale jednotlivé čipy mohou vykazovat toleranci např. až 30%. V tomto případě použijeme `LOT` - parametry jsou 100% korelované. V praxi oba výše popsané extrémy nenastávají. Např. pro integrované obvody může být tolerance až 30 % mezi různými čipy a kolem 3% mezi tranzistory na stejném čipu. Zde použijeme kombinaci `LOT` a `DEV`:

LOT 30% DEV 3% .

V knihovnách dodávaných k PSpice nejsou tolerance specifikovány. Uživatel musí daný model upravit sám. Z prostředí schematického editoru je to možné volbou `Edit PSpice Model`.



Toleranci nelze definovat pro teplotní koeficienty v modelech.

Příklad 5.8 *Nastavení tolerance parametru **BF** tranzistoru Q2N2222.*

Pomocí `Edit PSpice Model` doplníme definici tolerancí:

```
.model Q2N2222 NPN (
..... ; zkráceno
Bf=255.9 DEV 1% LOT 100
..... ; zkráceno
Rb=10)
```

Je vidět, že tolerance mohou být definovány jako relativní (v %) nebo absolutní, případně se mohou kombinovat. Specifikace tolerance se píše za definici parametru, v našem případě $BF = 255.9$.

Poznámky:

1. Součástky R, C a L mají parametr TOLERANCE přímo ve schematické značce. Zde můžeme zadat pouze relativní toleranci (v %). Protože každý takový prvek ve schématu používá svůj model, tak tolerance jsou nekorelované (odpovídá DEV). Pokud je potřeba definovat model pro prvky, které běžně model nemají, tak použijeme součástky z knihovny *breakout* (Cbreak, Rbreak atd. – viz přílohy) a jejich model upravíme prostřednictvím volby Edit PSpice Model.
2. Definice tolerancí má více možností než bylo popsáno v této kapitole, kompletní přehled lze nalézt v referenční příručce.

Jednobodová analýza

Analýzy Monte Carlo a Worst Case pracují jako nadstavba nad některou základní analýzou (AC, DC, Transient). Pro určení statistických vlastností sledované funkce obvodu potřebujeme, aby výsledkem základní analýzy byla jediná hodnota. Obvykle jako výsledek dostaneme křivku vzniklou rozmítáním nezávislé proměnné. Existují dvě možnosti jak převést výsledek základní analýzy do jediné hodnoty:

1. Vícenásobná analýza pomocí Probe, kap. 4.2.4.
2. Jednobodová analýza.

Jednobodová analýza je případ, kdy nezávislá (rozmítaná) veličina na ose X nabývá pouze jediné hodnoty. Z fyzikálních důvodů je to možné pouze pro AC a DC, pro Transient nemá jednobodová analýza smysl.

AC: nastaví se stejná počáteční a koncová frekvence a 1 bod.

Příklad:	AC Sweep Type:	• Linear
	Total Points:	1
	Start Freq.:	1kHz
	End Freq.:	1kHz

DC: nastaví se „pseudorozmítání“ nějakého parametru, např. napájecího zdroje v jediné diskrétní hodnotě.

Příklad:	Sweep Variable	• Voltage Source
	Name:	V1
	Sweep Type:	• Value List
	Values:	12V

Výsledkem jednobodové analýzy je jediná hodnota, v Probe se zobrazí jako křížek v grafu.

Rozložení hustoty pravděpodobnosti

Při výběru hodnot z tolerančních intervalů se užívá náhodná veličina, jejíž nejdůležitější charakteristikou je rozložení hustoty pravděpodobnosti, které označíme jako $p(x)$. V anglických manuálech se používá název *distribution*.

Některé vlastnosti:

$$1. p(x) \geq 0, \quad \int_{-\infty}^{\infty} p(x) dx = 1$$

2. Pravděpodobnost, že reálná náhodná veličina ξ leží v intervalu $[x_1, x_2]$ je

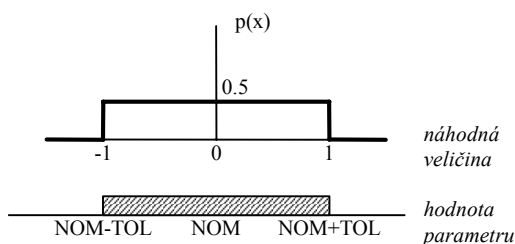
$$P(\xi \in [x_1, x_2]) = \int_{x_1}^{x_2} p(x) dx.$$

Výběr je prováděn podle vzorce $A = A_{\text{NOM}} (1 + \text{TOL}_{\text{REL}} \text{RND})$ kde TOL_{REL} je relativní tolerance a RND náhodné číslo s příslušným rozložením hustoty pravděpodobnosti. A_{NOM} je nominální hodnota parametru.

V programu PSpice jsou vestavěny dvě funkce hustoty a uživatel má možnost definovat vlastní. Jméno funkce hustoty se nastavuje jako parametr u analýzy Monte Carlo. Standardní hodnota je UNIFORM (= rovnoměrné rozložení), druhý vestavěný typ je GAUSS.

• UNIFORM

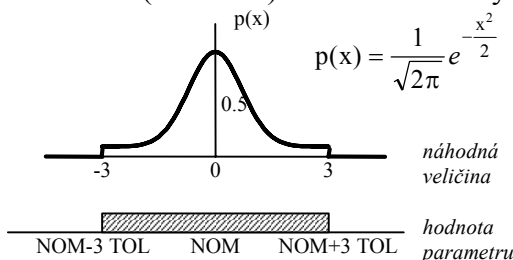
Rovnoměrné rozložení hustoty pravděpodobnosti na intervalu $[-1, 1]$. Každá hodnota z intervalu je stejně pravděpodobná.



Hodnota parametru se nachází v intervalu $[\text{NOM}-\text{TOL}, \text{NOM}+\text{TOL}]$, kde TOL je absolutní hodnota tolerance podle LOT a DEV, NOM je nominální hodnota parametru.

• GAUSS

Gaussovo (normální) rozložení hustoty pravděpodobnosti.



$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

I když $p(x) > 0$ pro $x \in \mathbb{R}$, tak v PSpice je interval náhodné veličiny omezen na $[-3, 3]$ z praktických důvodů. U normálního rozložení je pravděpodobnost že náhodná veličina leží v intervalu $[-1, 1]$ 68%.



Krajní hodnoty intervalu pro výběr jsou $\text{NOM} \pm 3 \text{TOL}$ (tj. interval 3σ). Při zadávání tolerancí je třeba dbát na to, aby při náhodném výběru nebyly vybrány záporné hodnoty, tedy mělo by platit $\text{NOM} > 3 \text{TOL}$.

• Uživatelská definice

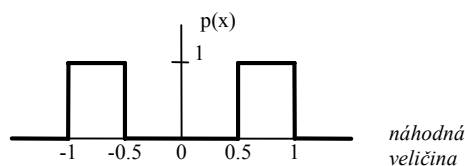
Definice funkce hustoty se vytváří v dialogovém okně, které otevřeme tlačítkem Distributions v nastavení analýzy Monte Carlo. Formát definice je:

$$(x1, p(x1)) (x2, p(x2)) \dots (xn, p(xn))$$

$x_i, p(x_i)$ Funkce hustoty pravděpodobnosti je aproximována lomenou přímkou. Zadávají se její uzlové body na intervalu [-1, 1].

Funkce se uloží jako součást projektu.

Příklad 5.9 Definice uživatelské funkce hustoty pravděpodobnosti.



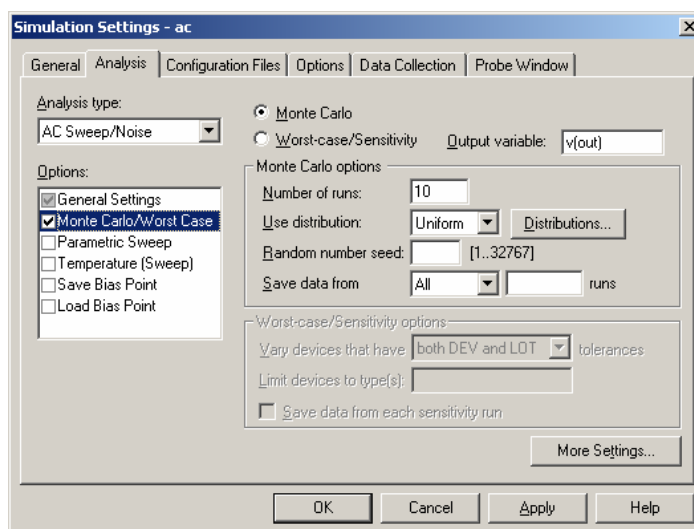
$(-1, 1) (-0.5, 1) (-0.5, 0) (0.5, 0) (0.5, 1) (1, 1)$

Poznámky:

1. Intervaly se mohou v krajních bodech překrývat, např. $(0.5, 0) (0.5, 1)$.
2. Plocha pod křivkou nemusí být rovna 1, PSpice provede automaticky normování.

Analýza Monte Carlo

Nastavení je ve společném okně s analýzou Worst Case. Obě analýzy nelze provádět současně. Výstup analýzy je vždy ukládán do textového výstupního souboru *.out*. Kromě toho lze výsledky zobrazit i v Probe. Ve výstupním souboru dostáváme detailnější informace, ale jsme velmi omezeni v možnostech pro výběr sledované funkce obvodu ve srovnání s Probe. Proto bude dále popsáno jen použití analýzy v Postprocesoru.



Obr. 5.9: Nastavení analýzy Monte Carlo

Output Variable - Definice sledované veličiny. **Neuplatní se při zpracování výsledků v Probe.** Syntaxe pro definici je v příloze. Pokud nebudeme používat výsledky v textovém výstupním souboru *.out*, tak do této položky můžeme nastavit libovolnou veličinu v obvodu.

Number of Runs - Počet běhů analýzy. Nejdříve se obvod simuluje pro nominální hodnoty parametrů a potom se vykoná počet výběrů stanovený v tomto poli. Ze statistického hlediska by měl být počet běhů co největší. Na druhou stranu jsme omezeni dobou řešení a velikostí výstupního datového souboru pro Probe. V nastavení profilu můžeme například omezit ukládání výsledků jen na ty uzly, kde jsou markery.

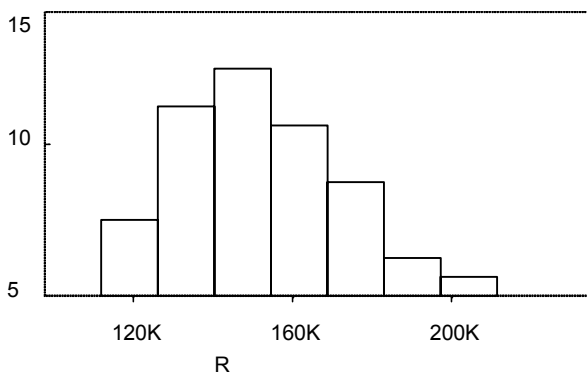
Use Distribution – volba globální funkce rozložení hustoty pravděpodobnosti. Uživatel může definovat vlastní rozložení (tlačítko **Distributions**).

Seed Záradečné číslo pro generátor náhodných čísel v rozsahu 1 - 32767. Pokud se nic neuvede, tak je použita hodnota 17533.

Save Data from - nastavení výstupu do datového souboru pro Probe. Zde by mělo být nastaveno **All**.

Nastavení v okně „**More Settings**“ se vztahuje k výsledkům v textovém souboru.

Zpracování výsledků v Probe



Obr. 5.10: Histogram

Pokud je výsledkem dílčího běhu analýzy Monte Carlo křivka, pak se v Probe zobrazí svazek křivek pro všechny běhy včetně nominálního. Pokud je výsledkem analýzy jeden bod (jednobodová analýza), pak je přímo zobrazen histogram, viz následující příklad. Pro zobrazení histogramu v prvním případě musíme v Probe zvolit tzv. vícenásobnou analýzu (Performance Analysis). Objeví se prázdný histogram, volbou **Trace/Add** musíme zadat měřicí funkci, tj. funkci, která z křivky odvodí

jedinou hodnotu pro statistické zpracování. Detailní popis srovnávací analýzy včetně příkladů je v kap. 4.2.4.

Histogram je sloupcový graf, který má na ose *X* sledovanou veličinu rozdělenou na intervaly. Výška každého sloupce udává četnost v procentech, že sledovaná veličina leží v intervalu určeném šířkou sloupce. Při dostatečném počtu běhů je histogram diskrétní aproximací funkce rozložení pravděpodobnosti. Počet dělení histogramu se nastavuje v Probe v položce **Tools/Options/Number of Histogram Divisions**, standardní hodnota je 10.

Pod histogramem jsou v Probe uvedeny některé statistické charakteristiky sledované veličiny:

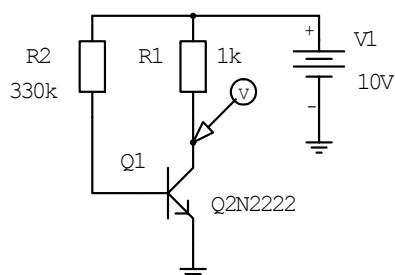
n samples	Počet běhů analýzy Monte Carlo.
n divisions	Počet dělení histogramu.
mean	Aritmetický průměr hodnot sledované veličiny všech běhů.

sigma	Směrodatná odchylka $\sigma = \sqrt{\frac{1}{n-1} \left[\sum_{j=1}^n x_j^2 - \frac{1}{n} \left(\sum_{j=1}^n x_j \right)^2 \right]}$, kde x_i jsou hodnoty sledované funkce.
minimum	Nejmenší hodnota sledované funkce.
10th %ile	Taková hodnota sledované veličiny (kvantil), že 10% běhů má nižší nebo rovnou hodnotu a 90% má vyšší hodnotu.
median	Hodnota sledované funkce (kvantil), kdy 50% běhů má menší hodnotu a 50% má větší hodnotu.
90th %ile	Taková hodnota sledované veličiny, kdy 90% běhů má menší nebo rovnou hodnotu a 10% větší hodnotu.
maximum	Největší hodnota sledované funkce.

Výše uvedené charakteristiky jsou samy o sobě náhodná čísla, protože byly stanoveny z konečného počtu výběrů. Čím je vyšší počet běhů, tím jsou přesnější.

Příklad 5.10 *Analýza rozptylu pracovního bodu*

Určete v jakém intervalu se bude nacházet U_{CE} v pracovním bodu. Tolerance rezistorů jsou 5% (nekorelované) a tolerance parametru BF tranzistoru (zhruba odpovídá β) je 30%.



Toleranci rezistorů nastavíme parametrem TOLERANCE a pomocí Edit PSpice Model doplníme model tranzistoru o definici tolerance: **BF = 255.9 DEV 30%**. Na kolektor umístíme napěťový marker.

Základní analýza DC bude nastavena jako jednobodová:

Sweep Variable:	• Voltage Source,	Name:	V1
Sweep Type:	• Value List	Values:	10V

Nastavení Monte Carlo:

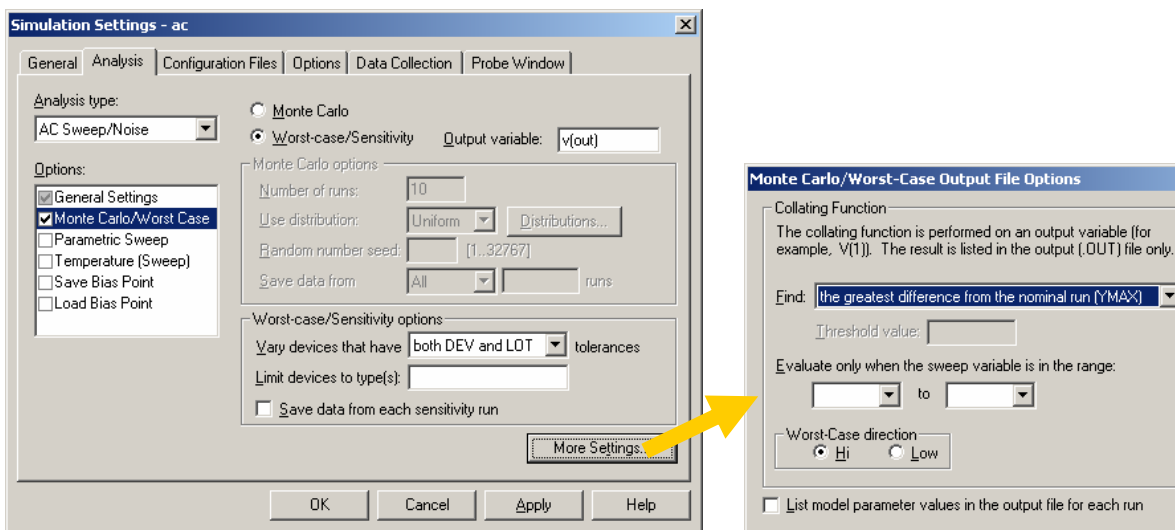
Analysis:	• Monte Carlo	Number of Runs:	100
Output Var.:	VC(Q1)	Distribution:	Uniform
Save data:	• All		

Nastavení výstupní proměnné má vliv pouze na výsledky ukládané do souboru *.out*. Analýzu spustíme pomocí PSpice/Run (F11). Po proběhnutí se v Probe ihned zobrazí histogram, protože výsledkem základní analýzy DC je jeden bod a ve schématu je umístěn marker.

Interval pro toleranci napětí U_{CE} můžeme odhadnout z údajů maximum a minimum (v Probe). V intervalu určeném 10% a 90% kvantilem se pracovní bod bude nacházet s 80% pravděpodobností. Uvedené údaje platí pro rovnoměrné rozložení hustoty pravděpodobnosti (UNIFORM). Statistické parametry jsou závislé na počtu běhů, o tom se můžeme přesvědčit opakováním analýzy pro jiné počty (položka Number of Runs).

Analyza Worst Case

Analýza určí takové hodnoty parametrů součástek v rámci jejich tolerančních intervalů, kdy je hodnota sledované veličiny v obvodu z nějakého hlediska nejhorší. Sledovaná veličina musí záviset monotónně na všech parametrech, u kterých je definována tolerance. V tom případě se všechny prvky budou nacházet v krajních bodech svých tolerančních intervalů. Výpočet probíhá tak, že je nejdříve určena citlivost výsledné funkce na jednotlivé parametry podléhající toleranci. Tím dostane program informaci o směru, ve kterém leží nejhorší případ. V posledním běhu je tento případ nastaven a spočítá se hodnota sledované funkce. Výstup analýzy je ukládán pouze do souboru `.out` a na rozdíl od Monte Carlo jsme omezeni jen na měřicí funkce YMAX, MAX, MIN, RISE, FALL. V postprocesoru je zobrazen pouze nominální a nejhorší průběh.



Obr. 5.11: Nastavení analýzy Worst Case

Output Variable - Definice sledované veličiny (syntaxe je uvedena v příloze).

Find Způsob, jakým se z výsledků základní analýzy odvodí jediná hodnota pro určení nejhoršího případu.

YMAX - největší odchylka od nominálního běhu,

MAX/MIN - maximální/minimální hodnota běhu (tj. křivky, jež je výsledkem základní analýzy),

RISE/FALL - hodnota nezávislé proměnné (čas, frekvence), při které průběh poprvé protne úroveň definovanou v položce Rise/Fall;

Range - omezení intervalu na ose X , kde se provádí hledání (nepovinné).

Limit devices.. Omezení tolerancí jen pro určité typy součástek uvedené v poli Devices kam se napíše seznam kódových písmen těchto modelů bez mezer (bipolární tranzistor - Q, rezistor - R, atd. viz příloha C).

Vary Dev/Lot/Both - Určení, které tolerance se budou uvažovat.

Direction Určení, kterým směrem od nominální hodnoty leží nejhorší případ. Pro YMAX a MAX je standardně nastavena hodnota Hi, pro ostatní je Lo.

Poznámky:

1. Pokud je nastaveno normální rozložení hustoty pravděpodobnosti (GAUSS), tak je uvažovaný toleranční interval 3 (LOT+DEV), jinak (LOT+DEV).

2. Pokud je nastaveno **Vary •Both** a parametr modelu má obě tolerance (LOT a DEV), tak je pro určení citlivosti použita jenom tolerance LOT, tj. parametr všech součástek odkazujících se na model se mění současně. Pokud se vyskytují modely s oběma definicemi, doporučuje se nejdříve provést **Vary •Lot**, ručně nastavit parametry podle výsledků a pak provést **Vary •Dev**.
3. Pokud je analýza Monte Carlo provedena s velkým počtem běhů, tak z hodnot **maximum** a **minimum** můžeme určit přibližně nejhorší případ. Potom není nutná monotónní závislost sledované veličiny na parametrech.

Interpretace výsledků

Výsledky analýzy se ukládají do výstupního souboru `.out`, který lze otevřít volbou **PSpice/View Output File**. Ve výstupu jsou uvedeny citlivosti na změnu parametrů a na konci je uvedena tabulka všech parametrů s uvedením hodnoty pro dosažení nejhoršího případu.

Příklad 5.11 *Určení nejhoršího případu odchylky polohy pracovního bodu*

Úkolem bude stanovit nejhorší odchylku pracovního bodu od nominální hodnoty pro stejný obvod jako v předchozím příkladu. Vzhledem k jednoduchosti obvodu můžeme intuitivně předpokládat monotónní závislost napětí U_{CE} na všech parametrech, u kterých je definována tolerance.

Základní analýza DC bude nastavena jako jednobodová:

Sweep Variable: • Voltage Source, Name: V1
Sweep Type: • Value List Values: 10V

Nastavení Worst Case:

Analysis: •Worst Case Output Var: VC(Q1)
Function: YMAX Vary: •Both
Direction: •Hi

Výsledek se nachází ve výstupním textovém souboru:

```

*****
                WORST CASE ALL DEVICES
*****
    DEVICE          MODEL          PARAMETER   NEW VALUE
    Q_Q1            Q2N2222-X   BF          179.13      (Decreased)
    R_R1            R_R1         R           .95         (Decreased)
    R_R2            R_R2         R           1.05       (Increased)
*****
****          SORTED DEVIATIONS OF VC(Q_Q1)  TEMPERATURE = 27.000 DEG C
                WORST CASE SUMMARY
*****
    RUN                MAX DEVIATION FROM NOMINAL
    ALL DEVICES        1.4676 higher at V_V1 = 10
                      ( 129.38% of Nominal)

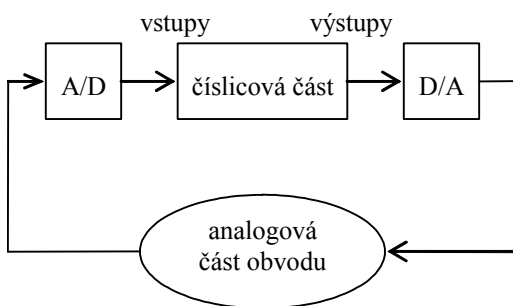
```

Hodnoty BF a R1 jsou na spodním okraji tolerančního intervalu, R2 má maximální hodnotu. Napětí U_{CE} je v tomto bodě o 1.47V vyšší než nominální hodnota.

5.7 Simulace číslicových a smíšených obvodů

5.7.1 Základní principy

Dělení na digitální a analogové obvody bylo zavedeno z praktických důvodů. Velikost běžných logických obvodů může dosahovat tisíců „hradel“. Pokud bychom každý logický člen nahradili analogovým modelem, pak dostáváme obvod o velikosti tisíců až desítek tisíc tranzistorů. Takový obvod prakticky není možné simulovat. Pokud při návrhu logické části dodržíme jistá kritéria (logický zisk, blokování napájecího napětí, přeslechy na sběrnici atd.), tak nás při analýze již nezajímají napětí a proudy, ale pouze logické stavy. Potom je možné použít zjednodušené modely logických členů.



Obr. 5.12: Rozdělení odvodu na analogovou a digitální část

Logické členy v PSpice zpracovávají signály nabývající diskretních hodnot (symbolů). Každý logický člen má jednoznačně definovaný vstup a výstup, modeluje pouze logickou funkci a zpoždění. Pokud propojíme analogový a digitální prvek, tak PSpice do místa spojení automaticky zařadí A/D nebo D/A převodník, který převádí analogové veličiny na logické stavy a naopak. Logické stavy jsou definovány jako signály, logické členy nemají referenční uzel ani napájení. Napájení potřebují pouze A/D a D/A převodníky. Vkládání převodníků a

zdrojů je zcela automatické. Pokud uživatel nemá žádné specifické požadavky, tak se tímto problémem nemusí zabývat.

Logické symboly:

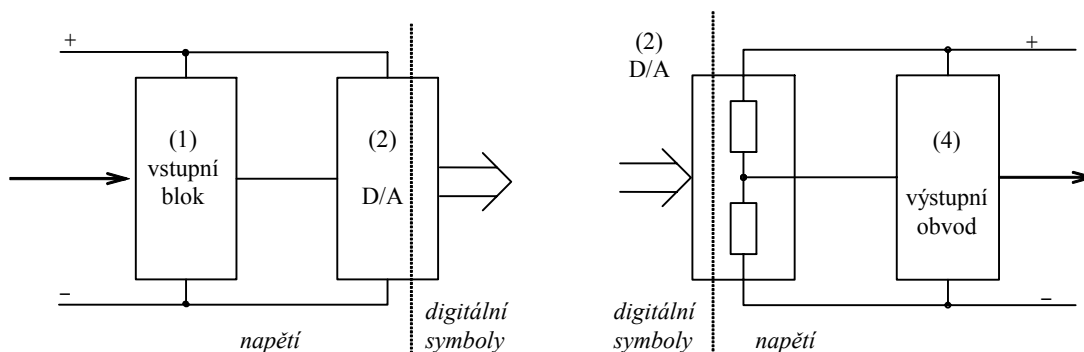
- 0, 1 Klasické prvky Booleovy algebry.
- R Přechod $0 \rightarrow 1$. Trvání tohoto symbolu vyjadřuje časovou neurčitost, v kterémkoliv okamžiku se může objevit přechod.
- F Přechod $1 \rightarrow 0$, podobně jako R.
- X Neurčitý stav. Vyjadřuje neurčitost hodnoty signálu.
- Z Stav vysoké impedance. Uplatní se pouze pro řízení D/A převodníků. Pokud je přiveden na vstup logického členu, tak se převádí na X.

Tabulka 5.1: Rozšíření základních logických funkcí

x	y	x AND y	x OR y	\bar{y}	poznámka
0	0	0	0	1	
0	1	0	1	0	
0	R	0	R	F	
0	F	0	F	R	
0	X	0	X	X	
1	1	1	1		
1	R	R	1		
1	F	F	1		
1	X	X	1		
R	R	R	R		
R	F	R	F		souběh neurčitostí (hazard)
R	X	X	X		
F	F	F	F		
F	X	F	X		
X	X	X	X		

A/D a D/A převodníky

A/D převodník je vždy zařazen při připojení analogového uzlu na vstup číslicové součástky. Převodník se skládá ze dvou bloků. Blok (1) modeluje vstupní část reálného logického obvodu, je to analogový makroobvod. Za ním následuje speciální součástka (2), která podle předpisu převádí napětí na logické symboly. Bloky (1) a (2) jsou „napájeny“ ze skrytých napěťových zdrojů (např. pro TTL a HC je to +5V).



Obr. 5.13: Struktura A/D a D/A převodníku

D/A převodník je zapojen vždy mezi výstup digitálního prvku a analogový uzel. Obsahuje speciální součástku (3) pro převod logického symbolu do analogové úrovně. Tento blok obsahuje dva proměnné rezistory, jejichž hodnota je závislá na přivedeném symbolu. Blok (4) představuje makroobvod pro modelování výstupní části logického členu.

5.7.2 Číslicové prvky a jejich parametry

Všechny číslicové součástky mají dva modely: časovací a vstupně výstupní, který se uplatní jen při připojení prvku na analogový uzel (podle modelu je nastaven D/A a A/D

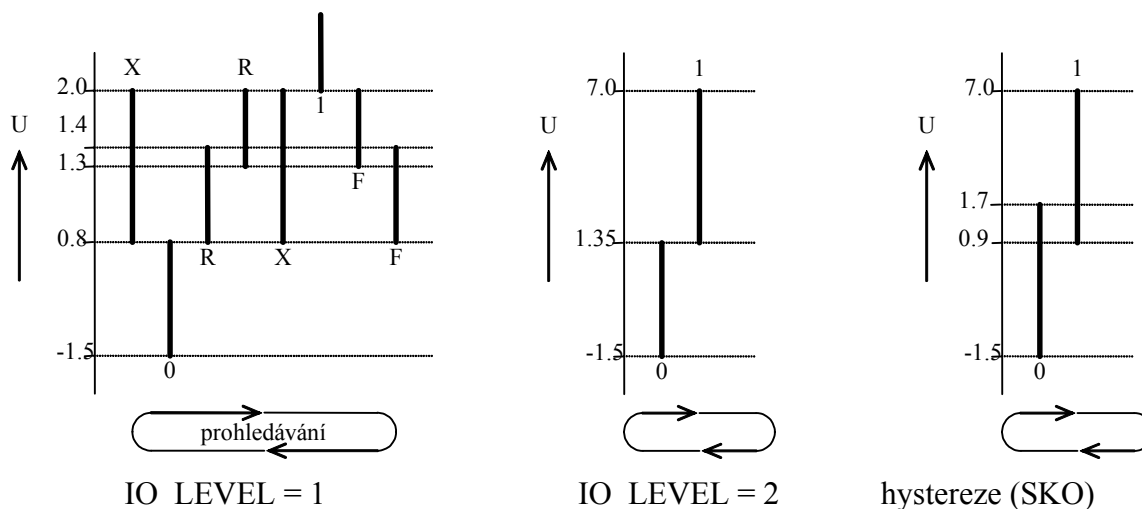
převodník). Pro všechny časové údaje existuje v modelu minimální, typická a maximální hodnota. Volba se provádí parametrem MNTYMXDLY, který může mít hodnoty:

- | | | | |
|----------|-------------------------------------------|----------|----------------------------------|
| 0 | - logický člen přebírá globální nastavení | 3 | - maximální hodnoty |
| 1 | - minimální hodnoty | 4 | - časování Worst Case (viz dále) |
| 2 | - typické hodnoty | | |

Standardně mají všechny prvky nastaveny hodnotu 0. Globální volba se provádí v simulačním profilu v záložce Options/Gate-level Simulation. Vstupně výstupní model umožňuje definici čtyř různých dvojic A/D a D/A převodníků. Volba se provádí parametrem IO_LEVEL:

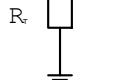
- | | | | |
|----------|---------------------------------------------------|----------|----------------|
| 0 | - logický člen přebírá globální nastavení | 3 | - $\equiv 1$ * |
| 1 | - A/D převodník produkuje symboly 0, 1, R, F, X | 4 | - $\equiv 2$ * |
| 2 | - zjednodušený A/D převodník jen pro úrovně 0 a 1 | | |

*) Úrovně 3 a 4 jsou prozatím totožné s úrovněmi 1 a 2. Pouze u knihoven prvků HC a HCT úrovně 1 a 2 používají D/A převodník pro pevné napětí 5V a teplotu 25°C. Úrovně 3 a 4 modelují výstupní obvod pro napájecí napětí v rozsahu 2 - 6V.



Obr. 5.14: Princip funkce A/D převodníku

V diagramech jsou vyznačeny napěťové intervaly pro jednotlivé logické úrovně při A/D převodu. Hodnoty jsou dány v knihovně pro každou logickou řadu. Při startu simulace se začíná od prvního intervalu, pokud v něm napětí neleží, tak se začnou prohledávat další intervaly v naznačeném směru. Pokud je hledání neúspěšné, zvolí se interval, který je napěťově nejbližší a simulátor vypíše varování. Pomocí překrývání intervalů je možné modelovat i hysterezi (používá se pro Schmittovy klopné obvody, např. 7414).



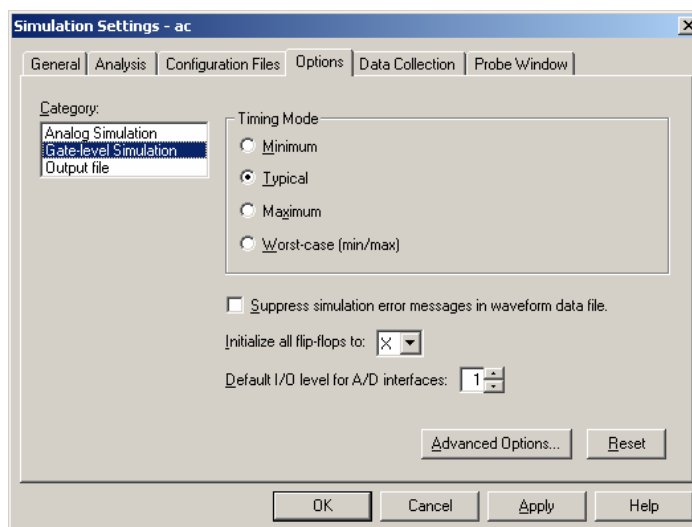
Obr. 5.15: Proměnné odpory D/A převodníku

Hodnoty R_L a R_H v D/A převodníku pro každý logický symbol jsou specifikovány v knihovně. Následující výpis je pro řadu TTL. Například pro logický signál 0 je ekvivalentní výstupní obvod tvořen napěťovým zdrojem 0.09V s vnitřním odporem 7 Ω .

```
.model DIN74 dinput (
+ s0name="0"      s0tsw=3.5ns   s0rlo=7.13    s0rhi=389    ;7ohm,    0.09v
+ s1name="1"      s1tsw=5.5ns   s1rlo=467     s1rhi=200    ;140ohm,  3.5v
+ s2name="X"      s2tsw=3.5ns   s2rlo=42.9    s2rhi=116    ;31.3ohm, 1.35v
+ s3name="R"      s3tsw=3.5ns   s3rlo=42.9    s3rhi=116    ;31.3ohm, 1.35v
+ s4name="F"      s4tsw=3.5ns   s4rlo=42.9    s4rhi=116    ;31.3ohm, 1.35v
+ s5name="Z"      s5tsw=3.5ns   s5rlo=200K    s5rhi=200K
+ )
```

Nastavení digitální části

V simulačním profilu v záložce Options/Gate-level Simulation lze nastavit globálně volbu I/O modelů a zpoždění a počáteční stav klopných obvodů.



Obr. 5.16: Nastavení pro simulaci číslicové části obvodu

Poznámky:

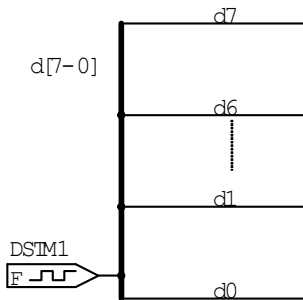
1. A/D převodník úrovně 1 se doporučuje použít tam, kde jsou logické obvody buzeny strmými impulsy. V případě pomalých přechodných dějů nebo pokud se vstupní napětí pohybuje v nerozhodné oblasti, je doporučeno použít model úrovně 2 nebo jinak dostaneme zbytečně pesimistické výsledky simulace (převodník bude produkovat stav X).
2. Detailní popis modelů digitálních prvků lze nalézt v referenční příručce.

5.7.3 Programování číslicových zdrojů

Číslicové zdroje slouží podobně jako analogové k buzení obvodu. Teoreticky bychom mohli použít pro generování vstupních signálů analogové zdroje, ale mezi zdroj a obvod by musel být zařazen A/D převodník. Číslicové zdroje navíc poskytují větší možnosti při definici průběhu signálu a mohou mít vícebitový výstup. Nacházejí se v knihovně *source*.

Připojení zdroje na sběrnici

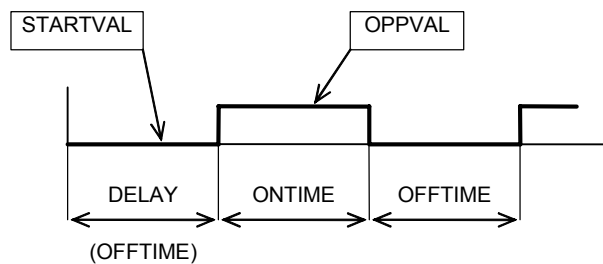
S výjimkou DigClock mají číslicové zdroje variabilní bitovou šířku výstupu. Mohou být připojeny k vodiči (1 bit) nebo na sběrnici (vícebitové). Při definici zdroje se zadávají hodnoty výstupu v určitém čase. Nejužívanější způsob zápisu je formát MSB ... LSB. Toto pořadí musí být zachováno i při definici jména sběrnice, které slouží jako šablona pro jména připojených vodičů. Např. pro osmibitovou datovou sběrnici definujeme jméno D[7-0], viz příklad na obrázku. Potom vodič D7 odpovídá MSB.



Obr. 5.17: Připojení zdroje na sběrnici

•DigClock

Jednabitový generátor hodinového signálu. V diagramu je vysvětlený význam parametrů zdroje.



Pokud není parametr DELAY definován použije se na počátku hodnota OFFTIME.

•FileStim

Vícebitový generátor jehož průběh je definován v externím souboru ve formě vzorků (tj. *čas - hodnota*). Soubor je možné definovat ručně nebo jej vytvořit uložením výsledků předchozí simulace (pseudosoučástka VECTOR). Parametry zdroje:

- FILENAME** Jméno souboru pro definici zdroje. Pokud se nachází v adresáři schématu nebo knihoven stačí udat jméno, jinak se musí uvést úplná cesta.
- SIGNAME** Nepovinný seznam jmen signálů definovaných v souboru, které budou použity (může jich být využito méně než je definováno). Pokud není tato položka definovaná, tak se jména signálů odvodí ze jmen vodičů připojené sběrnice. V případě definice odpovídá seznam pořadí a počtu jmen vodičů sběrnice.

Formát souboru:

```
[TIMESCALE = <value>]
<signame 1>...<signame n>...
OCT(<signame bit 3> ... <signame bit 0>) ...
HEX(<signame bit 4> ... <signame bit 0>) ...

<time > <value >
<time > <value >
```

V hlavičce souboru je uveden seznam jmen. Pokud hlavička pokračuje na dalším řádku nesmí se používat kontinuační „+“ jako v případě souboru s popisem obvodu. Každému jménu v hlavičce odpovídá jeden bit. Pořadí a jména musí souhlasit se sběrnici v případě, že není použit parametr **SIGNAME** v součástce. Signály můžeme sdružovat po trojicích nebo

čtveřicích (OCT nebo HEX). V tom případě se jejich hodnota udává jako osmičkové nebo šestnáctkové číslo. Mezi hlavičkou a samotnou definicí průběhu musí být alespoň jeden prázdný řádek.

Průběh signálů sběrnice je definován formou dvojic čas - hodnota ($\langle time \rangle \langle value \rangle$). Čas může být definován jako absolutní (př. 10ns, 1.2e-6) nebo relativní vzhledem k poslední události. Relativní hodnota začíná znakem „+“ (př. +10ns, +1.2e-6). Pokud je v hlavičce definován parametr TIMESCALE, pak jsou jeho hodnotou násobeny všechny časové údaje. Jestliže TIMESCALE = 10ns, tak časový údaj 10 představuje skutečný čas 100ns. Hodnota signálu se udává v pořadí deklarace v hlavičce. Povolené hodnoty jsou v následující tabulce.

formát:	binární	osmičkový (OCT)	šestnáctkový (HEX)
logická hodnota	0, 1	0 - 7	0 - F
stav X	X	X (všechny bity)	X (všechny bity)
stav Z	Z	Z (všechny bity)	Z (všechny bity)
stav R	R	R (všechny bity)	
stav F	F	F (všechny bity)	

Příklad 5.12 *Definiční soubor pro zdroj FileStim.*

```
Clock Reset In1 In2
HEX(Addr7 Addr6 Addr5 Addr4) HEX(Addr3 Addr2 Addr1 Addr0)
ReadWrite

0 0000 00 0 ; pro sdružení signálů se mohou požit mezery
10n 1100 4E 0
20n 0101 4E 1
30n 1110 4E 1
40n 0111 FF 0
```

Jestliže bychom nepoužili parametr SIGNAME, tak by jména vodičů připojené sběrnice musela souhlasit s hlavičkou. Parametr SIGNAME umožňuje vybrat jen některé signály s definičního souboru.

•DigStim

Vícebitový zdroj jehož časový průběh je graficky specifikován pomocí externího programu Stimulus Editor. Po dvojitým kliknutí na součástku zadáme jméno a otevře se Stimulus Editor. Jména definovaných signálů musí souhlasit se jmény vodičů sběrnice.

•Stim_n

Programovatelný vícebitový číslicový zdroj. V knihovně jsou definovány zdroje se šířkou 1, 4, 8 a 16 bitů (změnou atributu WIDTH lze šířku libovolně změnit). Zdroj STIM1 jde připojit k vodiči, ostatní musí být připojeny ke sběrnici.

WIDTH Počet výstupních bitů. Hodnotu lze změnit.

FORMAT Posloupnost číslic udávající formát v jakém budou definovány hodnoty signálů. Číslice mohou být 1 (binární), 3 (osmičková) a 4 (šestnáctková).

Jejich součet musí být roven parametru WIDTH. Příklad: WIDTH=8, FORMAT=11111111 - hodnoty se definují binárně (bez mezer) např. 10100011, WIDTH=8, FORMAT=11114 - první čtyři bity se definují binárně, zbylé pomocí šestnáctkových číslic (bez mezery) např. 1010B.

TIMESTEP Nepovinný parametr definující délku hodinového cyklu. Pokud je definován tak můžeme použít při udávání času příponu „c“. Hodnota se pak interpretuje jako násobek hodinového taktu. Příklad: TIMESTEP=100ns, časový údaj 10c znamená čas 1us.

COMMAND1- COMMAND16 - Příkazy pro definici časového průběhu. Na každém řádku může být jen jeden příkaz.

Formát příkazů:

```
< time > < value >
LABEL=< label name >
< time > GOTO < label name > < n > TIMES
< time > GOTO < label name > UNTIL GT < value >
< time > GOTO < label name > UNTIL GE < value >
< time > GOTO < label name > UNTIL LT < value >
< time > GOTO < label name > UNTIL LE < value >
< time > INCR BY < value >
< time > DECR BY < value >
REPEAT FOREVER
REPEAT < n > TIMES
ENDREPEAT
```

<time> Čas může být definován jako absolutní (př. 10ns, 1.2e-6) nebo relativní vzhledem k poslední události. Relativní hodnota začíná znakem „+“ (př. +10ns, +1.2e-6). Pokud je nastaven atribut TIMESTEP může se použít přípona „c“ pro udání času formou násobku hodinového cyklu. Hodnota času musí mít neklesající charakter.

<value> Hodnota signálů musí odpovídat vždy poli FORMAT. Tabulka udává povolené hodnoty.

formát:	binární	osmičkový (OCT)	šestnáctkový (HEX)
logická hodnota	0, 1	0 - 7	0 - F
stav X	X	X (všechny bity)	X (všechny bity)
stav Z	Z	Z (všechny bity)	Z (všechny bity)
stav R	R	R (všechny bity)	
stav F	F	F (všechny bity)	

LABEL=< label name > - Označení návěští pro definici skoku příkazem GOTO.

< n > Nastavení počtu opakování. Hodnota -1 znamená „stále“.

<time> GOTO <label name> UNTIL GT|GE|LT|LE <value> - Podmínečný skok pokud není splněna podmínka. Operátory GT ≡ >, GE ≡ ≥, LT ≡ <, LE ≡ ≤ srovnávají hodnotu signálů sběrnice (jako celek) s hodnotou uvedenou v příkazu.

INCR BY|DECR BY - Zvýšení nebo snížení hodnoty sběrnice o zadané číslo, které musí vyhovovat poli FORMAT.

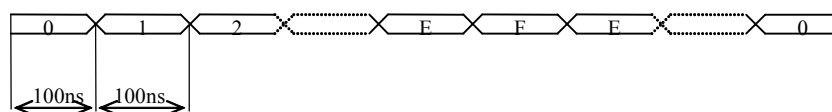
REPEAT EDREPEAT - Definice cyklu s pevným počtem opakování.

Poznámky:

1. Ve smyčce tvořené příkazem GOTO jsou časy automaticky překonvertovány na relativní. Skok je možný jen na návěští (*label*), které již bylo definováno.
2. Příkaz GOTO < *label name* > způsobí skok na příkaz který je uveden bezprostředně za deklarací návěští. Časový údaj u tohoto příkazu se neuvažuje.
3. Pro cykly s pevným počtem opakování se doporučuje použít konstrukci REPEAT/ENDREPEAT.

Příklad 5.13 *Nastavení zdroje STIM4*

Definice čtyřbitového zdroje který po 100ns zvyšuje svůj stav od 0 do F a nazpět a pak se zastaví na 0.



Použijeme zdroj STIM4. Parametr FORMAT změním z hodnoty 1111 na 4, tj. stav výstupu bude určen jednou šestnáctkovou číslicí. Požadovaný výstupní průběh se dosáhne pomocí cyklů, tabulka udává dvě možná řešení.

COMMAND	varianta 1	varianta 2
1	0s 0	0s 0
2	REPEAT 15 TIMES	LABEL=L1
3	+100ns INCR BY 1	+100ns INCR BY 1
4	ENDREPEAT	+100ns GOTO L1 UNTIL GE F
5	REPEAT 15 TIMES	LABEL=L2
6	+100ns DECR BY 1	+100ns DECR BY 1
7	ENDREPEAT	+100ns GOTO L1 UNTIL LE 0

5.7.4 Hlášení o chybových stavech během simulace

Simulace logických obvodů má smysl především v časové oblasti, u smíšených obvodů je ještě možná analýza DC. Při určování pracovního bodu (DC) se sice uvažují funkce logické části smíšeného obvodu, ale logické proměnné nejsou ukládány do výstupního souboru. Pokud není možné dosáhnout řešení v DC, tak se všechny logické výstupy nastaví na X a obvod se řeší jako analogový.

PSpice má propracovanou diagnostiku vnitřních chybových stavů v logických obvodech (hazardů). V obvodu mohou být použity konektory EXTERNAL_OUT nebo EXTERNAL_BI. Pokud hazardní stav dojde na tyto konektory nebo vstup sekvenčních obvodů (D, JK, SR, ROM, RAM) je o tom podáno hlášení. Simulátor identifikuje dva druhy problémů:

1. **Porušení časování** - omezení jsou typu:
WIDTH - minimální šířka impulsu,

SETUP - minimální doba ustálení signálu před příchodem hodinového pulsu,
 HOLD - minimální doba zachování stavu od příchodu hodinového impulsu

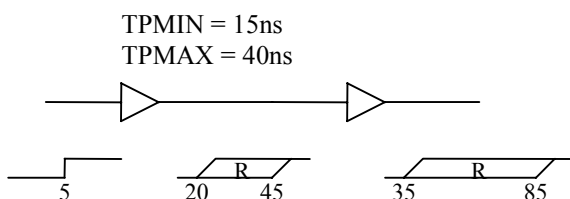
2. **Časovací hazard** - pokud se na vstupu logického obvodu setkají časově neurčité signály R, F (nejedná se o neurčitost úrovně X), takže může nebo nemusí dojít ke změně na jeho výstupu. Tato situace je označena jako **AMBIGUITY CONVERGENCE** (souběh neurčitostí). Jestliže taková neurčitost dorazí na konektor nebo změní stav sekvenčního obvodu je situace označena jako **PERSISTENT HAZARD** (trvalý hazard).

Hlášení o chybách se ukládají do datového souboru pro zobrazení v Probe. Zde mohou být zobrazena buď ihned po načtení datového souboru nebo volbou **Tools/Simulation Messages**.

5.7.5 Analýza číslicového obvodu v režimu Worst Case

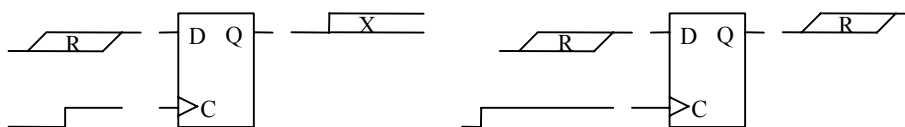
Pro logické členy výrobce udává vždy toleranční interval jejich parametrů. Analýza nejhoršího případu v digitální oblasti řeší situaci, že časové charakteristiky (např. zpoždění hradla) leží „někde uvnitř“ tolerančního intervalu. Každý logický prvek má definovanou minimální (MIN) a maximální (MAX) hodnotu časového parametru. Analýza je založena na principu šíření časových neurčitostí vyjádřených stavy R a F. Nedoporučuje se kombinovat analýzu Worst Case v analogové a digitální oblasti kvůli zásadní odlišnosti principů výpočtu.

Uvažme jednoduché opakovací se zpožděním TP stejným pro přechod $H \rightarrow L$ a $L \rightarrow H$ na výstupu (skutečné modely mají tyto časy rozdílné).



Obr. 5.18: Šíření neurčitosti

V čase 5ns nastane na vstupu přechod $0 \rightarrow 1$. Jestliže se skutečná hodnota zpoždění nachází někde uvnitř tolerančního intervalu, tak se na výstupu může objevit přechod v kterémkoliv okamžiku. Neurčitost R trvá od 20ns do 45ns. Po průchodu dalším hradlem se pásmo neurčitosti rozšíří.

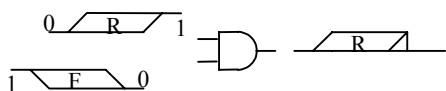


Obr. 5.19: Působení neurčitostí na sekvenční obvod

Jestliže je neurčitý stav R navzorkován znamená to stav X na výstupu. Pokud neurčitost přijde na hodinový vstup, tak se šíří dál (vstup D stabilní). V důsledku šíření neurčitostí vznikají různé kritické stavy (hazardy) o kterých informuje simulátor.

CONVERGENCE HAZARD

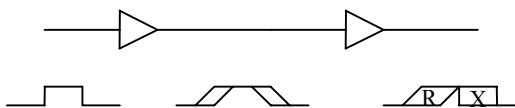
Tento hazard vzniká v důsledku překrytí časových neurčitostí na vstupu logického členu. Není možné určit přesně odezvu hradla. Uvažme situaci pro logický člen AND.



Obr. 5.20: Souběh neurčitostí

Na výstupu se může ale nemusí objevit impuls 0-1-0 (*glitch*). Tuto situaci simulátor řeší impulsem 0-R-0 případně 0-F-0 na výstupu a hlášením **AMBIGUITY CONVERGENCE**. Pokud takový záskit způsobí změnu stavu paměťového prvku, je o tom podáno hlášení. U jiných logických prvků může být reakce na souběh neurčitostí jiná.

CUMULATIVE AMBIGUITY HAZARD

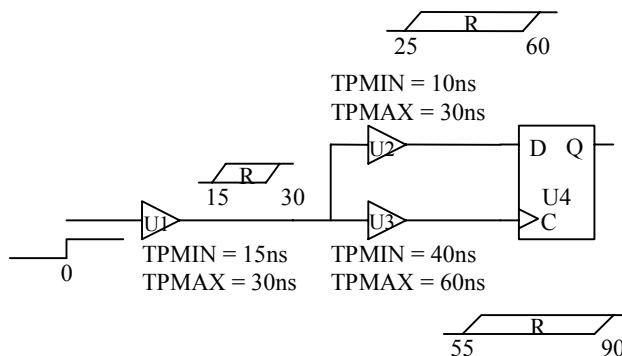


Obr. 5.21: Kumulace neurčitostí

Pokud se při průchodu krátkého impulsu obvodem překryjí neurčitosti nástupné a sestupné hrany, simulátor o tom podá hlášení. Pokud dojde k této situaci, znamená to, že simulátor není schopen určit polohu impulsu, i když reálný obvod může pracovat (tam jsou zpoždění pevná). Pokud je tento stav překážkou pokračování simulace, doporučuje se u prvků, které ho způsobují nastavit jiné časování než **Worst Case** pomocí parametru **MNTYMXDLY = 2**.

Simulátor rozeznává situaci, kdy signály se společným počátkem a s časovou neurčitostí dorazí na vstup logického členu.

RECONVERGENCE HAZARD



Obr. 5.22: Neurčitosti se společným základem

Na vstupu klopného obvodu došlo k překrytí intervalů neurčitosti. Jestliže ale porovnáme zpoždění členů U2 a U3, tak k této situaci u reálného obvodu nemůže dojít. Proto simulátor při určování hazardu neuvvažuje společnou neurčitost, která v příkladu vznikla na U1.

Potlačení krátkých impulsů (glitch suppression)

U reálných logických obvodů je nutné přivést na vstup jistou minimální energii aby se výstup změnil. Tato vlastnost je vyjádřena minimální délkou vstupního impulsu. V PSpice je situace ošetřena tak, že pokud na vstup logického členu přijde impuls kratší než aktuální nastavené zpoždění, tak dojde k jeho potlačení.

Příloha 1: Operátory, funkce a proměnné v PSpice

Numerické hodnoty je možné psát jako celá nebo desetinná čísla, či ve tvaru mantisy a exponentu. K číslu je možno přidat násobitel a fyzikální jednotku ve tvaru přípon. Fyzikální jednotka má pouze význam komentáře, který usnadňuje orientaci. Výraz musí být vždy uzavřen ve složených závorkách { }. Velikost písma nerozhoduje.

Násobitelé:

F	P	N	U	M	K	MEG	G	T	MIL
10^{-15}	10^{-12}	10^{-9}	10^{-6}	10^{-3}	10^3	10^6	10^9	10^{12}	$25.4 \cdot 10^{-6}$
femto	piko	nano	mikro	mili	kilo	mega	giga	tera	

Operátory:

Aritmetické		Relační (pro funkci IF)	
+	sčítání (nebo spojování řetězců)	==	test rovnosti
-	odčítání	!=	test nerovnosti
*	násobení	>	větší než
/	dělení	>=	menší nebo rovno
**	umocňování	<	menší než
Logické		<=	menší nebo rovno
~	NOT		
	OR		
^	XOR		
&	AND		

Funkce:

Funkce	Význam	Poznámka
ABS(x)	$ x $	
ACOS(x)	arkus kosinus	$-1.0 \leq x \leq +1.0$, výsledek v radiánech
ARCTAN(x)	arkus tangens	výsledek v radiánech
ASIN(x)	arkus sinus	$-1.0 \leq x \leq +1.0$
ATAN(x)	arkus tangens	výsledek v radiánech
ATAN2(y, x)	arkus tangens z (y/x)	výsledek v radiánech
COS(x)	$\cos(x)$	x je v radiánech
COSH(x)	hyperbolický kosinus	
DDT(x)	derivace x podle času	použitelné pouze v analýze Transient
EXP(x)	e^x	
IF(t, x, y)	vrací x pokud t = TRUE y pokud t = FALSE	t je logický výraz, x a y jsou číselné hodnoty nebo výrazy
IMG(x)	imaginární část x	vrací 0.0 pro reálná čísla
LIMIT(x, min, max)	vrací min pro $x < min$	

	x pro $min \leq x \leq max$ max pro $x > max$	
LOG(x)	ln(x)	přirozený logaritmus
LOG10(x)	log(x)	dekadický logaritmus
M(x)	absolutní hodnota x	stejně jako ABS(x).
MAX(x, y)	maximum z x a y	
MIN(x, y)	minimum z x a y	
P(x)	fáze komplexního čísla x	vrací 0.0 pro reálná čísla
PWR(x, y)	$ x ^y$ nebo $\{x^{**}y\}$	stejnou funkci má binární operátor**
PWRS(x, y)	sgn(x) $ x ^y$ (pro $x > 0$)	
R(x)	reálná část komplexního x	
SDT(x)	časový integrál x	použitelné pouze v analýze Transient
SGN(x)	-1 pro $x < 0$ 0 pro $x = 0$ 1 pro $x > 0$	
SIN(x)	sin(x)	x je v radiánech
SINH(x)	hyperbolický sinus	
STP(x)	1 pro $x > 0.0$, 0 pro $x \leq 0.0$	jednotkový skok
SQRT(x)	\sqrt{x}	
TAN(x)	tan(x)	x je v radiánech
TANH(x)	hyperbolický tangens	
TABLE($x_1, y_1, x_2, y_2, \dots, x_n, y_n$)	lineární interpolace $y = f(x)$	pro $x < x_1$ je $y = y_1$, pro $x > x_n$ je $y = y_n$

Systemové proměnné:

TEMP	Teplota obvodu. Pokud není nastaveno rozmítání teploty, tak TEMP = TNOM.
TIME	Čas v analýze Transient, v ostatních analýzách není definován. Může se použít jen ve výrazech pro funkční zdroje napětí a proudu.

Proměnné:

Proměnné jsou napětí a proudy v obvodu. Používají se pro specifikaci výstupu různých analýz a pro funkční zdroje (ABM).

Napětí:

v[AC suffix](*<out id>* [, *<out id>*])
v[terminal]* [AC suffix](*<out device>*)

Proud:

i[AC suffix](*<out device>* [:terminal])
i[terminal][AC suffix](*<out device>*)

<out id > úplné jméno uzlu nebo jméno pinu ve formátu *<out device>*:*<terminal>*

<out device > úplné jméno prvku

terminal určuje jméno vývodu pro vícepólové součástky (např. B - báze, E - emitor)

GaAs MESFET	D/G/S	JFET	D/G/S
MOSFET	D/G/S/B	bipolární transistor	C/B/E/S
IGBT	C/E/G	vedení	A/B

<i>AC Suffix</i>	přípony pro analýzu AC, kde jsou veličiny komplexní čísla (nic) nebo M modul (absolutní hodnota)
DB	modul v decibelech
P	fáze
R	reálná část
I	imaginární část
G	skupinové zpoždění

Příklady:

v(out)	napětí uzlu out proti referenci 0
v(D23:1)	napětí pinu 1 (anoda) diody D23
vdb(R1)	modul napětí v decibelech na rezistoru R1 (jen pro AC). Stejný význam má vdb(R1:1,R1:2).
vce(Q1)	napětí na kolektoru bipolárního transistoru Q1
i(R1)	proud rezistorem R1
ic(Q1)	kolektorový proud bipolárního transistoru Q1, stejný význam má i(Q1:C)

Příloha 2: Některé příkazy a analogové prvky jazyka (P)Spice

V následujícím seznamu jsou uvedeny jen příkazy se kterými se uživatel může setkat při manipulaci s textovými modely a nebo pro které grafické prostředí neumožňuje nastavení. Zcela je vynechán popis analýz. Parametry modelů jsou vypsány jen u těch prvků, kde uživatel může provést ruční definici. U polovodičů nejsou parametry uvedeny, neboť k pochopení jejich významu je nutné uvést vzorce modelu, což ale přesahuje rámec této příručky. Přesný popis lze nalézt v [1]. Navíc existuje program Model Editor pro komfortní generování některých modelů.

První řádek příkazového souboru se neinterpretuje, slouží jako nadpis. Pokud řádek začíná „*“ tak je uvažován jako komentářový. Pokud je na řádku středník, tak se zbytek neuvažuje (slouží k zápisu komentářů do příkazových řádků). Jestliže se nějaký příkaz nevejde na jeden řádek, tak může pokračovat na dalším, ale prvním znakem pokračovacího řádku musí být „+“. V PSpice může být jméno uzlu jakýkoliv řetězec. Uzel číslo 0 je vyhrazen jako referenční, každý obvod jej musí obsahovat. Malá a velká písmena se nerozlišují. Příkazy PSpice (Spice) vždy začínají tečkou, deklarace prvků písmenem vyhrazeným pro daný prvek (např. rezistor - R).

Vybrané příkazy:

.DISTRIBUTION Definice uživatelské funkce rozložení hustoty pravděpodobnosti

.DISTRIBUTION <name> (<deviation> <probability>)*

Příklad: .DISTRIBUTION bi_modal (-1,1) (-.5,1) (-.5,0) (.5,0) (.5,1) (1,1)

.END Ukončovací příkaz vstupního souboru

.FUNC Definice Funkce

.FUNC <name> ([arg]*) {<body>}

Příklad: .FUNC MIN3(A,B,C) {MIN(A,MIN(B,C))}

.INC <"file name"> Vložení externího souboru

.LIB ["file name"] Vložení knihovny

.MODEL Definice modelu prvku

.MODEL <model name> [AKO: <reference model name>] <model type>

+ ([<parameter name> = <value> [tolerance specification]])*

+ [T_MEASURED=<value>] [[T_ABS=<value>]

+ [T_REL_GLOBAL=<value>] or [T_REL_LOCAL=<value>]]

[tolerance spec.]: [DEV [track & dist] <value>[%]] [LOT [track & dist] <value>[%]]

[track & dist] [/<lot #>][/<distribution name>]

Příklady: .MODEL DLOAD D (IS=1E-9 DEV .5% LOT 10%)

 .MODEL RTRACK RES (R=1 DEV/GAUSS 1% LOT/UNIFORM 5%)

.SUBCKT, .ENDS Definice makroobvodu

.SUBCKT <name> [node]* [OPTIONAL: <<interface node> = <default value> >*]
 + [PARAMS: <<name> = <value> >*] [TEXT: <<name> = <text value> >*]

...

.ENDS [subcircuit name]

Makroobvod je definován v bloku mezi příkazy .SUBCKT a .ENDS. Uvnitř může být deklarace součástek a příkazy .IC, .NODESET, .MODEL, .PARAM a .FUNC. Jména uzlů a případně deklarované modely mají lokální charakter. V bloku se může volat jiný Makroobvod (X) ale jeho definice uvnitř není dovolena. Uzel 0 je globální a pokud je použit v bloku tak dojde k propojení s hlavním referenčním uzlem obvodu.

Příklad: .SUBCKT RC_PAR 1 2 PARAMS: r = 1MEG c = 1PF
 R1 1 2 {r}
 C1 1 2 {c}
 .ENDS

Deklarace vybraných analogových prvků:**C Kapacitor**

C<name> <(+) node> <(-) node> [model name] <value> [IC=<initial value>]

Příklady: CLOAD 15 0 20pF
 CFDBCK 3 33 CMOD 10pF

Model: CAP

Parametr	Význam	Jednotky	Implicitní hodnota
C	násobitel kapacity	-	1
VC1	lineární napěťový koeficient	V ⁻¹	0
VC2	kvadratický napěťový koeficient	V ⁻²	0
TC1	lineární teplotní koeficient	K ⁻¹	0
TC2	kvadratický teplotní koeficient	K ⁻²	0

Pokud je model použit, tak

$$kapacita = \langle value \rangle \cdot C \cdot (1 + VC1 \cdot V + VC2 \cdot V^2) \cdot (1 + TC1 \cdot (T - T_{nom}) + TC2 \cdot (T - T_{nom})^2)$$

E Zdroj napětí řízený napětím

- E<name> <(+) out> <(-) out> <(+) in> <(-) in> <gain>
- E<name> <(+) out> <(-) out> POLY(<value>) <(+) in> <(-) in> >*

 + <<polynomial coefficient value> >*
- E<name> <(+) out> <(-) out> VALUE = { <expression> }
- E<name> <(+) out> <(-) out> TABLE { <expression> } = <<in. value>, <out. value> >*
- E<name> <(+) out> <(-) out> LAPLACE { <expression> } = { <transform> }
- E<name> <(+) out> <(-) out> FREQ { <expression> } = [KEYWORD]
 + <<frequency>, <magnitude>, <phase> >* [DELAY = <delay>]
- E<name> <(+) out> <(-) out> CHEBYSHEV { <expression> } =
 + <[LP] [HP] [BP] [BR]>, <cutoff frequencies>*, <attenuation>*

Příklady: EBUFF 1 2 10 11 1.0
 EAMP 13 0 POLY(1) 26 0 0 500


```

ESQROOT 5 0 VALUE = {5V*SQRT(V(3,2))}
ET2 2 0 TABLE {V(ANODE,CATHODE)} = (0,0) (30,1)
ERC 5 0 LAPLACE {V(10)} = {1/(1+.001*s)}
ELOWPASS 5 0 FREQ {V(10)}=(0,0,0)(5kHz, 0,0)(6kHz -60, 0) DELAY=3.2ms
ELOWPASS 5 0 CHEBYSHEV {V(10)} = LP 800 1.2K .1dB 50dB

```

Zdroj má sedm různých definic:

1. - lineární zdroj, parametr je zesílení (gain)
2. - polynomický zdroj
3. - funkční zdroj, výstupní napětí je dáno výrazem
4. - hodnota výstupu je dána tabulkou s max. 2048 vstupně/výstupními páry
5. - frekvenční vlastnosti jsou popsány přenosovou funkcí s Laplaceovým operátorem s
6. - frekvenční vlastnosti jsou popsány tabulkou

DELAY- zpoždění které je přidáno k fázi v tabulce aby byl zdroj kauzální

KEYWORD:MAG tabulka obsahuje modul přenosu

DB tabulka obsahuje modul přenosu v decibelech (implicitní)

RAD fáze je v radiánech

DEG fáze je ve stupních (implicitní)

R_I místo modulu a fáze je reálná a imaginární část přenosu

7. - Čebyševovy filtry

Typ

Definice

dolní propust LP (FPass FStop) RIPPLE STOP

horní propust HP (FPass FStop) RIPPLE STOP

pásmová propust BP (F0 F1 F2 F3) RIPPLE STOP

pásmová zádrž BR (F0 F1 F2 F3) RIPPLE STOP

F Zdroj proudu řízený proudem

F<name> <(+) out> <(-) out> <controlling V device name> <gain>

F<name> <(+) out> <(-) out> POLY(<value>) <controlling V device>* <polynomial coeffs> >*

Zdroj je řízen proudem přes nezávislý napěťový zdroj (V), na který se odkážeme jménem.

G Zdroj proudu řízený napětím

G<name> <(+) out> <(-) out> <(+) in> <(-) in> <transconductance>

G<name> <(+) out> <(-) out> POLY(<value>) <(+) in> <(-) in> >*
+ <polynomial coefficient value>*

G<name> <(+) out> <(-) out> VALUE = { <expression> }

G<name> <(+) out> <(-) out> TABLE { <expression> } = <(+) in>, <(-) in> >*

G<name> <(+) out> <(-) out> LAPLACE { <expression> } = { <transform> }

G<name> <(+) out> <(-) out> FREQ { <expression> } = [KEYWORD]
+ <frequency>, <magnitude>, <phase> >* [DELAY = <delay>]

G<name> <(+) out> <(-) out> CHEBYSHEV { <expression> } =
+ <type>, <cutoff frequencies>*, <attenuation>*

Deklarace je stejná jako u ZN/N (E).

H Zdroj napětí řízený proudem

H<name> <(+) out > <(-) out > <controlling V device name> <transresistance>

H<name> <(+) out > <(-) out > POLY(<value>) <controlling V device>* <polynomial coeffs>*

I Nezávislý proudový zdroj

I<name> <(+) out > <(-) out > [[DC] <value>] [AC <magnitude > [phase]]
+ [STIMULUS=<stimulus name>] [transient specification]

Příklady: IBIAS 13 0 2.3mA
IAC 2 3 AC .001
I3 26 77 DC .002 AC 1 SIN(.002 .002 1.5MEG)

[transient specification] definice chování v časové oblasti

EXP (<parameters>) exponenciální puls

PULSE (<parameters>) lichoběžníkový puls

PWL (<parameters>) po částech lineární signál

SFFM (<parameters>) FM signál

SIN (<parameters>) sinusový signál

Exponenciální puls EXP (<i1> <i2> <td1> <tc1> <td2> <tc2>)

<i1> počáteční proud

<i2> špičkový proud

<td1> počáteční doba s hodnotou <i1>

<tc1> časová konstanta náběhu

<td2> doba ukončení náběhu

<tc2> časová konstanta sestupu

Čas

Hodnota

0 až <td1>

$i1$

<td1> až <td2>

$i1 + (i2 - i1) \cdot (1 - e^{-(\text{TIME} - td1)/tc1})$

<td2> až TSTOP

$i1 + (i2 - i1) \cdot ((1 - e^{-(\text{TIME} - td1)/tc1}) - (1 - e^{-(\text{TIME} - td2)/tc2}))$

Lichoběžníkový puls PULSE (<i1> <i2> <td> <tr> <tf> <pw> <per>)

<i1> počáteční proud

<i2> pulsní proud

<td> počáteční doba s hodnotou <i1>

<tr> doba náběžné hrany

<tf> doba sestupné hrany

<pw> šířka pulsu s hodnotou <i2>

<per> perioda

Po částech lineární signál PWL [TIME_SCALE_FACTOR=<value>]
+ [VALUE_SCALE_FACTOR=<value>] (corner_points)*

(*corner_points*): - Určení vzorků mezi kterými je prováděna lineární interpolace.
 (<*tn*>, <*in*>) Definice zlomu. Čas *tn* může být absolutní nebo relativní vzhledem k poslední události (př. +10ms). Hodnota *in* může být dána vzorcem.
 FILE <*filename*> Soubor s dvojicemi (<*tn*>, <*in*>) na každém řádku.
 REPEAT FOR <*n*> (*corner_points*)* ENDREPEAT - Opakovat část průběhu <*n*>-krát, čas je relativně vzhledem k začátku cyklu.
n = -1 - opakuj stále *n* = 0 - ignoruje se
 REPEAT FOREVER (*corner_points*)* ENDREPEAT - Nekonečný cyklus.

FM signál SFFM (<*ioff*> <*iamp*> <*fc*> <*mod*> <*fm*>)
 <*ioff*> ss složka proudu
 <*iamp*> amplituda nosné
 <*fc*> frekvence nosné
 <*mod*> modulační index
 <*fm*> frekvence modulovaného signálu

$$i(t) = \langle ioff \rangle + \langle iamp \rangle \cdot \sin(2\pi \cdot \langle fc \rangle \cdot \text{TIME} + \langle mod \rangle \cdot \sin(2\pi \cdot \langle fm \rangle \cdot \text{TIME}))$$

Sinusový signál SIN (<*voff*> <*vamp*> <*freq*> <*td*> <*df*> <*phase*>)
 <*ioff*> ss složka proudu
 <*iamp*> amplituda
 <*freq*> frekvence
 <*td*> časový posun
 <*df*> exponent útlumu
 <*phase*> fáze

<u>Čas</u>	<u>Hodnota</u>
0 až < <i>td</i> >	$ioff + iamp \cdot \sin(2\pi \cdot phase/360^\circ)$
< <i>td</i> > až TSTOP	$ioff + iamp \cdot \sin(2\pi \cdot (freq \cdot (\text{TIME} - td) + phase/360^\circ)) \cdot e^{-(\text{TIME} - td) \cdot df}$

K Definice vazeb induktorů a vedení

1. K<*name*> L<*inductor name*> < L<*inductor name*> >* <*coupling value*>
2. K<*name*> < L<*inductor name*> >* <*coupling value*> <*model name*> [*size value*]
3. K<*name*>T<*transmission line name*>T<*transmission line name*>
 + Cm = <*capacitive coupling*> Lm = <*inductive coupling*>

Příklady: KTUNED L3OUT L4IN .8
 KXFRM L1 L2 L3 L4 .98 KPOT_3C8
 K2LINES T1 T2 Lm=1m Cm=.5p

Model: CORE

Existují tři definice:

1. Lineární vazba mezi induktory, činitel vazby musí být v intervalu [0, 1].

2. Nelineární magnetická vazba (magnetický materiál je popsán modelem). Hodnoty v deklaraci induktorů se interpretují jako počet závitů !
3. Vazba dvou vedení (T). Cm - kapacitní vazba [F/m], Lm - induktivní vazba [H/m]

L Induktor

L<name> <(+) node> <(-) node> [model name] <value> [IC = <initial value>]

Příklady: LLOAD 15 0 20mH
LCHOKE 3 42 LMOD .03

Model: IND

<u>Parametr</u>	<u>Význam</u>	<u>Jednotky</u>	<u>Implicitní hodnota</u>
L	násobitel indukčnosti	-	1
IL1	lineární proudový koeficient	A ⁻¹	0
IL2	kvadratický proudový koeficient	A ⁻²	0
TC1	lineární teplotní koeficient	K ⁻¹	0
TC2	kvadratický teplotní koeficient	K ⁻²	0

Pokud je model použit, tak

$$\text{indukčnost} = \langle \text{value} \rangle \cdot L \cdot (1 + \text{IL1} \cdot I + \text{IL2} \cdot I^2) \cdot (1 + \text{TC1} \cdot (T - T_{\text{nom}}) + \text{TC2} \cdot (T - T_{\text{nom}})^2).$$

R Rezistor

R<name> <(+) node> <(-) node> [model name] <value> [TC = <TC1> [, <TC2>]]

Příklady: RLOAD 15 0 2K
RFDBCK 3 33 RMOD 10K

Model: RES

<u>Parametr</u>	<u>Význam</u>	<u>Jednotky</u>	<u>Implicitní hodnota</u>
R	násobitel odporu	-	1
TC1	lineární teplotní koeficient	K ⁻¹	0
TC2	kvadratický teplotní koeficient	K ⁻²	0
TCE	teplotní exponent	% K ⁻¹	0

Pokud TCE není deklarován: $\text{odpor} = \langle \text{value} \rangle \cdot R \cdot (1 + \text{TC1} \cdot (T - T_{\text{nom}}) + \text{TC2} \cdot (T - T_{\text{nom}})^2)$

Pokud TCE je deklarován: $\text{odpor} = \langle \text{value} \rangle \cdot R \cdot 1.01^{\text{TCE} \cdot (T - T_{\text{nom}})}$

S Spínač řízený napětím

S<name> <(+) switch > <(-) switch > <(+) in> <(-) in> <model name>

Příklad: S12 13 17 2 0 SMOD

Model: VSWITCH

<u>Parametr</u>	<u>Význam</u>	<u>Jednotky</u>	<u>Implicitní hodnota</u>
RON	odpor spínače v sepnutém stavu	Ω	1.0
ROFF	odpor spínače v rozepnutém stavu	Ω	10 ⁶
VON	napěťová hladina sepnutí	V	1.0
VOFF	napěťová hladina rozepnutí	V	0.0

Pro správnou funkci by měly být plněny nerovnosti:

$$0 < \text{RON}, \text{ROFF} < 1/\text{GMIN} \quad (\text{GMIN je globální parametr simulátoru})$$

$$\text{ROFF/RON} < 10^{12}$$

$$|\text{VON-VOFF}| > \text{RELTOL} \cdot (\text{MAX}(|\text{VON}|, |\text{VOFF}|)) + \text{VNTOL}$$

VON > VOFF: pokud $V_c > \text{VON}$ pak $R_s = \text{RON}$, pokud $V_c < \text{VOFF}$ pak $R_s = \text{ROFF}$
 VON < VOFF: pokud $V_c < \text{VON}$ pak $R_s = \text{RON}$, pokud $V_c > \text{VOFF}$ pak $R_s = \text{ROFF}$

T Vedení

Ideální vedení

T<name> <A port (+)> <A port (-)> <B port (+)> <B port (-)> [model name]
 + Z0 = <value> [TD = <value>] [F = <value> [NL = <value>]]
 + IC = <near voltage> <near current> <far voltage> <far current>

Ztrátové vedení

T<name> <A port (+)> <A port (-)> <B port (+)> <B port (-)>
 + [<model name> [electrical length value]]
 + LEN = <value> R = <value> L = <value> G = <value> C = <value>

Příklady: T1 1 2 3 4 Z0=220 TD=115ns
 T4 1 2 3 4 LEN=1 R=.311 L=.378u G=6.27u C=67.3p

Model:	.MODEL <model name> TRN [model parameters]		
Parametr	Význam	Jednotky	Implicitní hodnota
<u>Ideální vedení</u>			
Z0	charakteristická impedance	Ω	-
TD	zpoždění	s	-
F	frekvence pro určení NL	Hz	-
NL	relativní vlnová délka	-	0.25
<u>Ztrátové vedení</u>			
R	měrný odpor	Ω /jednotka délky	-
L	měrná indukčnost	H/jednotka délky	-
G	měrná vodivost	S/jednotka délky	-
C	měrná kapacita	F/jednotka délky	-
LEN	elektrická délka	souhlasí s R,L,G,C	-

IC nastavení počátečních podmínek, musí se uvést všechny čtyři hodnoty

Ideální vedení může být specifikováno buď pomocí TD nebo F a NL. Všechny parametry vedení mohou být určeny výrazem, navíc R a G je možné popsat Laplaceovým výrazem.

V Nezávislý napěťový zdroj

V<name> <(+) node> <(-) node> [[DC] <value>] [AC <magnitude value> [phase value]]
 + [STIMULUS = <stimulus name>] [transient specification]

Parametry jsou stejné jako u nezávislého proudového zdroje (I).

W Proudem řízený spínač

W<name> <(+) switch > <(-) switch > <controlling V device name> <model name>

Spínač je řízen proudem přes napěťový zdroj.

Příklad: W12 13 17 VC WMOD

Model: .MODEL <model name> ISWITCH [*model parameters*]

<u>Parametr</u>	<u>Význam</u>	<u>Jednotky</u>	<u>Implicitní hodnota</u>
RON	odpor spínače v sepnutém stavu	Ω	1.0
ROFF	odpor spínače v rozepnutém stavu	Ω	10^6
ION	proudová hladina sepnutí	A	10^{-3}
IOFF	proudová hladina rozepnutí	A	0.0

Pro správnou funkci by měly být plněny nerovnosti:

$$0 < \text{RON}, \text{ROFF} < 1/\text{GMIN}$$

$$\text{ROFF}/\text{RON} < 10^{12}$$

$$|\text{ION}-\text{IOFF}| > \text{RELTOL} \cdot (\text{MAX}(|\text{ION}|, |\text{IOFF}|)) + \text{ABSTOL}$$

ION > IOFF: pokud $I_c > \text{ION}$ pak $R_s = \text{RON}$, pokud $I_c < \text{IOFF}$ pak $R_s = \text{ROFF}$

ION < IOFF: pokud $I_c < \text{ION}$ pak $R_s = \text{RON}$, pokud $I_c > \text{IOFF}$ pak $R_s = \text{ROFF}$

X Volání Makroobvodu

X<name> [*node*]* <*subcircuit name*> [PARAMS: <<name> = <value>>*]
+ [TEXT: <<name> = <text value> >*]

Příklad: X12 100 101 200 201 DIFFAMP

Detailní popis modelů polovodičových prvků je možné nalézt v referenční příručce programu PSpice [1].

prvek	deklarace	model
tranzistor GaAsFET	B	GASFET
dioda	D	D
tranzistor JFET	J	PJF, NJF
tranzistor MOSFET	M	NMOS, PMOS
bipolární transistor	Q	NPN, PNP
tranzistor IGBT	Z	NIGBT

Příloha 3: Operátory, funkce a proměnné v postprocesoru

Analogová část

Analogové proměnné specifikujeme stejně jako v PSpice (Příloha 1) s následujícími výjimkami:

- Specifikace V(<jméno>) a Vxy(<jméno>) nejsou přípustné, Vx(<jméno>) **můžeme** použít.
- Sum specifikujeme jako V(ONoise) a V(INoise).
- Rozmítaná veličina je přístupná pro použití ve výrazech.
- Existující křivky mohou být použité jako proměnné v rámci dané osy Y. Formát je #<pořadí>. Příklad: Existuje první křivka v(1), ve stejné ose můžeme nakreslit její pětinasobek výrazem #1*5.
- V případě vícenásobné analýzy (krokování) představuje proměnná všechny běhy současně. Dílčí běh je možné adresovat pomocí @n, kde n je pořadí běhu. Příklad: v(1)@5.
- Přípony mega a mili se značí jako M a m.

Analogové funkce:

Funkce	Význam	Funkce	Význam
ABS(x)	$ x $	SGN(x)	+1 (x>0), 0 (x=0), -1 (x<0)
SQRT(x)	\sqrt{x}	EXP(x)	e^x
LOG(x)	$\ln(x)$ (základ e)	LOG10(x)	$\log(x)$ (základ 10)
M(x)	modul x	P(x)	fáze x (ve stupních)
R(x)	reálná část x	IMG(x)	imaginární část x
G(x)	skupinové zpoždění x	PWR(x, y)	$ x ^y$
SIN(x)	$\sin(x)$ (x v radiánech)	COS(x)	$\cos(x)$ (x v radiánech)
TAN(x)	$\tan(x)$ (x v radiánech)	ATAN(x)	$\arctan(x)$ (výsledek v rad.)
ARCTAN(x)	$\arctan(x)$	DB(x)	modul x v decibelech
MIN(x)	minimum reálné části x	MAX(x)	maximum reálné části x
D(y)	$\frac{dy}{dx}$, x - proměnná osy X	S(y)	$\int_0^x y(\tau) d\tau$ - tekoucí integrál
AVG(y)	$\frac{1}{x} \int_0^x y(\tau) d\tau$ - tekoucí průměr	AVGX(y, d)	$\frac{1}{d} \int_{x-d}^x y(\tau) d\tau$ - tekoucí průměr v okně
RMS(y)	$\sqrt{\frac{1}{x} \int_0^x y(\tau)^2 d\tau}$ - kvadratický průměr		

Digitální část

Formát pro definici číslicového průběhu:

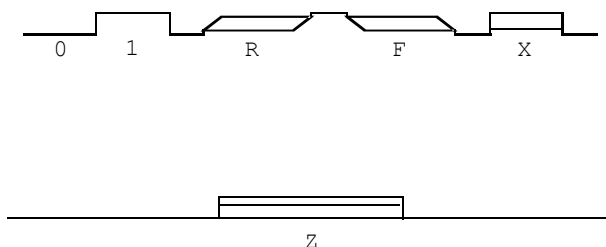
$\langle digital\ trace\ expression \rangle [;[\langle display\ name \rangle][;\langle radix \rangle]]$

$\langle display\ name \rangle$ Jméno které bude zobrazeno pro křivku. Pokud se neuvede tak je zobrazen přímo výraz

$\langle radix \rangle$ Specifikace číselného formátu pro sběrnice, H nebo X - hexadecimální, D - desítkový, O - osmičkový, B - binární.

$\langle digital\ trace\ expression \rangle$ - Výraz pro číslicový průběh. Může to být jednotlivý signál nebo sběrnice.

Signál: Číslicová proměnná se volá pouze jménem uzlu. Použitelné operátory a konstanty jsou v tabulce. Obrázek udává zobrazení jednotlivých logických stavů v grafu.



Sběrnice: Sběrnice je deklarována výčtem signálů v pořadí MSB...LSB uzavřeným do složených závorek. Příklad: { DATA4 DATA3 DATA2 DATA1 DATA0 }.

Pro specifikaci výčtu můžeme užít zkrácený zápis: $sss[xx : yy]$, kde

sss je alfanumerický řetězec

xx, yy jsou desítková čísla

Například {DATA[4:0]} je ekvivalentní s { DATA4 DATA3 DATA2 DATA1 DATA0 }.

Ve výrazech můžeme použít i sběrnice spolu s logickými operátory (operace bit-po-bitu) a aritmetickými operátory (operace s hodnotou sběrnice). Výsledek operace mezi sběrnici a signálem je sběrnice.

Logické a aritmetické operátory:

Logické operátory		Aritmetické operátory (jen pro sběrnice)	
()	asociace	*	násobení
~	NOT	/	dělení
&	AND	+	sčítání
^	XOR	-	odčítání
	OR		

Signálové konstanty: '0, '1, 'R, 'F, 'X, 'Z

Sběrníkové konstanty: formát: *r'ddd*, kde *r* je určení numerického formátu (x, h, d, o nebo b) a *ddd* je řetězec číslic příslušného formátu.

Příklady: x'3FFFF, h'5a (hexadecimální), d'79 (desítková), o'177400 (osmičková), b'100110, (binární)

Příloha 4: Popis některých prvků z knihoven

Knihovny schématických značek obsahují kromě modelů průmyslově vyráběných prvků, které většinou nemají žádné nastavitelné parametry také různé pasivní prvky, zdroje a funkční bloky které musí uživatel nastavit prostřednictvím atributů. Dále jsou popsány jen ty prvky, které nebyly vysvětleny v textu příručky. Obsah dílčích knihoven prvků si můžeme prohlédnout v Draw/Get New Part/Libraries.

Funkční bloky (knihovna ABM)

Funkční bloky slouží k simulaci blokových schémat. Jejich napěťové vstupy a výstupy se předpokládají proti referenčnímu uzlu. V následujícím popisu jsou vždy v závorce uvedeny atributy funkčního bloku.

Funkční bloky se vstupy a výstupy proti referenčnímu uzlu	
CONST (VALUE)	konstanta
SUM	sčítací blok
MULT	násobící blok
GAIN (GAIN)	zesilovač
DIFF	odčítací blok
LIMIT (LO, HI)	tvrdý omezovač, přenos=1
GLIMIT (LO, HI, GAIN)	zesilující omezovač, LO a HI jsou úrovně na výstupu
SOFTLIM (LO, HI, GAIN)	měkký omezovač, LO a HI jsou úrovně na výstupu
HILO	měkký omezovač
LOPASS (FP, FS, RIPPLE, STOP)	Čebyševova dolní propust
HIPASS (FP, FS, RIPPLE, STOP)	Čebyševova horní propust
BANDPASS (F0, F1, F2, F3, RIPPLE, STOP)	Čebyševova pásmová propust
BANDREJ (F0, F1, F2, F3, RIPPLE, STOP)	Čebyševova pásmová zadrž
INTEG (GAIN, IC)	integrátor, IC je počáteční podmínka
DIFFER (GAIN)	derivátor
TABLE (ROW1...ROW5)	převodní funkce (ss) je dána tabulkou
FTABLE (ROW1...ROW5)	frekvenční přenos daný tabulkou
LAPLACE (NUM, DENOM)	přenosová funkce je určena Laplaceovým výrazem, NUM a DENOM je čitatel a jmenovatel
ABS, SQRT, PWR (EXP), PWRS (EXP), LOG, LOG10, EXP, SIN, COS, TAN, ATAN, ARCTAN	bloky realizující základní matematické funkce
ABM	nezávislý zdroj, výstup je dán výrazem
ABM _i ABM _i /I	Bloky s $i = 1..3$ vstupy s přenosem určeným vzorcem. Pomocí V(%IN), V(%IN1),... se odkazujeme

	na napětí jednotlivých vstupů. Bloky ABM/I mají proudový výstup.
Funkční bloky s plovoucími vstupy a výstupy (jsou založeny na ZN/N (E) a ZP/N (G))	
EVALUE, GVALUE (EXPR)	ZN/N resp. ZP/N s přenosem určeným výrazem
ESUM, GSUM	sumační ZN/N resp. ZP/N se dvěma vstupy
EMULT, GMULT	násobící ZN/N resp. ZP/N se dvěma vstupy
ETABLE, GTABLE (EXPR, TABLE)	tabulkou definovaný ZN/N resp. ZP/N
EFREQ, GFREQ (EXPR, TABLE)	přenosová funkce ZN/N resp. ZP/N je definovaná tabulkou
ELAPLACE, GLAPLCE (EXPR, XFORM)	Laplaceovy ZN/N resp. ZP/N

Základní lineární řízené zdroje

E (ZN/N), G (ZP/N), F(ZP/P), H(ZN/P) všechny mají parametr GAIN

Číslicové zdroje (kapitola 5.7.3)

DigClock	programovatelný zdroj hodin
DigStim	zdroj pro definici v programu Stimulus Editor
FileStim	časový průběh je definován v souboru
STIM <i>i</i>	programovatelné <i>i</i> -bitové zdroje
Hi, Lo, X	číslicové konstanty

Nezávislé zdroje

V PSpice existuje pouze jeden nezávislý napěťový (V) a jeden proudový (I) zdroj. Jejich parametry jsou identické, příloha C - zdroje V a I.

Nezávislý zdroj má tři skupiny parametrů: stejnosměrné (DC), střídavé (AC) a časové. Následující tabulka udává jak se tyto parametry uplatní v základních analýzách (DC, AC, Transient). V závorkách v prvním sloupci jsou uvedena jména atributů u příslušných schematických značek.

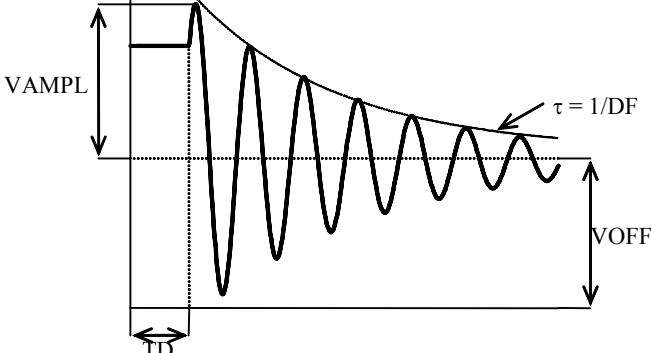
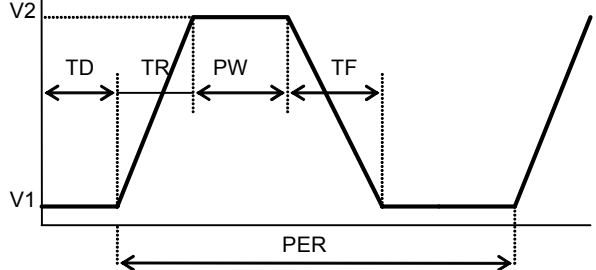
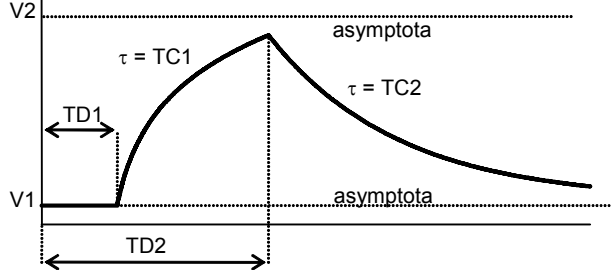
parametry zdrojů	Analýzy		
	DC	AC	Transient
stejnosemné (DC)	Stejnosemné napětí nebo proud. Pokud DC specifikace chybí, odvodí se z časové složky.	Pro výpočet stejnosemného prac. bodu platí totéž co pro DC	Stejnosemné napětí nebo proud. Pokud je definován časový průběh, má přednost.
střídavé (AC,ACMAG, ACPHASE)	neuplatní se	Amplituda a fáze komplexního fázoru	neuplatní se
časové	Uplatní se pro určení ss složky pokud chybí DC parametr.	Pro výpočet stejnosemného prac. bodu platí totéž co pro DC	Časový průběh. Pokud jeho specifikace chybí uplatní se DC složka.



Parametr DC není stejnosemná složka časového průběhu.

V knihovně je několik různých typů nezávislých zdrojů s možností nastavit pouze některé parametry obecných zdrojů V a I. V tabulce je jejich přehled s vysvětlením významu atributů. Atributy AC a DC mohou zůstat prázdné, v tom případě se uvažuje nulová hodnota.

ISRC VSRC	<i>Univerzální zdroj</i> Umožňuje nastavit všechny parametry.	
IDC VDC	<i>Stejnosemný zdroj</i> DC - stejnosemné napětí nebo proud, uplatní se v DC, AC (pracovní bod) i Transient	
IAC VAC	<i>Střídavý (fázorový) zdroj</i> DC - stejnosemná složka, uplatní se v DC, AC (pracovní bod) i Transient ACMAG, ACPHASE - amplituda a fáze fázoru, jen pro analýzu AC	
ISIN VSIN	<i>Sinusový zdroj</i> DC - stejnosemná složka, uplatní se v DC a AC (pracovní bod) AC - amplituda fázoru, jen pro analýzu AC časové parametry: VOFF- stejnosemný posuv, VAMPL - amplituda, FREQ – frekvence, PHASE - počáteční fáze, TD - počáteční zpoždění, DF - činitel tlumení	

	 <p> $0 \leq t < TD: v = voff + v ampl \cdot \sin(2\pi \cdot \text{phase} / 360^\circ)$ $TD \leq t \quad v = voff + v ampl \cdot \sin(2\pi \cdot (\text{freq} \cdot (t - td) + \text{phase} / 360^\circ)) \cdot e^{-(\text{TIME} - td) \cdot df}$ </p>
<p>IPULSE VPULSE</p>	<p><i>Pulzní zdroj</i> DC - stejnosměrná složka, uplatní se v DC a AC (pracovní bod) AC - amplituda fázoru, jen pro analýzu AC časové parametry: V1, V2, TD, TR, TF, PW, PER</p> 
<p>IEXP VEXP</p>	<p><i>Exponenciální zdroj</i> DC - stejnosměrná složka, uplatní se v DC a AC (pracovní bod) AC - amplituda fázoru, jen pro analýzu AC časové parametry: V1, V2, TD1, TD2, TC1, TC2</p> 
<p>ISFFM VSFFM</p>	<p><i>Frekvenčně modulovaný zdroj</i> DC - stejnosměrná složka, uplatní se v DC a AC (pracovní bod) AC - amplituda fázoru, jen pro analýzu AC časové parametry: VOFF - stejnosměrný posuv, VAMPL - amplituda nosné, FC - frekvence nosné FM - modulační frekvence, MOD - index modulace $v(t) = voff + v ampl \cdot \sin(2\pi \cdot fc \cdot \text{TIME} + mod \cdot \sin(2\pi \cdot fm \cdot \text{TIME}))$</p>
<p>IPWL, VPWL, IPWL_ENH, VPWL_ENH, IPWL_FILE, VPWL_FILE</p>	<p><i>Po částech lineární zdroje</i> DC - stejnosměrná složka, uplatní se v DC a AC (pracovní bod) AC - amplituda fázoru, jen pro analýzu AC časové parametry (pro všechny zdroje): TSF - měřítko časové osy, VSF - měřítko hodnot Ti, Vi - uzlové body po částech lineárního průběhu FIRST_nPAIRS, ... THIRD_nPAIRS - skupiny párů (ti, vi)</p>

	REPEAT_VALUE - počet opakování skupin FILE - jméno externího souboru s páry t_i , v_i
ISTIM, VSTIM	Napětový a proudový zdroj pro použití s programem Stimulus Editor

Knihovna *breakout*

Tato knihovna obsahuje součástky s prázdným modelem určené k uživatelské modifikaci. Všechny prvky mají ve svém názvu slovo „break“ (např. Cbreak, Rbreak, atd.). Modifikace jejich modelu se provádí pomocí volby Edit PSpice Model.

Různé prvky

R, L, C	základní pasivní součástky, parametr TOLERENCE se musí udávat v procentech
K_linear	pseudosoučástka pro nastavení vzájemné indukčnosti až šesti induktorů, COUPLING je činitel vazby
T	bezztrátové vedení, popis parametrů je v příloze C
TLOSSY	ztrátové vedení, viz příloha C
XFRM_LINEAR	lineární model transformátoru,
XFRM_NONLINEAR	nelineární transformátor, musí se nastavit model magnetického jádra
Sw_tOpen, Sw_tClose	časový spínač a rozpínač

Speciální součástky

IProbe	ampérmetr, zobrazí ve schématu proud pracovního bodu
NODEST1	nastavení první iterace pro výpočet pracovního bodu, používá se když obvod nekonverguje
PARAM	deklarace parametrů pro krokování
VIEWPOINT	zobrazení napětí pracovního bodu ve schématu
INCLUDE	vložení souboru
LIB	vložení knihovny

Seznam použité literatury

- [1] PSpice Reference Guide, Cadence Design, 2000