

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA TECHNOLOGIÍ A MĚŘENÍ

DIPLOMOVÁ PRÁCE

**Návrh software pro řešení elektrických obvodů -
parser**

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Pavel KUPILÍK**
Osobní číslo: **E09N0030P**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Komerční elektrotechnika**
Název tématu: **Návrh software pro řešení elektrických obvodů - parser**
Zadávací katedra: **Katedra technologií a měření**

Z á s a d y p r o v y p r a c o v á n í :

1. Prostudujte teorii formálních jazyků a seznamte se základy konstrukce překladačů.
2. Prostudujte podrobně syntaxi jazyka PSpice.
3. V jazyce C++ vytvořte modul "Parser", který bude zpracovávat vstupní textový soubor ve formátu PSpice, výstupem budou vhodně navržené třídy připravené pro solver.

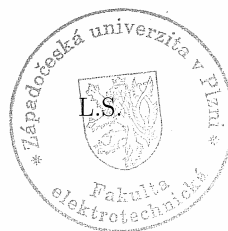
Rozsah grafických prací: podle doporučení vedoucího
Rozsah pracovní zprávy: 30 - 40 stran
Forma zpracování diplomové práce: tištěná/elektronická
Seznam odborné literatury:

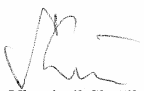
Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí diplomové práce: Ing. David Pánek, Ph.D.
Katedra teoretické elektrotechniky

Datum zadání diplomové práce: 18. října 2010
Termín odevzdání diplomové práce: 11. května 2012


Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan




Doc. Ing. Vlastimil Skočil, CSc.
vedoucí katedry

V Plzni dne 17. října 2011

Abstrakt

Předkládaná diplomová práce je zaměřena na základní část software pro řešení elektrických obvodů – parseru.

Klíčová slova

Parser, překladač, scanner, třída

Abstract

The master theses presents the principles of the basic part software for solving electrical circuit – parser.

Key words

Parser, compiler, scanner, class

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

V Plzni dne 10.5.2012

Bc. Pavel Kupilík

.....

Obsah

OBSAH	7
ÚVOD	8
SEZNAM SYMBOLŮ A ZKRATEK	9
1 KONSTRUKCE PŘEKLADAČE	10
1.1 LEXIKÁLNÍ ANALYZÁTOR - SCANNER	11
2 SYNTAXE JAZYKA PSPICE	12
2.1 VSTUPNÍ SOUBOR .CIR	12
2.2 PŘÍKAZY	12
2.3 ČÍSELNÉ KONSTANTY	12
2.4 NUMERICKÉ VÝRAZY	13
2.5 DEFINICE SOUČÁSTEK	15
2.5.1 Kapacitor <i>C</i>	15
2.5.2 Rezistor <i>R</i>	16
2.5.3 Indukčnost <i>L</i>	16
2.5.4 Transformátor	17
2.5.5 Dioda <i>D</i>	17
2.5.6 Nezávislý proudový zdroj <i>I</i>	17
2.5.7 Nezávislý zdroj napětí <i>V</i>	17
2.5.8 Napětový zdroj řízený napětovým zdrojem <i>E</i>	17
2.5.9 Napětový zdroj řízený proudovým zdrojem <i>G</i>	18
2.5.10 Bipolární tranzistor.....	18
2.5.11 Unipolární tranzistory.....	18
2.5.12 Přenosová linka <i>T</i>	19
2.5.13 Uživatelsky definovaná součástka <i>X</i>	19
2.5.14 Napětově řízený spínač <i>S</i>	19
2.6 ANALÝZA	19
2.6.1 Střídavá analýza <i>DC</i>	19
2.6.2 Stejnoseměrná analýza <i>DC</i>	20
2.6.3 Přechodová analýza <i>TRAN</i>	20
3 KONSTRUKCE PARSERU	21
ZÁVĚR	22
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	1

Úvod

Předkládaná práce je zaměřena návrh software pro řešení elektrických obvodů konkrétně na část, která se zabývá dekódováním zdrojového kódu jazyka PSpice a převodem do formátu použitelného v modulu řešiče.

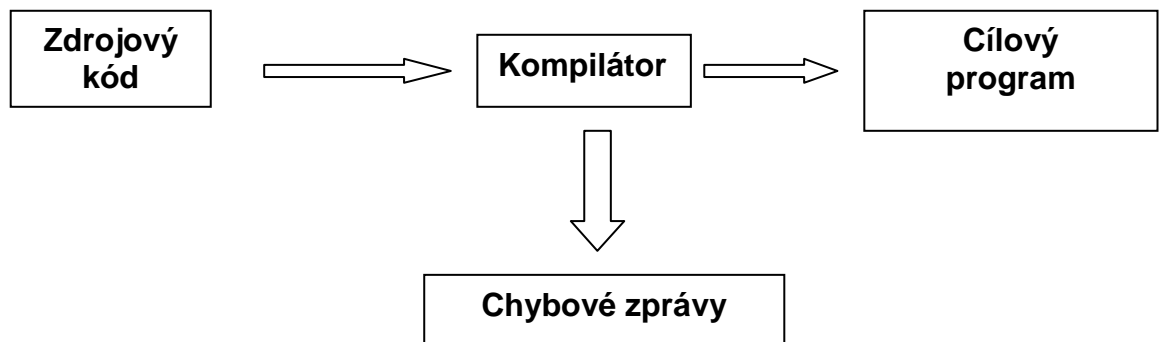
Text práce je rozdělen do tří částí. První se zabývá teorií formálního jazyka, konstrukcí a popisem překladače. Ve druhé části je popsána syntaxe jazyka PSpice. Třetí část popisuje samotný návrh modulu parseru pro analýzu vstupních dat v jazyce PSpice, které budou dále využívána pro modul solver neboli řešič. Pro naprogramování modulu parseru byl zvolen programovací jazyk C++ s využitím knihoven QT.

Seznam symbolů a zkratk

Qt	Knihovna jazyka C++
C++	Programovací jazyk
PSpice.....	Personal computer Simulation Program with Integrated Circuit Emphasis

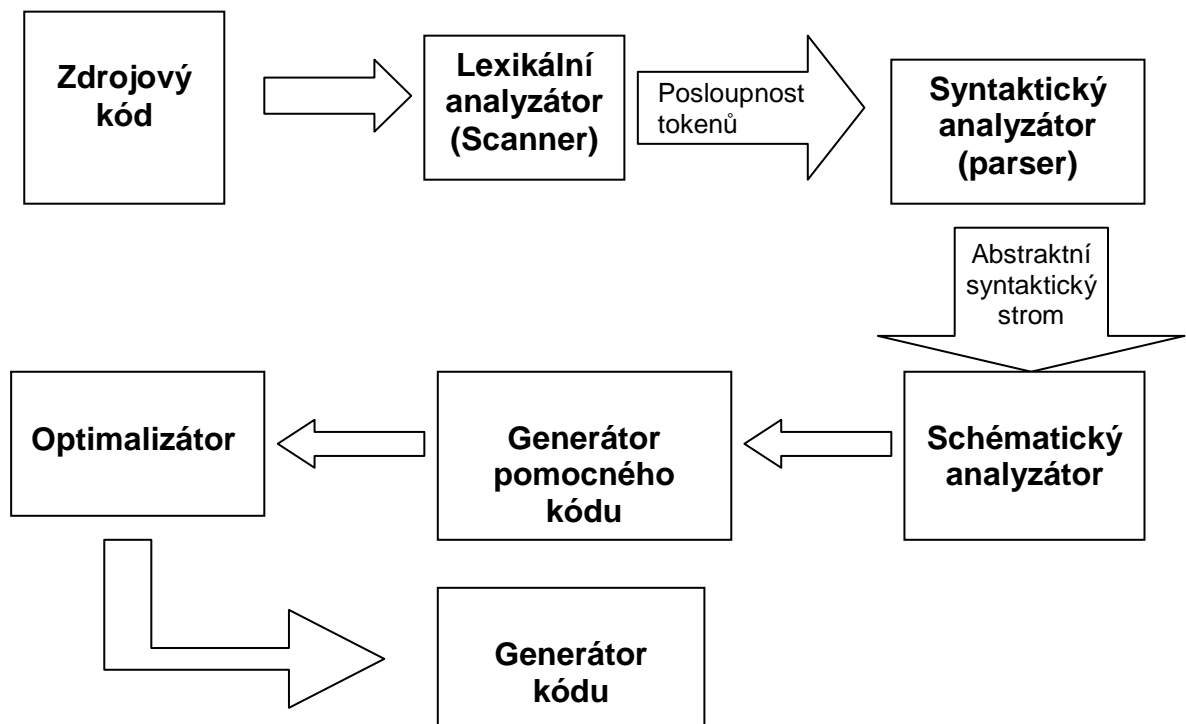
1 Konstrukce překladače

Překladač neboli kompilátor slouží k překladu zdrojového kódu vytvořeného programátorem do formy kterou využívá procesor. Překladač převádí kód z vyššího programovacího jazyka do strojového kódu procesoru. Na následujícím schématu je zjednodušeně znázorněno schéma překladače programu.



Obr.: 1.1. Zjednodušené schéma překladače programu

Základní principi kompilátoru spočívá v analýze zdrojového kódu, rozpoznání významných částí a převodu na cílový program, který je srozumitelný pro daný typ procesoru.



Obr.:1.2. Schéma kompilátoru

Překlad zdrojového kódu do cílového kódu probíhá v několika fázích, každá fáze převádí zdrojový soubor z jedné reprezentace na jinou. Proces kompilace se skládá z následujících fází:

Lineární analýza – v této fázi je zdrojový kód načten a jsou v něm vyhledávány posloupnosti znaků tvořících nedělitelný jednoznakový nebo víceznakový symbol neboli lexém, k tomuto lexému je přiřazen token. Tato fáze je zpracovávána ve scanneru.

Hierarchická analýza – v této fázi jsou tokeny sdruženy do celků které jsou určeny gramatickými pravidly.

Schématická analýza – v této fázi se provádějí kontroly, které zjišťují správný význam v kontextu programu. Scanner je dále odpovědný za rozpoznání identifikátorů

1.1 Lexikální analyzátor - scanner

Před začátkem vývoje parseru je třeba se zaměřit na scanner. Scanner rozpoznává symboly ve zdrojovém textu. Při konstrukci scanneru je nejprve nutné stanovit slovník

- / % + -
- = < > [] () .
- IF ELSE WHILE FOR

Tento slovník reprezentuje terminální symboly, které jsou nazývány rezervovanými znaky. Tyto znaky musí být rozeznány ve scanneru a není možné je použít například jako názvy proměnných.

Úkolem scanneru je následující:

- Přeskakovat konce řádků a prázdné znaky
- Rozpoznávat rezervované výrazy jako je např. if, else
- Sekvence písmen a číslic které začínají písmenem a zároveň nejsou rezervovaným výrazem jsou rozpoznány jako identifikátor
- Sekvence číslic je rozpoznána jako číslo
- Kombinace speciálních znaků jako = < > jsou rozpoznány jako symboly
- Komentáře programátora jsou přeskočeny
- Pokud scanner přečte neplatný znak vrací hodnotu null

2 Syntaxe jazyka PSpice

2.1 Vstupní soubor .cir

Vstupní soubor .cir slouží k popisu obvodu a začíná uvozovací řádkou, která slouží pro informaci, je možné ji i vynechat. Následuje netlist, což je popis součástek, zapojení v obvodu a popis vstupních parametrů, popřípadě odkaz do knihovny součástek nebo na model součástek, který následuje za netlistm. Pro popis k programu je vhodné používat komentáře. Uvozovací znak komentáře pokud začíná na novém řádku je *(hvězdička). V případě, že komentář nezačíná na začátku řádky ale například za příkazem je použit středník. Za netlustém pak dále pokračují příkazy simulátoru které jsou uvozeny tečkou. Konec definice obvodu je ukončen příkazem .END. V jednom souboru *.cir je možno použít více definic obvodů ale je nutné každou definici ukončit příkazem .END.

2.2 Příkazy

Příkazy v PSpice jsou další částí souboru *.cir. Lze je rozdělit do několika kategorií.

- standardní analýza - příkazy AC, DC, TRAN
- výstupní – PLOT, PRINT, PROBE
- práce se soubory – END, FUNC, LIB
- statistická analýza – MC, WCASE
- modelování součástek – MODEL
- inicializace stavů – IC, NODESET

2.3 Číselné konstanty

Číselné konstanty mohou být vyjádřeny jako celá čísla, čísla s plovoucí desetinou tečkou nebo jako čísla v exponenciálním tvaru. Dále je možno využít standardních symbolů:

F, f = 10^{-15}	P, p = 10^{-12}	N, n = 10^{-9}	U, u = 10^{-6}	M, m = 10^{-3}
K, k = 10^3	MEG, meg = 10^6	G, g = 10^9	T, t = 10^{12}	

Tab.: 2.1. Tabulka standardních symbolů pro číselné konstanty

U číselných konstant se nerozlišují velká a malá písmena (U i u znamenají mikro). Další znaky za symbolem jsou ignorovány až do oddělovače. Bezprostředně za konstantou je vhodné pro lepší orientaci uvádět fyzikální jednotky (1e3, 1.0e3, 1KHz reprezentují stejné číselné hodnoty).

2.4 Numerické výrazy

Pro numerické výrazy lze použít následující funkce:

Funkce	Význam	Popis
ABS(x)	x	Absolutní hodnota
ACOS(x)	arcuscosinus x	$-1.0 \leq x \leq +1.0$
ARCTAN(x)	$\tan^{-1}(x)$	výsledek v radiánech
ASIN (x)	arcussinus x	$-1.0 \leq x \leq +1.0$
ATAN(x)	$\tan^{-1}(x)$	výsledek v radiánech
ATAN2(y,x)	$\arctan(y/x)$	Výsledek v radiánech
COS(x)	$\cos(x)$	x v radiánech
COSH(x)	hyperbolický cosinus x	x v radiánech
EXP(x)	e^x	
IF(t, x, y)	x když t=TRUE y když t=FALSE	t je logický výraz hodnotou true nebo false, X a Y mají číselnou hodnotu nebo mohou být logické výrazy
IMG(x)	Imaginární část x	Hodnota 0.0 pro reálné číslo
LIMIT(x,min,max)		x=min - pokud $x < \min$ x=max – pokud $x > \max$ x=x v ostatních případech
LOG(x)	$\ln(x)$	Logaritmus se základem e
LOG10(x)	$\log(x)$	Logaritmus se základem 10
MAX(x,y)		Vrací větší z x nebo y
MIN(x,y)		Vrací menší z x nebo y
P(x)	Fáze x	Vrací 0.0 pro reálné číslo

PWR(x,y)	Mocnina	$ x ^y$
R(x)		Reálná část x
SIN(x)		x v radiánech
SINH(x)	Hyperbolický sin	x v radiánech
STP(x)	přepínač	x = 1 pro x>0.0 x = 0 pro x<0.0 např.:{v(1)*STP(TIME-10ns)} pro čas 0.0 až 10ns je hodnota 0 potom je hodnota v(1)
SQRT(x)	\sqrt{x}	
TAN(x)	tan(x)	x v radiánech
TANH(x)	Hyperbolický tan	x v radiánech
TABLE(x,x1,y1,x2,y2, xn,yn)		definuje tabulku
funkce	význam	popis
ABS(x)	x	Absolutní hodnota
ACOS(x)	arcuscosinus x	-1.0 ≤ x ≤ +1.0
ARCTAN(x)	tan-1(x)	výsledek v radiánech
ASIN (x)	arcussinus x	-1.0 ≤ x ≤ +1.0
ATAN(x)	tan-1(x)	výsledek v radiánech
ATAN2(y,x)	arctan(y/x)	Výsledek v radiánech
COS(x)	cos(x)	x v radiánech
COSH(x)	hyperbolický cosinus x	x v radiánech
EXP(x)	e^x	
IF(t, x, y)	x když t=TRUE y když t=FALSE	t je logický výraz hodnotou true nebo false, X a Y mají číselnou hodnotu nebo mohou být logické výrazy
IMG(x)	Imaginární část x	Hodnota 0.0 pro reálné číslo
LIMIT(x,min,max)		x=min - pokud x<min x=max – pokud x>max x=x v ostatních případech

LOG(x)	Ln(x)	Logaritmus se základem e
LOG10(x)	log(x)	Logaritmus se základem 10
MAX(x,y)		Vrací větší z x nebo y
MIN(x,y)		Vrací menší z x nebo y
P(x)	Fáze x	Vrací 0.0 pro reálné číslo
PWR(x,y)	Mocnina	$ x ^y$
R(x)		Reálná část x
SIN(x)		x v radiánech
SINH(x)	Hyperbolický sin	x v radiánech
STP(x)	přepínač	x = 1 pro x > 0.0 x = 0 pro x < 0.0 např.: {v(1)*STP(TIME-10ns)} pro čas 0.0 až 10ns je hodnota 0 potom je hodnota v(1)
SQRT(x)	\sqrt{x}	
TAN(x)	tan(x)	x v radiánech
TANH(x)	Hyperbolický tan	x v radiánech
TABLE(x,x1,y1,x2,y2, xn,yn)		definuje tabulku

Tab.: 2.2 Tabulka funkcí pro numerické výrazy

2.5 Definice součástek

2.5.1 Kapacitor C

C<název> <(+) uzel> <(-) uzel> [název modelu] <hodnota> [IC=<počáteční hodnota>]

Definice kapacitoru musí začínat písmenem C a dále bez mezery je uveden název součástky. Další argumenty určují zapojení součástky v obvodu. První uzel je definován jako kladný. Napětí na součástce je pak rozdíl napětí v jednotlivých uzlech. V případě, že je název modelu vynechán, hodnota kapacity je uvedena ve faradech. Pokud model definován je hodnota kapacity určují parametry uvedené v modelu. U součástek je rovněž možné nastavit počáteční hodnotu pomocí parametru IC.

Model kapacitoru CAP

Model kapacitou je definován následovně.

.MODEL <název modelu> CAP [parametry modelu]

Model kapacitou je definován jako lineární, teplotně závislý, ale tyto parametry v modelu jsou nastaveny na nulovou hodnotu. V případě, že je definován název modelu výpočet hodnoty kapacity je určen následujícím vzorcem:

$$\langle value \rangle \cdot C \cdot (1 + VC1 \cdot V + VC2 \cdot V^2) \cdot (1 + TC1 \cdot (T - Tnom) + TC2 \cdot (T - Tnom)^2)$$

C - kapacitní násobitel

TC1 - lineární teplotní koeficient

TC2 – kvadratický teplotní koeficient

VC1 – lineární napěťový koeficient

VC2 – kvadratický napěťový koeficient

2.5.2 Rezistor R

R<název> <(+)uzel> <(-) uzel> [název modelu] <hodnota> [TC = <TC1> [,<TC2>]]

Rezistor je definován obdobně jako kapacitor, definici uvozuje označení součástky R, následuje zapojení v obvodu, název modelu pokud je nadefinován a hodnota součástky.

Model odporu RES

.MODEL <název modelu> RES [parametry modelu]

Model odporu je stejně jako model kapacitou nadefinován jako teplotně závislý, lze nastavit parametry TC1 a TC2. Tyto parametry jsou opět nadefinovány na nulovou hodnotu. Hodnota odporu je stanovena následovně

$$\langle hodnota \rangle \cdot R \cdot (1 + TC1 \cdot (T - Tnom) + TC2 \cdot (T - Tnom)^2)$$

2.5.3 Indukčnost L

L<název> <(+) uzel> <(-) uzel> [název modelu] <hodnota> [IC=<počáteční hodnota>]

Model indukčnosti IND

.MODEL <název modelu> IND [parametry modelu]

$$\langle value \rangle \cdot L \cdot (1 + IL1 \cdot I + IL2 \cdot I^2) \cdot (1 + TC1 \cdot (T - Tnom) + TC2 \cdot (T - Tnom)^2)$$

2.5.4 Transformátor

K <název> L <jméno induktoru> < L <jméno induktoru>>* <vzájemná indukčnost>
 K <název> < L <jméno induktoru>>* <vzájemná indukčnost hodnota>
 + <název modelu> [parametry]

Model transformátoru CORE

.MODEL <název modelu> CORE [parametry modelu]

2.5.5 Dioda D

D <název> <(+) uzel> <(-) uzel> <název modelu> [další parametry]

Model diody D

.MODEL <název modelu> D [parametry modelu]

2.5.6 Nezávislý proudový zdroj I

I <název> <+ uzel> <- uzel> [[DC] <hodnota>] [AC <hodnota> [fázový posun]]
 +[přenosové funkce]

Zdroj proudu I lze definovat jako stejnosměrný, potom není nutné označení DC uvádět, nebo jako střídavý s nastavitelnou velikostí amplitudy. Dále je možné nastavovat přenosovou funkci a volit tak tvar proudu zdroje.

[přenosové funkce]

EXP (<parametry>) exponenciální funkce

PULSE (<parametry>) pulsní funkce

PWL (<parametry >) pulsně-šířková funkce

SFFM (<parametry >) frekvenčně modulovaná funkce

SIN (<parametry >) sinusová funkce

2.5.7 Nezávislý zdroj napětí V

V <název> <+ uzel> <- uzel> [[DC] <hodnota>] [AC <hodnota> [fázový posuv]]
 [přenosová funkce]

Obdobně jako u proudového zdroje lze volit stejnosměrný nebo střídavý zdroj. Lze nastavit také parametr přenosové funkce.

2.5.8 Napěťový zdroj řízený napěťovým zdrojem E

E <název> <+ svorka> <- svorka> <+ řídicí svorka> <- řídicí svorka> <svorka

substrát>

2.5.9 Napěťový zdroj řízený proudovým zdrojem G

G<název> <+ svorka> <- svorka> <+ řídicí svorka> <- řídicí svorka>

2.5.10 Bipolární tranzistor

Q<název> <kolektor> <báze> <emitor> [uzel substrát] <název modelu> [další parametry]

Bipolární tranzistor je definován jako součástka typu Q. Součástka je definována jako čtyřvývodová báze, emitor, kolektor a může být dále definován vývod substrát. Za názvem modelu lze definovat další parametry. Bipolární tranzistor lze definovat jako PNP nebo NPN.

Model bipolárního tranzistoru NPN, PNP

.MODEL <název modelu> NPN [parametry modelu]

.MODEL <název modelu> PNP [parametry modelu]

2.5.11 Unipolární tranzistory

2.5.11.1 JFET

J<název> <drain uzel> <gate uzel> <source uzel> <název modelu> +[další parametry]

Součástka tranzistoru JFET je definována jako čtyřvývodová. Za definicí vývodů je uveden

Model Unipolárního tranzistoru NJF, PJF

.MODEL <název modelu> NJF [parametry modelu]

.MODEL <název modelu> PJF [parametry modelu]

2.5.11.2 MOSFET

M<název> <drain uzel> <gate uzel> <source uzel> <substrát svorka> <název modelu> [L=<hodnota>] [W=<hodnota>]+ [AD=<hodnota>] [AS=<hodnota>] +[PD=<hodnota>] [PS=<hodnota>] [NRD=<hodnota>] [NRS=<hodnota>] +[NRG=<hodnota>] [NRB=<hodnota>] [M=<hodnota>] [N=<hodnota>]

Součástka typu MOSFET je definována jako čtyřvývodová. Má definovány vývody

gate, drain, source a substrát. Dále je třeba definovat model součástky. Model součástky může být definován jako typ NMOS nebo PMOS.

Model MOSFET NMOS, PMOS

.MODEL <název modelu> NMOS [parametry modelu]

.MODEL <název modelu> PMOS [parametry modelu]

2.5.12 Přenosová linka T

*T<název> <A port (+) uzel> <A port (-) uzel> <B port (+) uzel> <B port (-) uzel>
+ [<název modelu>] LEN=<hodnota> R=<hodnota> L=<hodnota> G=<hodnota>
C=<hodnota>*

Model vedení je definován jako čtyřpól, který je zapojen mezi svorky A port a B port. V modelu jsou definovány součástky odpor, indukčnost, svod a kapacita.

model

Model vedení TRN

.MODEL <název modelu> TRN [parametry modelu]

2.5.13 Uživatelsky definovaná součástka X

X<název> [seznam vývodů] <název obvodu >[parametry:<<název>=<hodnota>>]*]*

V PSpice je možné také nadefinovat vlastní typ součástky typu X.

2.5.14 Napětově řízený spínač S

S<název> <+ přepínaná svorka> <- přepínaná svorka> <+ řízená svorka> <- řízená svorka>

2.6 Analýza

2.6.1 Střídavá analýza DC

.AC <typ > <počet bodů> <počáteční frekvence> <konečná frekvence>

Střídavá analýza určuje chování obvodu v určitém frekvenčním intervalu, okolo pracovního bodu je obvod linearizován. Tuto linearizaci určuje typ

<počet bodů> je celé číslo a určuje počet bodů v intervalu dle typu linearizace.

<typ> jsou definovány tři typy linearizace - lineární(LIN), oktávová(OCT) a linearizace

po dekádách(DEC).

Lineární LIN - body pro zjištění chování obvodu jsou rozmístěny v celém rozsahu lineárně, hodnota počtu bodů určuje celkový počet bodů v analyzovaném frekvenčním rozsahu.

Oktávové OCT - body pro frekvenční analýzu jsou rozmístěny logaritmicky a počet bodů určuje počet bodů v jednotlivých oktávách

Po dekádách DEC - body jsou rozmístěny logaritmicky, počet bodů je hodnota jednotlivých dekádách.

Interval frekvenční analýzy

Počáteční frekvence musí být nižší než konečná a obě frekvence musí být vyšší než nula. Frekvenční rozsah musí být také dostatečně velký, aby mohl obsahovat alespoň jeden bod.

2.6.2 Stejnoseměrná analýza DC

DC [LIN] <jméno proměnné> <počáteční hodnota> <koneční hodnota> <krok>

*DC <typ logaritmického kroku> <jméno proměnné> <počáteční hodnota>
+<konečná hodnota> <počet bodů>*

Definice stejnosměrné analýzy s logaritmickým krokem musí obsahovat typ kroku (po oktávách OCT nebo po dekádách DEC) a následuje stejně jako lineární analýzy jméno proměnné. Dále následuje počáteční a konečný hodnota rozsahu změny zadané proměnné.

2.6.3 Přejchodová analýza TRAN

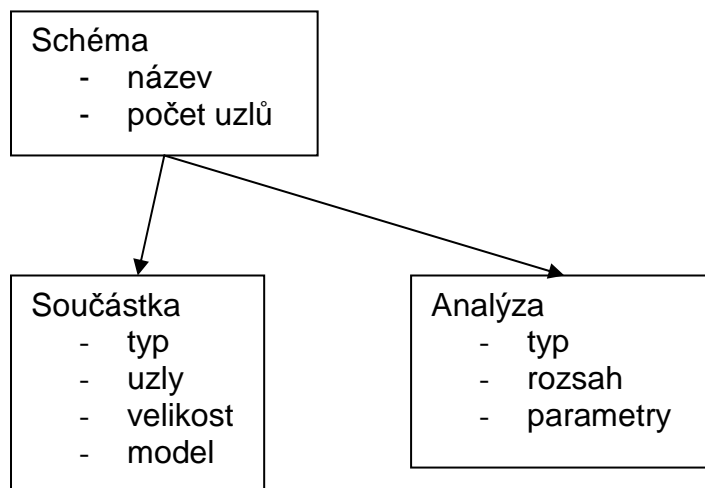
.TRAN[/OP] <časový krok> <konečný čas> +[počátek výpisu [mez časového kroku]]

Při přechodové analýze se je využita proměnná TIME která se mění od nuly do hodnoty konečného času. Hodnota časového kroku je využita při výpisu do výstupního souboru. Hodnota počátek výpisu udává od kdy se začnou výstupní hodnoty zapisovat do výstupního souboru.

3 Konstrukce parseru

Parser je navržen pro zpracování vstupního souboru *.cir, ze kterého získává data dle pravidel uvedených v předchozích kapitolách. Výstupem bude seznam součástek jejich parametrů v objektu typu LIST<>.

Vstupem do parseru bude název souboru, který bude zadán uživatelem nebo bude získán při spuštění programu jako vstupní parametr funkce main() *argv. Tento název souboru bude použit například pro výstupní soubor typu *.dat, do kterého jsou ukládány hodnoty po analýze obvodu. Následující schéma ukazuje strukturu výstupních dat parseru.



obr.: 3.1 Schéma výstupních dat

Virtuální třída součástka bude obsahovat virtuální metody pro zjištění typu součástky. Dále datové složky které budou pro všechny součástky stejné tj. typ součástky, uzly a model. Od této třídy budou odděněny další třídy pro jednotlivé součástky. V těchto třídách budou implementovány metody které budou ze vstupních dat načítat řetězce a pomocí regulárních výrazů parsovat vstupní data.

Závěr

Tato práce je rozdělena do tří kapitol. V první části je popsána konstrukce překladače rozdělení na jednotlivé části a jejich popis.

Druhá část se zabývá popisem jazyka PSpice. Jsou zde rozebrána syntaxe jazyka a jednotlivých komponent jako jsou jednotlivé součástky a analýzy obvodů. Dále se tato kapitola věnuje popisu jednotlivých funkcí které lze v jazyce PSpice využít.

Třetí kapitola pak popisuje vlastní konstrukci samotného parseru jako je struktura výstupních dat.

Seznam literatury a informačních zdrojů

- [1] WIRTH, Niklaus. Compiler Construction. Zürich, 2005, ISBN 0-201-40353-6
- [2] Referenc manual. [online]. Dostupné <http://www.electronic-lab.com/downloads/schematic/013/tutorial/PSPCREF.pdf>
- [3] Úvod do kompilátorů.[online].Dostupné <http://kifri.fri.uniza.sk/~bene/vyuka/kompilatory/prednasky-prezentace/Predn-01-text.pdf>