ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ

KATEDRA APLIKOVANÉ ELEKTRONIKY A TELEKOMUNIKACÍ

Studijní program Elektrotechnika a informatika

Studijní obor Dopravní elektroinženýrství a autoelektronika

DIPLOMOVÁ PRÁCE

VYUŽITÍ POKROČILÝCH POČÍTAČOVÝCH SIMULACÍ PRO ANALÝZU OBVODŮ V ŽELEZNIČNÍ ZABEZPEČOVACÍ TECHNICE

Petr Mazour

Vedoucí práce:
Doc. Ing. Ivan Konečný, CSc.
Katedra aplikované elektroniky a telekomunikací
Fakulta elektrotechnická Západočeské univerzity v Plzni

Konzultant:
Ing. Petr Hloušek, Ph. D
Katedra aplikované elektroniky a telekomunikací
Fakulta elektrotechnická Západočeské univerzity v Plzni

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: **2011/2012**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Petr MAZOUR**
Osobní číslo:	**E10N0161P**
Studijní program:	**N2612 Elektrotechnika a informatika**
Studijní obor:	**Dopravní elektroinženýrství a autoelektronika**
Název tématu:	**Využití pokročilých počítačových simulací pro analýzu obvodů v železniční zabezpečovací technice**
Zadávající katedra:	**Katedra aplikované elektroniky a telekomunikací**

Zásady pro vypracování:

1. Popis pokročilých analýz obvodových simulací v programovém balíku ICAP/4.
2. Prověření jednotlivých simulací na zadaných vzorových obvodech.
3. Zhodnocení dosažených výsledků a případný návrh dalších možností využití v oblasti železniční zabezpečovací techniky.

Rozsah grafických prací:	**dle doporučení vedoucího**
Rozsah pracovní zprávy:	**dle doporučení vedoucího**
Forma zpracování diplomové práce:	**tištěná/elektronická**
Seznam odborné literatury:	

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

Vedoucí diplomové práce:	**Doc. Ing. Ivan Konečný, CSc.** Katedra aplikované elektroniky a telekomunikací
Konzultant diplomové práce:	**Ing. Petr Hloušek, Ph.D.** Katedra aplikované elektroniky a telekomunikací
Datum zadání diplomové práce:	**17. října 2011**
Termín odevzdání diplomové práce:	**11. května 2012**

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

L.S.

Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 17. října 2011

Anotace

Diplomová práce se zabývá strukturou NETLISTU a kompatibilitou programů PSPICE a ICAP/4, a úpravou v textovém editoru programu ICAP/4. V práci je ukázán rozbor bezpečnosti poruchových stavů v železniční zabezpečovací technice.

Klíčová slova:

PSPICE, ICAP/4, kompatibilita programů SPICE, časově proměnné L a C, kapacitní dekodér, poruchové stavy součástek.

Abstract

This thesis is focused on structure of the NETLIST and a compatibility of the PSPICE and ICAP/4 programs, and possible transfer it into a text editor ICAP/4 program. In the topic is shown safety analysis of faults in railway signaling.

Key words:

PSPICE, ICAP/4, compatibility of SPICE programs, time-varying L and C, capacity decoder, safety analysis of parts.

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 Petr Mazour

..…..…….

Poděkování

Chtěl bych poděkovat vedoucímu diplomové práce Doc. Ing. Ivanu Konečnému, CSc. a konzultantovi Ing. Petru Hlouškovi, Ph.D. za ochotu, pomoc a poskytnutí profesionálních rad a připomínky při konzultacích diplomové práce.

Obsah

1 Úvod

Cílem a úkolem této práce je přiblížit problematiku kompatibility dvou programů SPICE. Jedná se o program PSPICE, který je součástí balíku OrCAD vytvořený firmou Cadence a program ICAP/4 (dále jen ICAP) od firmy Intusoft. Budou zde popsány rozdíly struktury NETLISTU a provedené změny v syntaxi pro zachování kompatibility obou programů.

Diplomová práce je rozdělena na dvě hlavní části. Jedna část se zaměřuje na kompatibilitu programů PSPICE a ICAP, a druhá část na rozbor bezpečnosti poruchových stavů zadaného obvodu.

Obsah první části diplomové práce se skládá z popisu struktury NETLISTU, kde je znázorněno, jaké části musí NETLIST obsahovat. V další části bude popsána změna syntaxe obou programů pro splnění jejich kompatibility. Jedná se o rozdíly v syntaxi NETLISTU a jejich popis s možnou úpravou z PSPICE do programu ICAP. Dále obsahuje popis převodu NETLISTU z PSPICE do ICAP v textovém editoru ISED5 integrovaným převodníkem od firmy Intusoft. Poslední část se zabývá možností modelování akumulačních prvků s proměnnými parametry L a C. Bude zde ukázán příklad modelování v programu PSPICE a úprava pro převod NETLISTU do programu ICAP.

V druhé části diplomové práce proběhne vyzkoušení poruchových stavů na zadaném obvodu kapacitního dekodéru v programu PSPICE a ICAP. Na tomto obvodu budou vyzkoušeny poruchové stavy *přerušení* a *zkrat* daných součástek a výsledky simulací pak porovnávány. Potřebné modely součástek pro analýzu poruchových stavů vytvořili v programu PSPICE Ing. Ivan Tuháček a Ing. Roman Tříska. Převod dostupných modelů součástek do programu ICAP/4 vytvořil Ing. Lukáš Roztočil.

Tato práce navazuje na diplomovou práci Lukáše Roztočila, který ve své práci popisuje kompatibilitu programů PSPICE a ICAP, přenesení modelů a knihoven z PSPICE do programu ICAP. Dále pak základní strukturu NETLISTU ICAP a převod parametrů poruchových stavů součástek pro analýzu obvodů z programu PSPICE do ICAP.

V práci byla využívána volně dostupná verze programu PSPICE 9.1 Student Version a plná verze programu ICAP/4 Professional: 8.2.11 Build 3839.

Jednotlivé typy písma použité v dalším textu mají následující význam:

Times New Roman	základní typ písma
Times New Roman, kurzíva	*definice a příklady struktury NETLISTU, výpis programů, chybové hlášky, záložky*
Times New Roman, tučné	***zvýrazněná důležitá slova a spojení***
Times New Roman, tučné	***úprava parametrů a příkazů***

V dalším textu použité značky a symboly:

(…) Označení klávesové zkratky v textu. Místo … je dosazena

příslušná zkratka

[…] Označení tlačítka. Místo … je popis tlačítka.

„…“ Označení symbolů a chybových hlášek. Místo … je dosazen symbol nebo chybová hláška.

„…./…/…“ Příkaz z nabídky menu. V uvozovkách je uvedena cesta.

Například: „Menu/File/New“ pro vytvoření nového schématu.

2 Kompatibilita programů PSPICE a ICAP

Program SPICE byl vyvinut v 70. letech 20. století. Postupem času se objevovaly další programy, které umožňovaly simulace elektronických obvodů. Tyto programy byly vyvinuty různými firmami, kde se syntaxe začala odlišovat od původního programu SPICE. Programy tak přestaly být vzájemně plně kompatibilní. Je tedy nutné částečně upravit jejich syntaxi, aby byla zachována kompatibilita programů.

Do rodiny programů SPICE patří ICAP a PSPICE. Syntaxe u většiny analogových obvodů v programu PSPICE a ICAP je stejná. Je zde ale několik malých rozdílů, které dělají PSPICE modely nefunkční v programu ICAP. Proto je nutné NETLIST, vytvořený v programu PSPICE přeložit či převést do podoby, vhodné pro správné fungování v programu ICAP.

V této kapitole budou popsány odlišnosti ve struktuře NETLISTU mezi oběma programy. PSPICE a ICAP nejsou plně kompatibilní a je tedy nutné NESTLIST částečně upravit. Tyto úpravy spočívají pouze v drobných změnách v syntaxi programu.

V první fázi bude popsána struktura zadávání syntaxe a případně i příklad v programu PSPICE. Následně poté provedena změna syntaxe do programu ICAP a nakonec popis definice struktury syntaxe s případným příkladem funkční v ICAP. V druhé části práce bude popsán převod NETLISTU, vytvořený v programu PSPICE, převodníkem textového editoru ISED5 do programu ICAP. Poslední část se zabývá možností modelování akumulačních prvků L a C. Bude ukázán příklad v programu PSPICE a jeho úprava pro převod do ICAP.

Struktura NETLISTU již byla popsána v diplomové práci Ing. Lukáše Roztočila, bude zde připomenuta jen ve spojení se zmíněnými odlišnostmi u obou programů.

2.1 Struktura NETLISTU programu ICAP

NETLIST je vstupní textový soubor programu ICAP. Obsahuje popis každé součástky obvodu, propojení součástek a jejich vlastnosti, včetně analýzy, kterou má příslušný program vytvořit. Vstupní textový soubor má koncovku *CIR*.

Každý vstupní textový soubor obsahuje 5 hlavních částí:

- *Název*
- *Příkazy pro vykonání analýz*
- *Příkazy pro popis výstupů programu*
- *Řádný popis obvodu*
- ***.END***

Každý NETLIST musí obsahovat *Název* a příkaz ***.END***. Název se nachází na prvním řádku NETLISTU a slouží k popisu programu vytvořeného obvodu. Tento řádek program automaticky ignoruje. Příkaz ***.END*** se nachází na posledním řádku a ukončuje vstupní textový soubor.

Příkazy pro vykonání analýz jsou skupinou příkazů a definují, jaké typy analýz bude program vykonávat pro vytvořený obvod. Jednotlivé typy analýz byly popsány v dokumentu ICAP_analyzy.pdf Lukáše Roztočila.

Další částí jsou *Příkazy pro popis výstupů programu*. Pro zobrazení výsledků simulací do výstupního souboru slouží příkazy ***.PRINT***, ***.PLOT***, ***.VIEW***. a příkaz ***.PROBE/CSDF***. Všechny tyto příkazy zapisují data do výstupního textového souboru s koncovkou *OUT*.

Příkazy ***.PRINT***, ***.PLOT*** a ***.VIEW*** byly popsány v diplomové práci Lukáše Roztočila [1] případně si je můžete prostudovat v manuálu od výrobce [2].

Příkaz ***.PROBE/CSDF*** je kompaktní datový formát, který slouží k uložení všech dat do výstupního textového souboru. Využívá se pouze v případě, pokud jsou použity příkazy analýz ***.AC***, ***.DC*** nebo ***.TRAN***. Je zde možnost využití tohoto příkazu pro kompatibilitu programů PSPICE a ICAP (viz kapitola 2.4.2).

Poslední částí je *Řádný popis obvodu*, kde jsou popsány použité součástky a jejich vzájemné propojení v obvodu. Každá součástka je zde popsána zvlášť buď pomocí definice pro pasivní součástky, nebo jako aktivní součástka pomocí modelů.

Příklad definice pasivní součástky:

R1 1 0 1k

Tento příkaz pasivní součástky definuje rezistor. Rezistor má název R1, v obvodu je zapojen mezi uzly 1 a 0 (zemí) s hodnotou 1 kΩ.

Příklad definice aktivní součástky pomocí modelu bude popsán v kapitole 2.2.1.

Pomocí podobvodů mohou být také vytvořeny obvody, obsahující značné množství součástek. Podobvodem lze i vytvořit pouze jednu součástku, kterou nelze definovat pomocí příkazu ***.MODEL***.

Podobvod začíná příkazem ***.SUBCKT***. Za tímto příkazem následuje jméno podobvodu, uzly zapojení a parametry. Parametry lze měnit vlastnosti obvodu. Podobvod musí být ukončen příkazem ***.ENDS***. Uvnitř podobvodu je umístěn popis součástek. Jméno podobvodu odkazuje na daný podobvod v NETLISTU.

Definice podobvodu:

.SUBCKT jméno podobvodu uzly parametry

popis součástek

.ENDS

Vytvořený podobvod je volán jako součástka začínající písmenem ***X***:

Xjméno uzly jméno podobvodu parametry

V této kapitole se čerpalo z literatury [1][2].

2.2 Úprava NETLISTU v textovém editoru

Jedná se o drobné změny, které provádí sám uživatel. Musí se ale počítat s časovou náročností a možností zvýšení chyb, které rostou se složitějšími obvody při provádění změn v syntaxi programu. Proto je tento postup vhodný pouze pro zkušené uživatele, kteří již mají s programy typu SPICE bohaté zkušenosti.

Bude zde uvedeno, jaké změny musí být provedeny pro splnění kompatibility obou programů. Prováděné změny jsou zvýrazněny tučnou červenou barvou a zobrazeny kurzívou.

Většina odlišností obou programů byla popsána v literatuře [3]. Ostatní odlišnosti byly čerpány z literatury [2][4-6].

2.2.1 Typ modelu v příkazu *.MODEL*

Příkaz *.MODEL* slouží k definování aktivních součástek. Tímto příkazem může uživatel změnit implicitně nastavené hodnoty parametrů konkrétní součástky podle svých představ. Používá se též i pro přesnější popis vlastností pasivních součástek (například pro toleranci součástek, teplotní závislosti atd.).

Modely jsou uloženy v knihovnách s koncovkou *LIB*. V programu PSPICE jsou všechny odvozené modely volány pomocí standardního knihovního souboru *NOM.LIB*, který bude popsán v kapitole 2.3.1.1.

Struktura modelové řady součástky v programech SPICE:

.MODEL *jméno modelu* *typ modelu* *parametry*

V této kapitole bude popsána změna typu modelu pouze pro pasivní součástky, například na jednoduchém příkladu modelu rezistoru. V NETLISTU programu ICAP musí být změněn parametr typ modelu ***RES*** na ***R***. Převod je pak prováděn podle následujícího pravidla.

Struktura PSPICE modelu rezistoru:

.MODEL *jméno modelu* ***RES*** *parametry*

.MODEL *RMOD* ***RES*** *parametry*

Struktura ICAP modelu rezistoru:

.MODEL jméno modelu ***R*** *parametry*

.MODEL RMOD ***R*** *parametry*

Stejný způsob převodu platí například i pro kondenzátory. Pak převod typu modelu kondenzátoru je následující.

Struktura PSPICE modelu kondenzátoru:

.MODEL CMOD ***CAP*** *parametry*

Struktura ISPICE modelu kondenzátoru:

.MODEL CMOD ***C*** *parametry*

2.2.2 Parametr měření teploty v příkazu *.MODEL*

Parametr měření teploty udává teplotu, při které jsou získávány (určeny či naměřeny) parametry součástky. Teplota je zadávána v °C. Pokud není zadána teplota uživatelem, je nastavena na standardní hodnotu 27 °C globální proměnnou *TNOM*.

V programu PSPICE je definován parametr měření teploty parametrem ***T_MEASURED***. Pokud parametr není definován, tak platí podmínka T_MEASURED = TNOM.

Struktura PSPICE modelu s parametrem měření teploty:

.MODEL jméno modelu typ modelu ***T_MEASURED***

V ICAP je definován parametr měření teploty parametrem ***TNOM***. Proto při zadávání parametru měření teploty v programu ICAP musí být parametr ***T_MEASURED*** změněn na parametr ***TNOM***.

Pak struktura modelové řady se změnou parametru měření teploty v programu ICAP vypadá následovně:

.MODEL *jméno modelu* *typ modelu* ***TNOM***

2.2.3 Jméno modelu a hodnota součástky

Pro popis jednoduchého modelu pasivní součástky, například rezistoru, stačí jej definovat pouze jedním řádkovým zápisem. Pak celý řádek obsahuje pouze jméno součástky, uzel vstupu a výstupu, jméno modelu a nakonec hodnotu součástky.

PSPICE definice a příklad rezistoru:

Rjméno	*uzel vstupu*	*uzel výstupu*	***JMÉNO MODELU***	***HODNOTA***
R1	*1*	*2*	***RMOD***	***1k***

V programu ICAP je tento model součástky popsán trochu odlišněji. Odlišnost spočívá pouze v záměně ***JMÉNA MODELU*** a ***HODNOTY*** součástky. Musí být tedy tyto parametry prohozeny. Ostatní parametry zůstávají nezměněny.

Definice a příklad rezistoru v ICAP:

Rjméno	*uzel vstupu*	*uzel výstupu*	***HODNOTA***	***JMÉNO MODELU***
R1	*1*	*2*	***1k***	***RMOD***

Stejný způsob převodu platí například i pro kondenzátory. Pak příklad prohození ***JMÉNA MODELU*** a ***HODNOTY*** u kondenzátoru je následující.

Příklad PSPICE kondenzátoru:

C1	*1*	*2*	***CMOD***	***10u***

Příklad ICAP kondenzátoru:

C1	*1*	*2*	***10u***	***CMOD***

2.2.4 Parametr konstantní teplota

Parametr konstantní teplota může být zadán buď pro vytvořený celý obvod, nebo pro zvolenou součástku ve vytvořeném obvodu, při níž bude obvod či součástka simulována. Parametr konstantní teplota nastavuje zvolenou teplotu, nezávislou na globální teplotě *TNOM*.

Stejně jako parametr měření teploty v příkazu *.MODEL* (viz kapitola 2.2.2), je parametr konstantní teplota zadáván v °C a defaultně nastaven na teplotu 27°C.

2.2.4.1 Konstantní teplota součástky

Konstantní teplota součástky je definována pomocí modelové řady. V programu PSPICE je definována parametrem ***T_ABS***. V programu ICAP musí být přejmenován parametr ***T_ABS*** na parametr ***TEMP*** a přemístěn na řádek instance, který volá příkaz *.MODEL*.

Struktura PSPICE modelu součástky:

jméno instance *uzel vstupu* *uzel výstupu* *jméno modelu*

.MODEL *jméno modelu* *typ modelu* ***T_ABS***

Struktura ICAP modelu součástky:

jméno instance *uzel vstupu* *uzel výstupu* *jméno modelu* ***TEMP***

.MODEL *jméno modelu* *typ modelu*

2.2.4.2 Konstantní teplota obvodu

Konstantní teplota, pro kterou bude celý obvod simulován v programu PSPICE, je definována příkazem ***.TEMP***. Tento příkaz nastavuje teplotu všech teplotně závislých prvků obvodu. Pokud je zadáno více teplot v příkazu, simulace proběhne opakovaně pro každou zadanou teplotu.

Definice a příklad v PSPICE:

.TEMP *hodnoty*

.TEMP *-8 43 76*

Analýza proběhne opakovaně pro teploty -8°C, 43°C a 76°C.

Tato definice konstantní teploty pomocí příkazu *.TEMP* není podporována v programu ICAP. Zde se nastavuje teplota obvodu pomocí parametru ***TEMP*** v příkazu ***.OPTIONS***.

Definice konstantní teploty obvodu v programu ICAP:

.OPTIONS ***TEMP*** *hodnoty*

2.2.5 Typ modelu a parametry spínače

Převod popsaný v této kapitole bude prováděn pro spínač s hysterezní charakteristikou. Tento spínač je řízen buď napětím, nebo proudem. Nejprve zde bude proveden převod pro spínač řízený napětím a poté pro spínač řízený proudem.

Modelová řada spínače řízeného napětím obsahující typ modelu ***VSWITCH*** vypadá v programu PSPICE následovně.

PSPICE model spínače:

.MODEL SMOD ***VSWITCH*** *RON ROF* ***VON VOFF***

SMOD je jméno modelu spínače, ***VSWITCH*** je typ modelu, *RON* je odpor v sepnutém stavu a *ROFF* je odpor v rozepnutém stavu.

Pro správnou funkčnost spínače v programu ICAP musí být změněn typ modelu ***VSWITCH*** na typ ***SW***, parametry *RON* a *ROFF* zůstávají nezměněny. Pak pro napěťově řízený spínač dále musí být změněny parametry ***VON***, ***VOFF*** na ***VT***, ***VH*** a přepočteny podle následujícího pravidla.

VON = VT+VH,

VOFF = VT-VH,

pak ***VT = (VON+VOFF)/2***,

VH = VON-VT

Parametr ***VON*** udává napěťovou hladinu sepnutí spínače, parametr ***VOFF*** udává napěťovou hladinu rozepnutí spínače. Parametr ***VT*** je prahové napětí spínače a parametr ***VH*** je napětí hystereze.

Příklad převodu modelové řady spínače řízeného napětím z programu PSPICE do ICAP je následující.

PSPICE model příkladu:

.MODEL SMOD ***VSWITCH*** *RON = .1m ROFF = 1e15* ***VON = 1.21***
+ ***VOFF = 1.19***

ICAP model příkladu:

.MODEL SMOD ***SW*** *RON = .1m ROFF = 1e15* ***VT = 1.2***
*+****VH = .01***

Přepočtení podle výše uvedeného pravidla platí i pro spínač řízený proudem. Je ale nutné zaměnit parametry ***ION***, ***IOFF*** za ***IT***, ***IH***. Dále změnit typ modelu ***ISWITCH*** pro spínač řízený proudem na typ modelu ***CSW***.

PSPICE model příkladu:

.MODEL SMOD ***ISWITCH*** *RON = .1m ROFF = 1e15* ***ION = 1.21***
*+****IOFF = 1.19***

ICAP model příkladu:

.MODEL SMOD ***CSW*** *RON = .1m ROFF = 1e15* ***IT = 1.2***
*+****IH = .01***

2.2.6 Zpožďovací linka se ztrátami

Zpožďovací linka je v podstatě přenosové vedení, které využívá pro přenos signálů dvou párů vodičů. Proto při definování zpožďovací linky v programech SPICE se udávají čtyři uzly, kde prvý *uzel A* je spojen s prvým *uzlem B* a druhý *uzel A* je spojen s druhým *uzlem B*. Uzly A značí jeden konec zpožďovací linky a uzly B značí druhý konec. Není zde definován směr přenosu, každý uživatel si směr přenosu volí sám.

Struktura modelové řady zpožďovací linky se ztrátami v PSPICE:

T*jméno uzel A uzel A uzel B uzel B jméno modelu*

.MODEL jméno modelu typ modelu ***LEN PARAMETRY VEDENÍ***

Parametr ***LEN*** je elektrická délka vedení. ***PARAMETRY VEDENÍ*** jsou primárními parametry R, L, G a C na jednotku délky.

Pro splnění kompatibility programů PSPICE a ICAP musí být prohozen parametr ***LEN*** a ***PARAMETRY VEDENÍ***. Dále musí být na začátku řádky modelové řady změněn prvek ***T*** na ***O***.

Struktura modelové řady zpožďovací linky se ztrátami v ICAP:

O*jméno* *uzel A* *uzel A* *uzel B* *uzel B* *jméno modelu*

.MODEL *jméno modelu* *typ modelu* ***PARAMETRY VEDENÍ*** ***LEN***

2.2.7 Napětím řízený zdroj napětí

V této kapitole bude popsána změna převodu napětím řízeného zdroje napětí z programu PSPICE do ICAP. Napětím řízený zdroj napětí je zde definován pomocí tabulky modelem *TABLE*.

Definice modelu *TABLE* v ICAP:

A*jméno* *vstup* *výstup* *jméno modelu*

.MODEL *jméno modelu* ***PWL (XY_ARRAY = [****hodnoty****]***

+INPUT_DOMAIN = *0.0*

+FRACTION = FALSE)

Parametr ***PWL*** zde zastává funkci po částech lineárního zdroje. Do parametru ***XY_ARRAY*** příkazu ***.MODEL*** se zadávají vztahy mezi vstupními a výstupními hodnotami pomocí souřadnic bodů v X Y pořadí. Parametry ***INPUT_DOMAIN*** a ***FRACTION*** snižují možnost nekonvergence. Předpokládá se hladká první derivace přenosové funkce. Parametr *FRACTION* je typu Boolean, může tedy být buď ***TRUE*** nebo ***FALSE***. Pokud bude *TRUE*, do parametru *INPUT_DOMAIN* musí být zadáno desetinné číslo, jinak musí být

zadána absolutní hodnota. Parametr *INPUT_DOMAIN* vyhlazuje přenosovou funkci každým bodem zvlášť ze souřadnic bodů.

Napětím řízený zdroj napětí v PSPICE definovaný modelem *TABLE*:

E1 3 4 TABLE {V(1,2)} = (hodnoty)

V programu PSPICE je napětím řízený zdroj napětí definovaný modelem *TABLE* zadáván jiným způsobem než v ICAP. Pak postup změny napětím řízeného zdroje napětí definovaný modelem *TABLE* v ICAP vypadá následovně. Před napěťově řízený zdroj napětí je nutné přidat prvek ***A*** určující, že hodnoty budou zadávány tabulkou modelem *TABLE*. Poté stačí pouze vložit příkaz ***.MODEL*** se všemi jeho parametry z definice modelu *TABLE*. Pokud jsou datové body (*hodnoty*) v pořadí X Y, mohou být jednoduše zkopírovány z PSPICE modelu *TABLE* do parametru ***XY_ARRAY*** modelu programu ICAP.

Napětím řízený zdroj napětí v ICAP definovaný modelem *TABLE*:

***A**E1 %VD(1,2) %ID(3,4) TMOD*

.MODEL TMOD PWL (XY_ARRAY = [**hodnoty**]

+INPUT_DOMAIN = 0.0

+ FRACTION = FALSE)

Parametry ***%VD*** a ***%ID*** reprezentují diferenční hodnoty napětí a proudu mezi zvolenými uzly uvedené v závorkách. ***TMOD*** je jméno modelu. Ostatní parametry již byly popsány.

2.2.8 Napětím řízený zdroj proudu

Úprava parametrů napětím řízeného zdroje proudu je zde téměř stejná jako u napětím řízeného zdroje napětí. Význam všech změněných parametrů byl vysvětlen v kapitole 2.2.7.

Napětím řízený zdroj proudu v PSPICE definovaný modelem *TABLE*:

G1 3 4 TABLE {V(1,2)} = (hodnoty)

Pak po změně parametrů vypadá napětím řízený zdroj proudu v ICAP definovaný modelem *TABLE* následovně:

AG1 %VD(1,2) %VD(3,4) TMOD

.MODEL TMOD PWL (XY_ARRAY = [*hodnoty****]***

+INPUT_DOMAIN = *0.0*

+ FRACTION = FALSE)

Úprava u tohoto zdroje spočívá pouze v zaměnění parametrů diferenční hodnoty proudu a napětí, tedy ***%ID*** za ***%VD***.

2.2.9 Převod *E*, *G*, *S* prvků a rovnic na rovnice prvku *B*

Prvek *E* je napětím řízený zdroj napětí a prvek *G* je napětím řízený zdroj proudu. Prvek *S* je spínač s hysterezní charakteristikou. Prvek B obsahuje vylepšení, které je dáno novými možnostmi analogového chování. Nelineárně závislý zdrojový prvek *B* umožňuje přístup pro vložení algebraických rovnic, trigonometrických či transcendentních operátorů, uzlů napětí a proudů. Umožňuje také zadat funkce *IF-THEN-ELSE* a logické výrazy typu Boolean, užitečné pro různé kombinace režimů simulace.

2.2.9.1 Převod *E* a *G* prvků

Příklad *E* a *G* zdroje v PSPICE:

Etmax tmax 0 ***VALUE = {****Tr * v(tm1) + (Ts-Tr) * v(tm2)****}***

Gout 101 102 ***VALUE = {****V(201,202) * i(vsense)****}***

Převod do ICAP modelu je následující. Na začátek řádku je nutné přidat prvek ***B***. Dále v prvním řádku pro *E* zdroj, bude nahrazen parametr ***VALUE*** prvkem ***V*** a v druhém řádku parametr ***VALUE*** prvkem ***I*** pro *G* zdroj. Poté se odstraní složené závorky **{ }** před i za výrazem a umístí kolem jednoho nebo více parametrů, které budou vyhodnocovány.

Příklad *E* a *G* zdrojů v ICAP:

B*Etmax* *tmax* *0* ***V*** *= **{**Tr**}** * v(tm1) + **{**(Ts-Tr)**}** * v(tm2)*

B*Gout* *101* *102* ***I*** *= V(201,202) * i(vsense)*

2.2.9.2 Převod *S* prvku

Pro prvek *S* (spínač s hysterezí) platí převod na prvek *B* následující pravidlo. U prvku *S* je nutné přidat na začátek řádku prvek ***A*** v programu ICAP.

PSPICE model:

S1 *1* *2* *201* *0* *switch*

ICAP model:

A*S1* *1* *2* *201* *0* *switch*

2.2.10 Syntaxe funkce *IF-THEN-ELSE*

Funkce ***IF-THEN-ELSE*** se používá k výběru ze skupiny výrazů, která se mají vykonat v závislosti na vyhodnocení podmínky.

Definice syntaxe funkce ***IF-THEN-ELSE*** v programu PSPICE:

IF (*PODMÍNKA****,*** *VÝRAZ1****,*** *VÝRAZ2****)***

Pro znázornění syntaxe funkce ***IF-THEN-ELSE*** budou uvedeny dva příklady použití v programu PSPICE pro prvek *E*.

Příklad v PSPICE:

E1 1 0 ***VALUE*** *=* ***{IF (****V(3) > 5****,*** *10****,*** *100m****)}***

E1 1 0 ***VALUE*** *=* ***{IF (****V(3) > 5****,*** *5****, IF (****V(3) < 100m****,*** *100m****,*** *V(3))))}***

Definice syntaxe funkce ***IF-THEN-ELSE*** v programu ICAP:

PODMÍNKA ***?*** *VÝRAZ* ***:*** *VÝRAZ2*

Funkce ***IF-THEN-ELSE*** v obou případech programů SPICE pracuje stejně. Pokud je *PODMÍNKA* splněna, pak platí *VÝRAZ1*, jinak *VÝRAZ2*. *VÝRAZ* může být buď hodnota, nebo vzorec.

Musí být provedeny následující změny. V programu ICAP je nutné na začátek řádky přidat prvek ***B***, protože se jedná o logickou podmínku typu Boolean. Prvek *B* umožňuje vložit funkci ***IF-THEN-ELSE*** do prvku *E*. Dále parametr ***VALUE*** musí být nahrazen prvkem ***V***. Poté je ještě nutné odstranit funkci ***IF*** včetně složených závorek ***{ }*** a kulatých závorek ***( )***. Čárka za *PODMÍNKOU* je nahrazena otazníkem ***?***, a čárka za *VÝRAZEM1* je nahrazena dvojtečkou ***:***.

Upravený příklad v ICAP:

B*E1 1 0* ***V*** *= V(3) > 5* ***?*** *10* ***:*** *100m*

B*E1 1 0* ***V*** *= V(3) > 5* ***?*** *5* ***:*** *V(3) < 100m* ***?*** *100m* ***:*** *V(3)*

2.2.11 Analogové chování rovnic, výrazů a funkcí

Program ICAP používá prvky *B* a PSPICE používá prvky *G* nebo *E* s klíčovým slovem *VALUE* pro zadávání algebraických rovnic, výrazů a funkcí. V této kapitole bude popsána syntaxe PSPICE a její ekvivalent v ICAP znázorněné v tabulce 1. ICAP má mnohem více funkcí, které je možné přečíst v uživatelské příručce.

Tabulka 1 PSPICE a její ekvivalent v ICAP

Operace	PSPICE syntaxe	ICAP syntaxe
Sčítání	+	+
Odčítání	-	-
Násobení	*	*
Dělení	/	/
MOCNINA	**	^
Jednoduchý NOT	~	~
OR	|	|
XOR	^	***---NEPODPOROVÁN---***
AND	&	&
ROVNOST	==	***---NEPODPOROVÁN---***
NEROVNOST	!=	***---NEPODPOROVÁN---***
Větší než	>	>
Větší nebo rovno	>=	>=
Menší než	<	<
Menší nebo rovno	<=	<=
$\|x\|$	*ABS(x)*	*ABS(x)*
$x^{1/2}$	*SQRT(x)*	*SQRT(x)*
e^x	*EXP(x)*	*EXP(x)*
LN(X)	***LOG(X)***	***LN(X)***
log(x)	*LOG10(x)*	*log(x)* *log10(x)*
$\|x\|^y$	*PWR(x,y)*	*PWR(x,y)*
$+\|x\|^y$ *(if x > 0)* $-\|x\|^y$ *(if x < 0)*	*PWRS(x,y)*	*PWRS(x,y)*
sin(x)	*SIN(x)*	*SIN(x)*
$sin^{-1}(x)$	*ASIN(x)*	*ASIN(x)*
sinh(x)	*SINH(x)*	*SINH(x)*
cos(x)	*COS(x)*	*COS(x)*
$cos^{-1}(x)$	*ACOS(x)*	*ACOS(x)*
cosh(x)	*COSH(x)*	*COSH(x)*

tan(x)	*TAN(x)*	*TAN(x)*
$tan^{-1}(x)$	*ATAN(x)* *ARCTAN(x)*	*ATAN(x)*
$tan^{-1}(y/x)$	*ATAN2(y,x)*	*ATAN2(y,x)*
tanh(x)	*TANH(x)*	*TANH(x)*
Velikost X	*M(x)*	*M(x)* *mag(x)* *magnitude(x)*
Fáze X	*P(x)*	*P(x)* *ph(x)* *phase(x)*
Reálná část X	*R(x)*	*R(x)* *re(x)* *real(x)*
Imaginární část X	*IMG(x)*	*IMG(x)* *im(x)* *imag(x)*
DERIVACE X PODLE ČASU	***DDT(X)***	***---NEPODPOROVÁN---***
INTEGRACE X PODLE ČASU	***SDT(X)***	***---NEPODPOROVÁN---***
HODNOTY Y JAKO FUNKCE X	***TABLE(X,X1,Y1,...)***	***PŘEVOD ZDROJE E A G*** **(**2.2.9.1**)**
Minimální X a Y	*MIN(x,y)*	*MIN(x,y)*
Maximální X a Y	*MAX(x,y)*	*MAX(x,y)*
LIMIT (X,MIN,MAX)	***LIMIT(X,MIN,MAX)***	***---NEPODPOROVÁN---***
if x > 0, pak +1 *if x = 0, pak 0* *if x < 0, pak -1*	*SGN(x)*	*SGN(x)*
if x > 0, pak 1 *jinak 0*	*STP(x)*	*STP(x)*
SPLNĚNÍ PODMÍNKY T, PAK X JINAK Y	***IF(T,X,Y)***	***T ? X : Y*** **(**2.2.10**)**

2.3 Úprava NETLISTU převodníkem textového editoru

Společnost INTUSOFT vytvořila pro zjednodušení v programu ISED5, který je součástí ICAP/4, možnost převodu textové podoby vytvořeného obvodu v programu PSPICE do ICAP. V této kapitole se čerpalo z literatury [7].

Program ISED5 se otevře po spuštění souboru s koncovkou *CIR*, ale pouze v případě, že program je využíván vždy při otevírání souborů tohoto typu. Nebo vyhledáním příslušného programu v nabídce „*WINDOWS/START/PROGRAMY/ICAP_4*“, či za pomoci průzkumníka nalezením cesty uložení příslušného programu ve vašem počítači, například *C:\spice8\PR\ISED5*.

Jméno souboru s koncovkou *CIR* představuje jméno vstupního textového souboru se zadáním v syntaxi kteréhokoliv programu SPICE, vytvořený libovolným textovým editorem a uloženým s příslušnou koncovkou.

Po spuštění příslušného vstupního textového souboru v ISED5, je možné provést převod do programu ICAP. V nabídce menu „***EDIT***“ a následným spuštěním pomocí tlačítka ***[CONVERT PSPICE TO ICAP]***, nebo stisknutím klávesové zkratky (***CTRL+I***). Vytvoří se další okno s vytvořeným převedeným NETLISTEM.

Bohužel tento převod je omezen od výrobce. Funguje pouze pro analogové obvody a to ještě **NE** pro všechny PSPICE modely. Pro číslicové obvody převod není možný. Pokud je potřeba převod uskutečnit bez použití integrovaného převodníku nebo bez vlastní úpravy pro splnění kompatibility, která byla popsána v kapitole 2.2, je možné kontaktovat výrobce a po domluvě jej nechat převést.

Poté, až se vytvoří převedený NETLIST vytvořený integrovaným převodníkem a po zkontrolování provedených změn v syntaxi programu zjistíte, že NETLIST obsahuje příkaz *.LIB*. Tento příkaz definuje použité knihovny pro příslušné modely, na které se v NETLISTU odkazuje. Je tedy součástí NETLISTU v programu PSPICE a před spuštěním simulace v ICAP, je nutné tento příkaz včetně knihovny odstranit.

Po vytvoření okna s převedeným NETLISTEM pomocí integrovaného převodníku si jej můžete uložit do souboru a pojmenovat jej podle sebe s příslušnou koncovkou *CIR*.

2.3.1 Knihovny v programu PSPICE a ICAP

Program ICAP oproti PSPICE využívá jiný způsob volání knihoven. Oba způsoby volání knihoven budou popsány v následujících kapitolách. Protože ICAP neobsahuje v NETLISTU příkaz *.LIB* volající knihovny modelů jako je tomu v programu PSPICE, musí být provedeny ještě drobné změny ve vytvořeném převedeném NETLISTU. Této změny docílíte odstraněním celého příkazu *.LIB* včetně knihovny odkazující na modely součástek.

Nyní již nic nebrání tomu, aby mohla být spuštěna simulace vytvořeného obvodu bez jakýchkoliv problémů, pokud již byly provedeny změny v NETLISTU pro splnění kompatibility obou programů, který byl popsán v kapitole 2.2. Některé změny budou popsány v kapitole 2.4.

2.3.1.1 Knihovny v programu PSPICE

Nejvíce využívanou knihovnou v programu PSPICE je knihovna *NOM.LIB*. V této kapitole bude popsán její obsah, tedy odkazy na další knihovny modelů součástek. Je zde i možnost rozšířit nabídku modelů vložením další knihovny pomocí příkazu *.LIB* nebo vložením souboru pomocí příkazu *.INC*. Podrobnější popis přidání knihoven byl popsán v literatuře [4-6].

Knihovna *NOM.LIB* je standardní knihovnou v PSPICE, v níž jsou obsaženy odkazy na knihovny modelů. Jedná se o hlavní knihovní soubor, jejíž obsah všech odkazů je prohledáván až při volání výše zmíněné knihovny. Protože hlavní knihovna odkazuje na další knihovny, proces vyhledávání všech knihoven zpomaluje běh celého programu. Proto pro zrychlení procesu vyhledávání je vytvořen indexový soubor s koncovkou *IND*. Tento indexový soubor je vytvořen pokaždé. Pokud PSPICE zjistí neplatnost tohoto souboru, vytvoří jej automaticky. Nachází se v adresáři *ORCAD\CAPTURE\LIBRARY\PSPICE*.

V případě příliš dlouhé doby vyhledávání knihovny může být vhodné zakomentovat odkazy na knihovny, které nepoužíváme. Je to jednoduché, stačí pouze otevřít knihovní soubor *NOM.LIB* v programu ISED5 či v poznámkovém bloku nebo v jiném textovém editoru. Zakomentování je možné pomocí symbolu „*****“ nebo „**;**“. Je tedy možné obejít přístup na odkazy v knihovním souboru *NOM.LIB* a jednoduše zkopírovat model nebo podobvod, který chceme použít, přímo do NETLISTU.

V tomto odstavci bude popsán obsah knihovního souboru *NOM.LIB*. Tento knihovní soubor obsahuje soubor utilit podobvodů a modelů. Do utilit podobvodů a modelů patří knihovny *BREAKOUT.LIB*, kde jsou obecně definovány součástky pro schematický editor, dále *FILTSUB.LIB*, kde jsou definovány standardní filtry druhého řádu používané v NETLISTU, a nakonec *TLINE.LIB*, kde jsou definovány modely přenosové linky a podobvodů. Dále obsahuje nejvíce používané součástky, jako jsou bipolární tranzistory, diody, tyristory, JFETy, jednoduché modely ADC a DAC, digitální (logické) obvody apod. Obsahuje též i příspěvky od výrobců (dodavatelů), které jsou obsaženy v knihovně *VENDOR.LIB*, a patří sem například operační zesilovače od společnosti BURR BROWN, TEXAS INSTRUMETS, dále pokročilé lineární operační zesilovače, bipolární a unipolární tranzistory od společnosti PHILIPS, MOTOROLY, ZETEX atd. A nakonec tento knihovní soubor obsahuje evropské a japonské knihovní soubory *EUROPE.LIB* a *JAPAN.LIB*, které odkazují na knihovní soubory pro evropské a japonské modely.

2.3.1.2 Knihovny v programu ICAP

V tomto programu je způsob volání knihovních souborů odlišný oproti PSPICE. Program ICAP nepoužívá standardní knihovnu. Volání se zde uskutečňuje automaticky přeložením NETLISTU při spuštění simulace. To znamená, že není potřeba zadávat příkaz *.LIB* pro volání knihoven. Program při volání zpřístupní databázi knihoven a prohledá ji. V databázi jsou odkazy na knihovní soubory všech modelů součástek, které byly vytvořeny pro program ICAP.

Soubor obsahující cestu k adresáři knihoven *LIB.@@@* je umístěn v instalačním adresáři *SPICE8\SN*. Odkazuje na textovou podobu modelů knihovních souborů s koncovkou *LIB*. Tyto knihovní soubory jsou umístěny v instalačním adresáři *SPICE8\PR*.

Soubor obsahující seznam všech souborů sloučených symbolů *SYM.@@@* je umístěn v instalačním adresáři *SPICE8\SN*. Odkazuje na grafickou podobu modelů s koncovkou *SYM*. Tyto knihovní soubory jsou umístěny v instalačním adresáři *SPICE8\SN\SYMBOLS*.

Soubory *DBASE.@@@* a *INDEX.@@@* jsou vytvořeny po sestavení databáze symbolů a knihoven. Soubor *DBASE.@@@* je vytvořen řádkou **SRC* v knihovním souboru s koncovkou *LIB*. Soubor *INDEX.@@@* obsahuje název každého modelu a podobvodu.

Knihovní soubor s koncovkou *LIB* obsahuje model nebo podobvod a k němu odpovídající symbol **SYM*.

Nové knihovny do programu ICAP mohou být přidány dvěma způsoby. Prvním způsobem je možnost zadání příkazu **INCLUDE* v NETLISTU vytvořeného programu. Tento příkaz musí být zadán s cestou uložení daného knihovního souboru, který chceme zpřístupnit. Tím se automaticky zpřístupní knihovní soubory všech modelů součástek nebo podobvodů uložené v dané knihovně. Druhou možností je vložení knihovních souborů a symbolů do příslušných instalačních adresářů, kde se program na ně odkazuje. Poté musí být spuštěn program ICAP a z nabídky menu „***FILE***“ tlačítkem [***UPDATE PART DATABASE…***] bude provedena aktualizace databáze knihoven a zpřístupní se dané knihovní soubory a symboly.

2.4 Modelování akumulačních prvků s proměnnými parametry

Modelování prvků a následná simulace je efektivním způsobem analýzy elektronických obvodů. V programech typu SPICE je možnost modelovat aktivní i pasivní součástky a simulací zjistit či ověřit jejich funkce a parametry. Jedná se o behaviorální modelování, tedy chování součástky při možnosti změn vlastních parametrů.

Dotazy, týkající se modelování a počítačové analýzy elektrických obvodů, jsou tématem na nejrůznějších diskusních fórech. K častým dotazům patří otázka, zda je možné v programu PSPICE modelovat lineární akumulační prvky s časově proměnnými kapacitami a indukčnostmi.

Z učebnic a manuálů programu PSPICE se dozvíte pouze o způsobu přímého modelování těchto prvků pro konstantní parametry L a C. Vzniká pak problém, chceme-li provést časovou analýzu parametrických obvodů paměťových prvků, které mění své parametry podle zadaných funkcí času.

Tyto funkce mohou být zadány dvěma způsoby. Prvním způsobem je zadání pomocí analytických vzorců a druhým je možnost zadání množinami bodů z měřících procesů.

V následujících kapitolách proběhnou ukázky možností modelování akumulačních prvků s proměnnými parametry v programu PSPICE a možnosti převodu vytvořeného programu do programu ICAP. V těchto kapitolách bylo čerpáno z literatury [2-6] [8][9].

V příloze na CD jsou uloženy vytvořené vstupní textové soubory pro časově proměnnou kapacitu s názvem *C_promenna.cir* a pro časově proměnnou indukčnost *L_promenna.cir*. Rovněž jsou zde uloženy výsledné grafy pod názvem *C_promenna.pdf* a *L_promenna.pdf*.

2.4.1 Modelování akumulačních prvků s proměnnými parametry v PSPICE

V programu PSPICE lze L a C prvky s proměnnými parametry modelovat pomocí řízených zdrojů. Časové derivace obvodových proměnných, tedy časově proměnné indukčnosti L a kapacity C, obsažené v rovnici (1) nejsou vhodné pro modelování v PSPICE. Příčinou můžou být numerické chyby a konvergenční problémy.

$$u_L = \frac{d}{dt}\big(L(t)i_L(t)\big), i_C = \frac{d}{dt}\big(C(t)u_C(t)\big) \tag{1}$$

Existuje jednoduchý postup jak překonat problémy zmíněné v předchozím odstavci včetně variant definování počáteční podmínky pro induktory a kondenzátory, kterými jsou proud induktorem a napětí na kondenzátoru v čase 0. Lze použít jednoduchý trik a to takový, že bude provedena integrace obou stran rovnic (1) podle času a tím se získá vhodnější model pro behaviorální modelování. Tento model pak navíc obsahuje počáteční hodnoty obvodových proměnných $i_L(0)$ a $u_C(0)$:

$$i_L = \frac{L(0)i_L(0) + \int_0^t u_L(t)dt}{L(t)}, L(t) \neq 0, \tag{2}$$

$$u_C = \frac{C(0)u_C(0) + \int_0^t i_C(t)dt}{C(t)}, C(t) \neq 0, \tag{3}$$

Z rovnic (2) a (3) je vidět, že induktor může být modelován řízeným zdrojem proudu a kondenzátor řízeným zdrojem napětí. Úplný výpis včetně popisů modelů časově proměnného induktoru a kondenzátoru v PSPICE jsou uvedeny v následujících kapitolách.

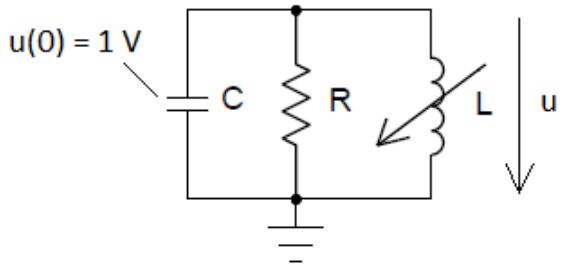
Oba modely mají dvě části. První část definuje parametry *L* a *C* jako funkce času. V druhé části jsou pak modelovány rovnice (2) a (3) pomocí řízených zdrojů typu *G* a *E*. Program PSPICE umožňuje počítat časové integrály pomocí funkce *SDT()*.

2.4.1.1 Modelování časově proměnné indukčnosti

Induktor s časově proměnnou indukčností je připojen paralelně ke kondenzátoru o fixní kapacitě 7,5 nF a rezistoru o odporu 100 kΩ. Pokud by indukčnost byla konstantní o hodnotě 54 uH, pak pomocí Thompsonova vztahu (4) se dá zjistit rezonanční frekvence, která činí 250 kHz. Na počáteční podmínku *u(0)* = 1 V, odpovídající nabitému kondenzátoru na počáteční napětí 1 V, by rezonanční obvod *RLC* reagoval formou tlumených harmonických kmitů.

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \tag{4}$$

Mění-li se indukčnost periodicky v čase s opakovací frekvencí 500 kHz, dochází k dodatečnému „pumpování“ energie do obvodu. V příloze I je znázorněna situace, kdy dochází k rovnováze mezi ztrátami, vyvolanými disipací energie na rezistoru, a výše zmíněnou energií. Do obvodu musely být připojeny diody, které udržují stálou napěťovou hladinu RLC obvodu. Výsledkem jsou pak ustálené kmity napětí a proudu. Počítačovou simulací lze snadno ověřit, že tato rovnováha bude narušena při změně frekvence, nebo odpojením jedné z diod, nebo změnou hodnoty prvku L a C. Příklad aplikačního RLC obvodu je znázorněn na obrázku 2.1.



Obrázek 2.1 Paralelní rezonanční RLC obvod

Výpis programu pro modelování časově proměnného induktoru:

```
*Casove promenna indukcnost
.lib nom.lib
*
R    1   0   100k
D1   1   0   D1N4148
D2   0   1   D1N4148
C    1   0   7.5nF   IC=1
*---------volani podobvodu
X    1   0   inductor_var
*---------zatek podobvodu
.subckt   inductor_var   + -   params: IL=0
.param   f0=500kHz
.func   L(time) {54u*(1+0.8*sin(2*3.1416*f0*time))}
*
GL   + -   value={(sdt(V(+,-))+L(0)*IL)/L(time)}
.ends
*---------konec podobvodu
.TRAN   1n  50u   UIC
.PROBE
.END
```

Podobvod je označen ve výpisu programu začátkem a koncem. Ten je volán s parametrem *IL*, který definuje počáteční proud induktorem. Modelování časově proměnného induktoru bude uvedeno pro ilustraci definicí časově proměnné indukčnosti pomocí analytického vzorce, který je volán v programu funkcí *.FUNC L(TIME)*. Řízený zdroj proudu je v programu podobvodu zadán pomocí rovnice (2). Následující rovnice se používá pro modelování časově proměnné indukčnosti:

$$L(t) = L_0(1 + 0.8\sin(2\pi f_0 t)) \tag{5}$$

Počáteční podmínka pro nabitý kondenzátor počátečním napětím 1 V, je zde zadána v rámci definice kondenzátoru C. Za velikost hodnoty kapacity se přidá příkaz *IC (Initial Conditions)*, který umožňuje zadat počáteční podmínku.

Časová analýza pro zobrazení výsledků počítačové simulace příkazem *.TRAN* je nastavena na sledování do času 50µs. Časový krok je zde zadán libovolně 1ns. Pokud je uvedena volba *UIC (Use Initial Conditions)*, počáteční podmínka je dána deklarací C.

2.4.1.2 Modelování časově proměnné kapacity

Zde nastává změna oproti programu pro časově proměnnou indukčnost. Výpis časově proměnného kondenzátoru obsahuje model časově proměnné kapacity ve formě tabulky změřených hodnot. Pokud by kapacita byla konstantní o hodnotě 7,5 nF, rezonanční frekvence by byla 250 kHz, stejná jako v případě časově proměnné indukčnosti, pak pomocí Thompsonova vztahu (4) zjistíme indukčnost, která činí 54 µF. Hodnota rezistoru zůstává totožná, tedy jeho odpor činí 100 kΩ.

Výpis programu pro modelování časově proměnného kondenzátoru:

```
*Casove promenna capacita
.lib "nom.lib"
*
L     2    0   54uH IC=1mA
R     1    0   100k
R1    1    2   10
D1   1    0   D1N4148
D2   0    1   D1N4148
*---------volani podobvodu
X    1    0  capacitor_var
*---------zacatek podobvodu
.subckt   capacitor_var  +  -  params: UC=0
.param    C0  5nF
Vc    c    0   PWL
+ REPEAT FOREVER
+ (1ns, {C0})
+ (1us, 5nF)
+ (1.001us, 10nF)
+ (2us, 10nF)
+ ENDREPEAT
*
Ec  +  -  value={(sdt(I(Ec))+C0*UC)/V(c)}
.ends capacitor_var
*---------konec podobvodu
.TRAN     1n    70u    UIC
.PROBE
.END
```

Pokud bude stejné zapojení, které bylo použito pro časově proměnnou indukčnost (obrázek 2.1), program by ohlásil chybu „*Voltage source and/or inductor loop involving L*“. Důvodem je připojení zdroje paralelně s induktorem L. Toto není povoleno, protože induktor

bude nutit nulové napětí přes nenulový zdroj napětí. Vzniká tak kolize a program ohlásí chybovou hlášku. Problém lze vyřešit přidáním rezistoru R1 o malé hodnotě do série s induktorem L. Diody, stejně jako v předchozí kapitole, mají zde omezovací charakter, protože se jedná o RLC obvod s proměnnou kapacitou.

Počáteční podmínka pro proud tekoucí induktorem je zde nastavena na hodnotu 1mA v rámci definice induktoru L zadáním za velikost hodnoty induktoru parametrem *IC*.

Podobvod je označen ve výpisu programu začátkem a koncem. Ten je volán s parametrem *UC*, který definuje počáteční napětí kondenzátoru v čase 0.

Časový průběh kapacity je zde definován jako signál nezávislého po částech lineárního zdroje napětí *Vc* typu *PWL* pro docílení periodičnosti funkce *C(t)*.

Časová analýza pro zobrazení výsledků počítačové simulace příkazem *.TRAN* je nastavena na sledování do času 70us. Počáteční podmínka proudu tekoucí induktorem je povolena volbou *UIC*.

Kapacita se skokovými změnami hodnot po 1ns mění svoji hodnotu mezi 5 nF a 10 nF v časových okamžicích (1, 2, 3, …) µs. Výsledkem je pak periodičnost pro časově proměnnou kapacitu měnící se mezi 5 nF a 10 nF, znázorněno v příloze I.

2.4.2 Modelování akumulačních prvků s proměnnými parametry v ICAP

Program ICAP pracuje trochu odlišněji než PSPICE. Zadávání časově proměnných prvků zde nemůže probíhat pomocí interní integrální funkce. Program ICAP totiž tuto funkci nepodporuje, stejně tak nepodporuje funkci derivace (viz tabulka 1). Tyto funkce totiž zjednodušují celé řešení pro modelování prvků L a C s časově proměnnými parametry. Modelování prvků L a C musí být řešeno tedy jiným způsobem. Bohužel tento problém se nepodařilo zcela vyřešit. Bude zde popsán jen způsob převodu do programu ICAP a úpravy v NETLISTU pro splnění kompatibility obou programů.

Pokud je potřeba ověřit kompatibilitu programu PSPICE a ICAP, je nutné nejdříve upravit části programu, které program ICAP nepodporuje. Tato úprava zde bude popsána.

Vstupní textový soubor, který bude přeložen do programu ICAP, musí být otevřen v programu ISED5. Komentář je možný pomocí symbolu „*****“ nebo „**;**“. Vše, co program ICAP nepodporuje, musí být buď zakomentováno, nebo smazáno. Program totiž po převedení NETLISTU smaže všechny komentáře, kromě první řádky, kterou program ignoruje, protože slouží k popisu názvu programu. Pro oba případy s časově proměnnými parametry musí být smazán příkaz ***.LIB***, který slouží k volání knihoven v programu PSPICE, protože program ICAP tento příkaz nepodporuje (viz kapitola 2.3.1.2). Pokud potřebujeme definovat konstantu či proměnnou pomocí příkazu ***.PARAM***, musí být přiřazení hodnoty či rovnice provedeno rovnítkem **=**.

V případě induktoru s časově proměnnými parametry musí být v NETLISTU smazána definice řízeného zdroje, protože ICAP nepodporuje funkci ***SDT()*** (viz tabulka 1). Dále se v původním programu v PSPICE nachází příkaz ***.FUNC***. Tento příkaz program ICAP nepodporuje, musí být tedy také smazán před převodem NETLISTU. Místo příkazu *.FUNC* je v programu ICAP možno zadat rovnice přímo v deklaraci dané součástky, což velice zjednodušuje psaní programu. Stačí tedy zadat příslušný vzorec či rovnici za definici uzlů součástky. Nabízí se zde možné řešení zadání algebraické rovnice přímo v deklaraci dané součástky. Bohužel nebylo zjištěno z manuálů od výrobce, zda v programu ICAP je možné zadat pouze algebraickou rovnici deklarací dané součástky a program by již sám tuto rovnici použil pro výpočet integrálních funkcí. Tímto způsobem byl vytvořen převedený NETLIST v programu ISED5, ale výsledky nebyly stejné. Problém tedy bohužel nebyl vyřešen.

V případě kondenzátoru s časově proměnnými parametry při definici zdroje *Vc* typu *PWL* nastává změna, protože program ICAP neumožňuje zadat nekonečný cyklus pomocí příkazu ***REPEAT FOREVER*** a ukončit jej příkazem ***ENDREPEAT***. V programu ICAP se zadává příkazem ***REPEAT*** na konci parametrů zdroje.

Po splnění výše uvedených podmínek může být NETLIST převeden do programu ICAP. Je to velice jednoduché, stačí již v otevřeném a upraveném vstupním textovém souboru programu ISED5 spustit tlačítko ***[CONVERT PSPICE TO ICAP]*** v menu „***EDIT***“. Program tak vytvoří další okno s převedeným NETLISTEM.

V NETLISTU se vytvoří řádek * *Translated from PSpice to ICAP format by IsEd5.exe* hned pod názvem programu, který značí, že převod proběhl z programu PSPICE do ICAP úspěšně. Dále je vytvořen řádek **#SAVE all*, který uloží všechny napětí a proudy v obvodu.

Příkaz *.PROBE/CSDF* byl již popsán v kapitole 2.1. Slouží k uložení všech definovaných dat do výstupního textového souboru pomocí **#SAVE*.

3 Analýza poruchových stavů

Dlouhodobými a stále doplňovanými zkušenostmi železničních správ a výrobců zabezpečovacích zařízení byly postupně vytvářeny různé katalogy poruch pro širokou škálu stavebních prvků, určující uvažované a neuvažované poruchy těchto prvků pro aplikaci v zabezpečovacích obvodech. Uvažované a neuvažované poruchy jsou definovány v katalogu poruch normou ČSN EN 50129. Neuvažované poruchy jsou poruchy nepravděpodobné. Vznik těchto poruch je vyloučen fyzikálními důvody nebo nesplněnými technologickými opatřeními (viz norma ČSN EN 50129). Uvažované poruchy budou aplikovány na zadaném obvodu.

V této části práce budou popsány možnosti analýzy poruchových stavů pro obvod kapacitního dekodéru. Budou zde popsány principy zajištění technické bezpečnosti v železniční zabezpečovací technice. Dále princip kapacitního dekodéru jakožto dekodéru impulsního signálu a nakonec zde budou provedeny analýzy poruchových stavů v programu PSPICE a ICAP, a srovnání jejich výsledků.

Program PSPICE, na rozdíl od programu ICAP/4, neumožňuje analýzu poruchových stavů prvků v obvodu. Proto byly v programu PSPICE vytvořeny modely součástek, které možnosti analýzy poruchových stavů dovolují. Tyto modely byly popsány v literatuře [10][11]. Do programu ICAP byl pak proveden převod těchto modelů, který je popsán v literatuře [1]. Výhodou programu ICAP je možnost realizovat poruchové stavy na prvcích obvodu přímo. V dalším textu budou možnosti analýzy poruchových stavů probrány.

3.1 Principy zajištění technické bezpečnosti

V železniční zabezpečovací technice je kladen velký důraz na bezpečnost. Bezpečné chování zařízení při jakékoliv možné poruše tohoto zařízení je nutnou podmínkou při návrhu. Při poruše zařízení se musí zařízení zachovat tak, aby neohrozilo bezpečnost železniční dopravy. Musí být splněn požadavek na technickou bezpečnost, tj. předepsané chování zařízení při poruchách.

Při výchozím předpokladu zabezpečovací techniky, že ***v jednom okamžiku může vzniknout pouze jedna nezávislá porucha***, lze požadavek na bezpečnou konstrukci vyjádřit tzv. *zabezpečovací šesterkou*:

- žádnou poruchou nesmí dojít k ohrožení bezpečnosti jízdy vlaků. K ohrožení nedojde, pokud výstupy zařízení nebudou ani při poruše povolovat vlakům větší volnost než odpovídá stavu vstupů,
- každá porucha se musí vhodně a dostatečně rychle (s přihlédnutím k četnosti poruch) projevit, aby bylo možné vyloučit, že se objeví jakákoliv další porucha, která by v kombinaci s poruchou první mohla ohrozit bezpečnost. Riziko je zde funkcí velikosti časového intervalu a četnosti poruch. Proto by detekce chyb měla být nezávislá na toku informací. U klasických zařízení (bez mikroprocesorů) se za přijatelnou ale obvykle považuje i detekce až při následné činnosti či obsluze zařízení,
- není-li některá porucha detekována ve smyslu předchozího odstavce, je nutné předpokládat vznik jakékoliv další poruchy, přičemž současné působení obou těchto poruch se musí projevit ve smyslu předchozích odstavců,
- pokud by vlivem jedné poruchy mohlo dojít ke vzniku následných (závislých) poruch, je nutné uvažovat také všechny kombinace těchto poruch jako poruchu jednu,
- po detekci poruchy by mělo bezprostředně samočinně dojít k odstavení vadného zařízení nebo vadné části zařízení. Podle povahy zařízení lze ale za vhodný projev poruchy považovat i zastavení nebo podstatné (a tedy nepřehlédnutelné) omezení železničního provozu. V každém případě však výstupy vadného zařízení (vadné části) musí zůstat nebo neprodleně přejít do stavu, který neohrožuje bezpečnost dopravy (negace poruchy),
- po odstavení zařízení pro poruchu nesmí ani další poruchou dojít k samovolnému obnovení funkce. K obnovení funkce zařízení odstaveného pro poruchu může dojít až za účasti udržujícího pracovníka nebo po jiném bezpečném zjištění, že zařízení je bez chyb.

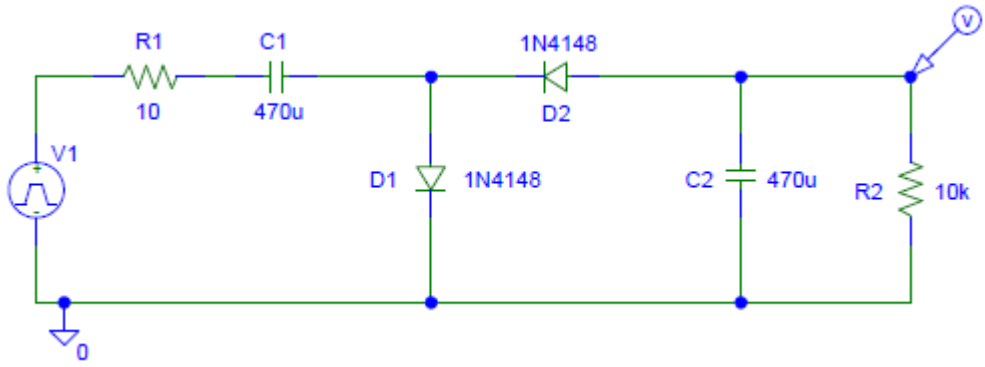
Splnit požadavky na bezpečnou konstrukci je možné různými způsoby. V zásadě lze rozlišit tři základní principy reagující na skutečnost, zda lze u některých konstrukčních prvků určité poruchové stavy vyloučit či nikoliv.

Tyto principy jsou uvedeny v literatuře [12].

3.2 Kapacitní dekodér

Jedná se o dekodér impulsního elektrického signálu. Zapojení lze využít nejen v železniční zabezpečovací technice, ale i v aplikacích, kde se vyžaduje vysoký stupeň funkčního zabezpečení elektronických obvodů.

Na tomto obvodu budou realizovány poruchové stavy *přerušení* a *zkrat* všech zapojených součástek. Je zde také možnost realizace *zkrat na pouzdro*, ale tento poruchový stav nebude zde realizován. Zapojení kapacitního dekodéru použitého v programu PSPICE je na obrázku 3.1.



Obrázek 3.1 Kapacitní dekodér v PSPICE

Na vstup obvodu je přiveden impulsní elektrický signál. Diody D1 a D2 mají usměrňující účinek. Zdroj *V1* zastává funkci pulzního zdroje typu *VPULSE*. Ve fázi, kdy zdroj *V1* dodává signál do obvodu, je kondenzátor C1 nabíjen přes diodu D1. V druhé fázi, kdy zdroj *V1* má nulovou hodnotu, kondenzátor C1 se vybíjí přes diodu D2 a je nabíjen kondenzátor C2. Kondenzátory C1 a C2 jsou nabíjeny a vybíjeny přes diody stejnosměrným napětím. Na výstupu obvodu rezistoru R2 je inverzní polarita stejnosměrného napětí, tedy na horním uzlu prvku R2 je záporný pól napětí. Místo zatěžovacího rezistoru R2 se většinou připojuje relé (viz kapitola 3.3.3), které zde bude dále uvažované.

Pokud vznikne například porucha některého obvodového prvku nebo ztráta budícího impulsního vstupního signálu, dojde k bezpečnému chování výstupního signálu. Pro splnění požadavků na bezpečné chování obvodu kapacitního dekodéru (viz kapitola 3.1) se musí výstup projevit takovým způsobem, aby došlo buď k zániku výstupního signálu, nebo výstupní signál nepřesáhne hodnotu přítahu relé.

Uvažované poruchy pro rozbor bezpečnosti obvodu kapacitního dekodéru, vzhledem k normě ČSN EN 50129, jsou zobrazeny v tabulce 2.

Tabulka 2 Uvažované poruchové stavy

Prvek	Realizované poruchy
C	*přerušení a zkrat*
D	*přerušení a zkrat*

V obou programech budou řešeny poruchové stavy pro vstupní impulsní elektrický signál s hodnotou 24 V a s frekvencí kmitání 5 Hz. Všechny výsledky jsou uloženy na přiloženém CD.

Princip kapacitního dekodéru byl vysvětlen v literatuře [13].

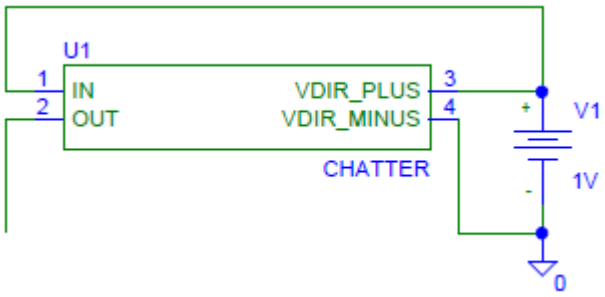
3.3 Poruchové stavy v programu PSPICE

Popis vytvořených modelů součástek již byl popsán v literatuře [10][11]. V této kapitole bude popsán způsob zapojení vytvořených modelů součástek do obvodu kapacitního dekodéru. Dále bude provedena realizace poruchových stavů na těchto vytvořených modelech a nakonec zhodnocení výsledků chování kapacitního dekodéru.

Pro názornost zde bude uveden výsledek chování obvodu kapacitního dekodéru pouze pro prvek D2 s poruchovými stavy *přerušení* a *zkrat*. Ostatní výsledky pro všechny prvky obvodu jsou na přiloženém CD.

3.3.1 Model *CHATTER* v PSPICE

Vytvořený model *CHATTER* slouží k realizování poruchových stavů *přerušení* a *zkrat*. Model je v podstatě napětím řízený spínač nebo rozpínač, který je řízen pomocí vnějšího řídícího stejnosměrného napětí s velikostí 1 V. Tento podobvod byl popsán v literatuře [1][10][11]. Zde bude ukázáno, jakým způsobem se musí zapojit jednotlivé vývody, aby mohly být realizovány poruchové stavy.



Obrázek 3.2 Zapojení obvodu CHATTER

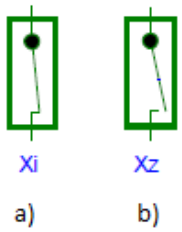
Na obrázku 3.2 je vidět zapojení modelu podobvodu *CHATTER*. Vývody *VDIR_PLUS* a *VDIR_MINUS* se připojí na stejnosměrný zdroj napětí *V1* typu *VDC*. Vstup *IN* je spojen s vývodem *VDIR_PLUS*, je tedy připojen na zdroj napětí *V1*. Vývod *OUT* modelu *CHATTER* slouží jako výstup pro realizaci poruchových stavů *přerušení* a *zkrat*. Připojí se na příslušný vývod poruchových stavů, které jsou popsány v následující kapitole.

3.3.2 Poruchové stavy *přerušení* a *zkrat* v PSPICE

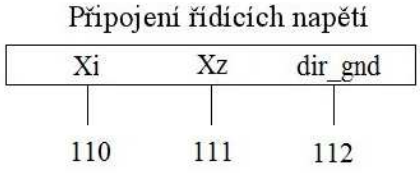
Poruchový stav *přerušení* je zde realizován pomocí rozpínacího kontaktu x_i. Řídící napětí se připojuje mezi svorky *110* a *112* vytvořených modelů součástek. Rozpínací kontakt je na obrázku 3.3.

Poruchový stav *zkrat* je zde realizován pomocí spínacího kontaktu x_z. Řídící napětí se připojuje mezi svorky *111* a *112* vytvořených modelů součástek. Spínací kontakt je na obrázku 3.3.

Svorka *112* slouží jako společnou zem pro všechna řídící napětí. Je zde označena jako *dir_gnd*. Poruchový stav *zkrat na pouzdro* (svorka *100*) se realizuje připojením na společnou zem a změnou hodnoty odporu *R_casex*.



Obrázek 3.3 Schematické značky pro obvod CHATTER: a) přerušení, b) zkrat



Obrázek 3.4 Připojení řídících napětí obvodu CHATTER

V této kapitole bylo čerpáno z literatury [1][10][11].

3.3.3 *Přerušení* a *zkrat* kapacitního dekodéru v PSPICE

V předchozích kapitolách byl vysvětlen princip zapojení poruchových stavů pro vytvořené modely součástek. V této kapitole bude proveden rozbor bezpečnosti poruchových stavů *přerušení* a *zkrat* na příkladu kapacitního dekodéru zvoleného prvku.

Pro demonstraci poruchových stavů byla zvolena dioda D2. Nejdříve bude realizován poruchový stav *přerušení* a poté *zkrat*. V příloze II jsou ukázány zapojení modelů součástek v obvodu kapacitního dekodéru a výsledky simulací pouze pro poruchový stav *přerušení* a *zkrat* diody D2. Zapojení obvodu a výsledky pro všechny poruchové stavy je uloženo na přiloženém CD.

Při poruchovém stavu *zkrat* na diodě D2, má výstupní signál pulsující charakter. Pokud by bylo připojeno na zátěž místo rezistoru R2 relé, nebude vybuzeno. Pak obvod se chová bezpečně z hlediska zabezpečovací techniky. K vybuzení může dojít v případě přivedení stejnosměrného signálu nebo střídavým signálem, pokud je na něj relé dimenzováno. Toto chování obvodu platí pro jakýkoliv pulsující signál.

Při poruchovém stavu *přerušení* na diodě D2, napětí výstupního signálu vzroste na ustálenou hodnotu -2,3 µV (princip kapacitního dekodéru byl popsán v kapitole 3.2). Hodnota však nezpůsobí přítah relé, například pro relé NMŠ1-1500T bylo změřeno v literatuře [13] napětí přítahu 15,6 V. Relé tedy nebude vybuzeno a obvod se chová bezpečně z hlediska zabezpečovací techniky. Toto chování obvodu platí pro jakýkoliv výstupní signál menší, než je napětí přítahu relé.

Výsledné chování obvodu kapacitního dekodéru při projevu poruchového stavu *přerušení* a *zkra*t jsou znázorněny v tabulce 3. Chování obvodu při poruše bylo popsáno v předchozích odstavcích.

Vývod *OUT* modelu *CHATTER* je připojen na požadovanou svorku poruchového stavu, který chceme realizovat. Pro realizaci poruchového stavu *přerušení*, musí být vývod *OUT* spojen se svorkou *110*. Pro poruchový stav *zkrat*, vývod *OUT* se musí spojit se svorkou *111*. Svorka *112* a *100* je spojena se zemí.

Tabulka 3 Simulované poruchy vytvořených prvků obvodu v PSPICE

Prvek	Porucha	Projev poruchy na výstupu	Zhodnocení projevu poruchy
C1	*přerušení*	*pokles napětí na 1,2µV*	*Bezpečné*
C1	*zkrat*	*pokles napětí na 28µV*	*Bezpečné*
D1	*přerušení*	*ustálené napětí na -2µV*	*Bezpečné*
D1	*zkrat*	*pulsující napětí od 0,45pV do -1,6pV*	*Bezpečné*
C2	*přerušení*	*pulsující napětí od 0V do -23V*	*Bezpečné*
C2	*zkrat*	*špičky napětí do -2µV*	*Bezpečné*
D2	*přerušení*	*vzrůst napětí na -2,3µV*	*Bezpečné*
D2	*zkrat*	*pulsující napětí od 0,8V do -11,4V*	*Bezpečné*

Bezpečný projev poruchy splňuje požadavek na bezpečné chování obvodu (viz kapitola 3.2). Nebezpečný projev poruchy tento požadavek nesplňuje, a proto obvod nemůže být použit jako zařízení s bezpečným projevem poruchy v zabezpečovací technice železniční dopravy. Žádnou poruchou v tabulce 3 nedojde k nebezpečnému chování obvodu kapacitního dekodéru.

V této kapitole bylo čerpáno z literatury [1][10][11].

3.4 Poruchové stavy v programu ICAP

V této kapitole bude popsán způsob zapojení vytvořených modelů součástek do obvodu kapacitního dekodéru. Dále bude popsán způsob realizace poruchových stavů na těchto vytvořených modelech a způsob realizace poruchových stavů na prvcích obvodu přímo. Nakonec budou zhodnoceny výsledky chování zadaného obvodu kapacitního dekodéru pro všechny poruchové stavy prvků obvodu.

Pro názornost zde bude uveden příklad chování obvodu kapacitního dekodéru pouze pro jeden prvek obvodu s poruchovými stavy *přerušení* a *zkrat*. Zapojení obvodu a výsledky pro všechny poruchové stavy je uloženo na přiloženém CD.

3.4.1 Model *CHATTER* v ICAP

Princip vytvořeného modelu *CHATTER* v programu PSPICE byl popsán v kapitole 3.3.1. Do programu ICAP byl proveden převod popsaný v literatuře [1].

Tento převedený model do programu ICAP se liší oproti vytvořenému v PSPICE. Zapojení modelu podobvodu CHATTER je stejné jako je na obrázku 3.2. Rozdíl je v připojení vývodu *OUT* pro realizaci poruchových stavů *přerušení* a *zkrat*. Model *CHATTER* je totiž v programu ICAP vytvořen pouze rozpínacím kontaktem a spínací kontakt zde není uvažován. Odkazuje se na model rozpínacího kontaktu řízeného napětím, který je definován v knihovním souboru *DEVICE.LIB* typem *PSW2*.

Model *CHATTER* byl převeden do programu ICAP a popsán v knihovním souboru s názvem *DIPLOMKA_PF.LIB* v literatuře [1]. Parametry *RON* a *ROFF*, které byly popsány v kapitole 2.2.5, jsou přiřazeny parametrům *Rmin* a *Rmax* při volání tohoto modelu. Po spuštění simulace poruchových stavů s původními parametry vytvořeného knihovního souboru, program ICAP nahlásil chybu. Musely být tedy definovány tyto parametry jiným způsobem. Pomocí příkazu *.PARAM* bylo možné odstranit tuto chybu a následně spustit simulaci bez problémů. Problém byl vyřešen. Takto změněný knihovní soubor s názvem *diplomka.lib* je uložen na přiloženém CD.

3.4.2 *Přerušení* a *zkrat* kapacitního dekodéru v ICAP

V této kapitole bude proveden rozbor bezpečnosti poruchových stavů *přerušení* a *zkrat* na příkladu kapacitního dekodéru zvolené součástky. Dále zde budou popsány změny ve vytvořených knihovních souborech použité pouze pro případ kapacitního dekodéru. Nakonec bude popsána realizace poruchových stavů *přerušení* a *zkrat*.

Ve vytvořených knihovních souborech, které jsou popsány v literatuře [1], jsou odkazy na volání knihovního souboru modelu *CHATTER*. Volání knihoven v programu ICAP již bylo popsáno v kapitole 2.3.1.2. V knihovních souborech *DIODA_OF.LIB*, *KAPACITOR_PF.LIB* a *REZISTOR_PF.LIB* nemusí být volán knihovní soubor *diplomka.lib* příkazem **INCLUDE*, proto může být celý řádek smazán (viz kapitola 2.3.1.2). Změněné knihovní soubory jsou přiloženy na CD včetně originálů vytvořených v literatuře [1].

Pro realizaci poruchových stavů byla zvolena dioda D2. Nejdříve bude realizován poruchový stav *přerušení* a poté *zkrat*. V příloze II jsou ukázány zapojení modelů součástek v obvodu kapacitního dekodéru a znázorněny výsledky simulací pouze pro poruchový stav *přerušení* a *zkrat* diody D2.

V programu ICAP je realizace poruchových stavů odlišná oproti programu PSPICE. Jedná se o rozdílné zapojení vývodu *OUT* modelu *CHATTER* pro připojení na požadovanou svorku poruchového stavu, který bude realizován. Pro poruchový stav *přerušení*, musí být vývod *OUT* spojen se svorkou *110* a *111*. Pro poruchový stav *zkrat*, musí být vývod *OUT* spojen se všemi svorkami a ty jsou spojeny na společnou zem. Toto zapojení je nutné, protože realizace modelu *CHATTER* je zde odlišná oproti programu PSPICE (viz kapitola 3.3.3).

Z této možnosti zapojení vyplývá, že kontakt realizující poruchový stav *zkrat* není zde zapojen jako spínací, nýbrž jako kontakt rozpínací (viz kapitola 3.3.2). Má tedy stejnou funkci zapojení pro realizování poruchového stavu *přerušení*.

Chování obvodu kapacitního dekodéru při projevu poruchového stavu *přerušení* a *zkra*t bylo popsáno v kapitole 3.3.3. Zhodnocení simulovaných poruch na prvcích obvodu je znázorněno v tabulce 4. Výsledky poruchových stavů *přerušení* a *zkrat* jsou zobrazeny v příloze II.

Tabulka 4 Simulované poruchy vytvořených prvků obvodu v ICAP

Prvek	Porucha	Projev poruchy na výstupu	Zhodnocení projevu poruchy
C1	*přerušení*	*vzrůst napětí na -50mV*	*Bezpečné*
C1	*zkrat*	*vzrůst napětí na -4,5mV*	*Bezpečné*
D1	*přerušení*	*vzrůst napětí na -230mV*	*Bezpečné*
D1	*zkrat*	*vzrůst napětí na -1,05V*	*Bezpečné*
C2	*přerušení*	*pulsující napětí od 0V do -23V*	*Bezpečné*
C2	*zkrat*	*špičky napětí do -1,4V*	*Bezpečné*
D2	*přerušení*	*vzrůst napětí na -230mV*	*Bezpečné*
D2	*zkrat*	*pulsující napětí od 0,8V do -11,5V*	*Bezpečné*

Bezpečný projev poruchy splňuje požadavek na bezpečné chování obvodu (viz kapitola 3.2). Nebezpečný projev poruchy tento požadavek nesplňuje, a proto obvod nemůže být použit jako zařízení s bezpečným projevem poruchy v zabezpečovací technice železniční dopravy. Žádnou poruchou v tabulce 4 nedojde k nebezpečnému chování obvodu kapacitního dekodéru.

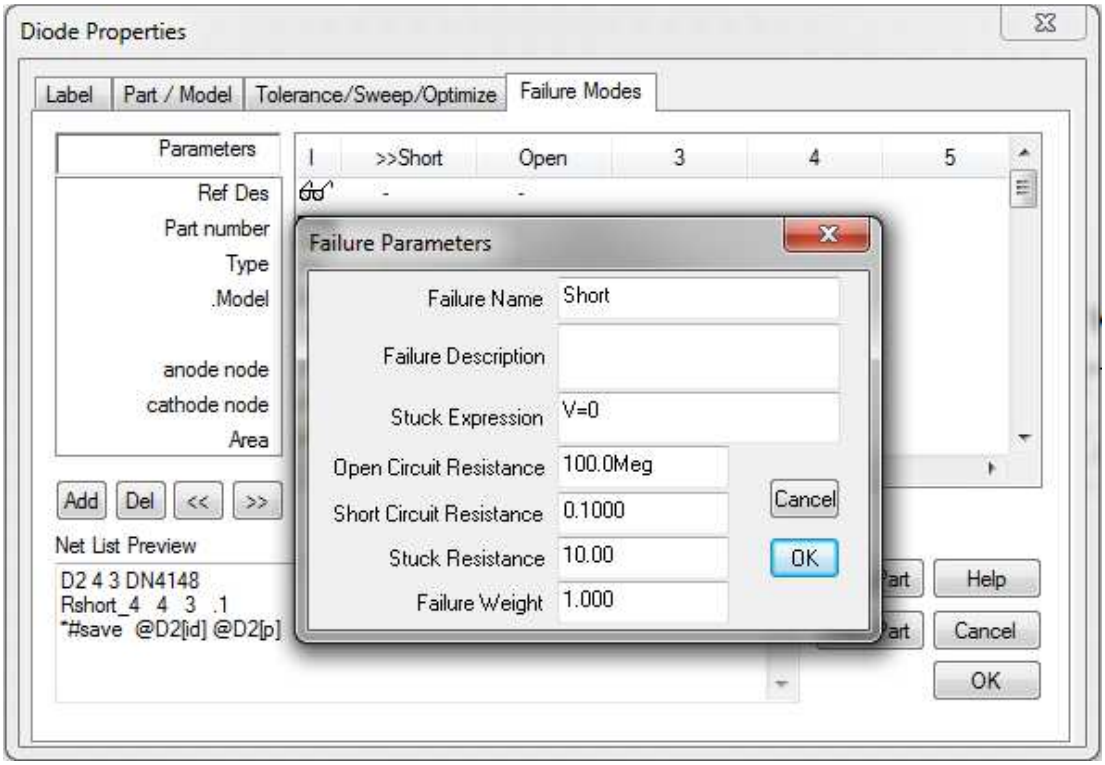
3.4.3 Interní mód poruchových stavů programu ICAP

Program ICAP umožňuje realizovat poruchové stavy modelů součástek přímo při zadávání vlastností dané součástky. Stačí pouze vybrat prvek, otevřít okno vlastnosti prvku poklepáním a vybrat kartu *Failure Modes*.

Pokud je vybrán typ *Short*, jsou uzly spojeny. Jedná se tedy o zkrat součástky. Pokud je vybrán typ *Open*, je umístěn velký odpor mezi uzlem prvku a uzlem obvodu. Jedná se tedy o přerušení obvodového prvku. Je-li vybrán typ *Stuck*, jsou uzly spojeny s odporem připojený ke zdroji B. Druhý vývod zdroje B je spojen se zemí. Tento způsob umožňuje zamrznutí daného prvku, který zde nebude uvažován.

Existují dva typy poruch. V této práci budou realizovány pouze tvrdé poruchy.

- měkké poruchy jsou poruchy, kdy jeden nebo více parametrů prvku změníme mimo rozsah jeho normální funkce (například změnou tolerance prvku),
- tvrdé poruchy jsou poruchy *přerušení*, *zkrat* a *zamrznutí* prvku.



Obrázek 3.5 Možnosti nastavení poruchových stavů

Na obrázku 3.5 je vidět možnost nastavení velikostí odporů pro poruchové stavy *přerušení*, *zkrat* a *zamrznutí* součástky. To je možné po otevření okna vlastnosti prvku a na záložce *Failure Modes* kliknutím na parametr prvku s označením «*Short*. Může být zadán i výraz pro zdroj B pro poruchový stav *zamrznutí* součástky. Dále popis a název poruchy a nakonec je umožněna porucha zatížení součástky. Tato porucha se používá pro výpočet entropie při provádění automatického testovacího návrhu, který bude popsán v kapitole 3.4.3.2.

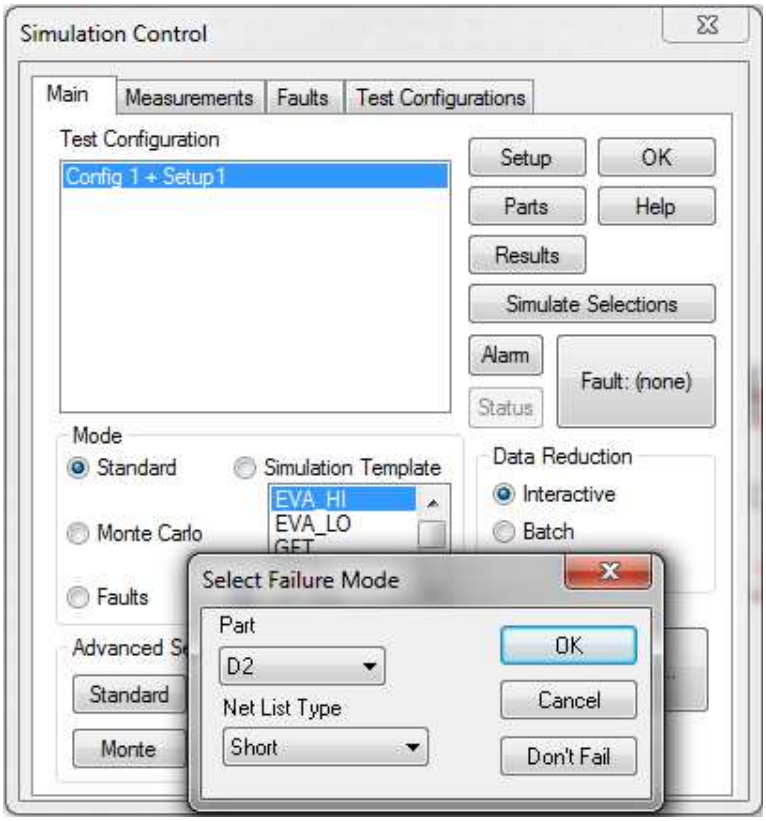
V této kapitole se čerpalo z literatury [14].

3.4.3.1 Realizace poruchových stavů kapacitního dekodéru

Následující příklad realizuje poruchové stavy kapacitního dekodéru v programu ICAP pro prvek D2. Bude zde zkoumán účinek poruchového stavu *přerušení* a *zkrat*. Výsledky poruchových stavů pro všechny prvky, jsou uloženy na přiloženém CD. Jsou označeny anglickými názvy pro přerušení *Open* a pro zkrat *Short*, například *D2_preruseni_open.pdf* a *D2_zkrat_short.pdf*.

Pokud je vytvořen obvod ve schematickém editoru SpiceNet, v nabídce menu „***Actions***“ bude spuštěno tlačítko [***ICAPS/Simulation Control…***]. Otevře se okno s možností ovládání simulace pro poruchové stavy. Stisknutím tlačítka [***Fault: (none)***] na pravé straně, je možné vybrat jakýkoliv prvek obvodu, pro který bude realizován poruchový stav.

Poruchové stavy budou realizovány pro prvek D2. Z nabídky „***Net list type***“ bude vybrána možnost *Short* pro realizaci poruchového stavu *zkrat*. Pak stačí pouze vybrat standardní mód simulace a tlačítkem [***Simulate Selections***] spustit simulaci poruchového stavu *zkrat* na prvku D2. Pro poruchový stav *přerušení*, stačí pouze v nabídce „***Net list type***“ vybrat možnost *Open* a stejným postupem spustit simulaci obvodu. Tímto způsobem lze realizovat poruchové stavy *přerušení* a *zkrat* pro všechny prvky obvodu.



Obrázek 3.6 Nastavení poruchového stavu zkrat na prvku D2

Chování obvodu kapacitního dekodéru při projevu poruchového stavu *přerušení* a *zkra*t bylo popsáno v kapitole 3.3.3. Zhodnocení simulovaných poruch na prvcích obvodu

je znázorněno v tabulce 5. Výsledky poruchových stavů *přerušení* a *zkrat* jsou zobrazeny v příloze II.

Tabulka 5 Simulované poruchy prvků obvodu v ICAP

Prvek	Porucha	Projev poruchy na výstupu	Zhodnocení projevu poruchy
C1	*přerušení*	*pokles napětí na 34µV*	*Bezpečné*
C1	*zkrat*	*pokles napětí na 36µV*	*Bezpečné*
D1	*přerušení*	*vzrůst napětí na -1,23mV*	*Bezpečné*
D1	*zkrat*	*vzrůst napětí na -9,5µV*	*Bezpečné*
C2	*přerušení*	*pulsující napětí od 0V do -23V*	*Bezpečné*
C2	*zkrat*	*špičky napětí do -190mV*	*Bezpečné*
D2	*přerušení*	*vzrůst napětí na -1,15mV*	*Bezpečné*
D2	*zkrat*	*pulsující napětí od 0,8V do -11,5V*	*Bezpečné*

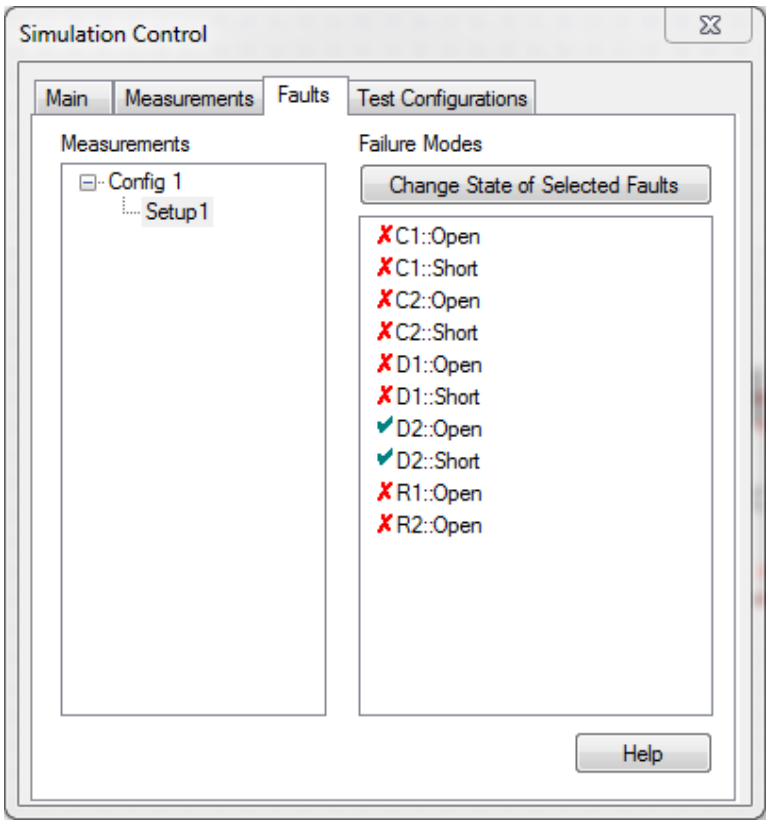
Bezpečný projev poruchy splňuje požadavek na bezpečné chování obvodu (viz kapitola 3.2). Nebezpečný projev poruchy tento požadavek nesplňuje, a proto obvod nemůže být použit jako zařízení s bezpečným projevem poruchy v zabezpečovací technice železniční dopravy. Žádnou poruchou v tabulce 5 nedojde k nebezpečnému chování obvodu kapacitního dekodéru.

3.4.3.2 Automatický testovací návrh prvků obvodu

Automatický testovací návrh je součástí ICAP/4, umožňuje výběr poruchového stavu na jakémkoliv prvku obvodu. Tato součást programu ICAP má defaultně nastaveny poruchové stavy *přerušení* a *zkrat* na všech prvcích obvodu kapacitního dekodéru, kromě obvodových prvků rezistorů R1 a R2. Na těchto prvcích je možné realizovat pouze poruchový stav *přerušení*.

V okně s možností ovládání simulací pro poruchové stavy na záložce *Faults* mohou být vybrány jakékoliv prvky obvodu s poruchovými stavy, pro jaké bude obvod simulován. Poruchové stavy prvků obvodu, které nebudou vybrány pro simulování, mohou být zrušeny označením a stisknutím tlačítka [***Change State of Selected Faults***]. Poté stačí přejít na záložku *Main* (viz obrázek 3.6) a změnit standardní mód na mód *Faults*. Jsou zde dvě možnosti, jakým způsobem budou zobrazeny výsledky simulací. To je možné změnou v *Data Reduction*, v záložce *Main* výběrem buď [***Interactive***], nebo [***Batch***]. První způsob zobrazí

výsledky simulací v okně grafického postprocesoru IntuScope pro každý poruchový stav prvku obvodu. Druhý způsob uloží data do výstupního souboru s koncovkou *CIR*. Po zvolení z těchto dvou možností je možné spustit simulaci tlačítkem [***Simulate Selections***]. Zobrazí se oznámení, že bude provedeno tolik simulací poruchových stavů, kolik jich bylo vybráno.



Obrázek 3.7 Výběr simulovaných poruchových stavů

Tímto způsobem je možné vytvořit simulace pro všechny zvolené poruchové stavy. Výsledky simulací jsou shodné s výsledky vytvořené v kapitole 3.4.3.1.

4 Závěr

Tato práce byla rozšířena i o další zadání na základě konzultace s Doc. Ing. Ivanem Konečným, CSc. a Ing. Petrem Hlouškem, Ph. D. Rozšíření téma diplomové práce spočívalo v problematice kompatibility programů PSPICE a ICAP, a možnostmi vytvoření akumulačních prvků L a C s časově proměnnými parametry v PSPICE a následným převodem do programu ICAP.

V kapitole 2 byla probrána problematika kompatibility PSPICE a ICAP. Jedná se o programy, které patří do rodiny SPICE. Mezi nimi jsou odlišnosti v syntaxi NETLISTU. Proto bylo nutné oba programy porovnat a popsat převod struktury NETLISTU do ICAP pro zachování kompatibility obou. V první fázi proběhla studie rozdílů v syntaxi NETLISTU. Vycházelo se ze známé struktury NETLISTU v PSPICE a čerpalo z manuálů a z domovské stránky výrobce Intusoft. Většina odlišností obou programů byla popsána v literatuře [3], musela být přeložena a následně zkontrolována. Tento popis slouží především uživatelům, kteří mají alespoň s PSPICE nějaké zkušenosti a chtěli by tento program porovnat s programem ICAP.

Problematika možností vytvoření akumulačních prvků L a C s časově proměnnými parametry v PSPICE a následným převodem do programu ICAP byla popsána v kapitole 2.4. Některé funkce ICAP nepodporuje, proto bylo nutné vyhledat jiné řešení zadávání časově proměnných parametrů. Bohužel tento problém se nepodařilo zcela vyřešit. Zmíněná problematika byla v této části úkolu popsána a je zde také nastíněn způsob převodu do ICAP.

Poslední úkol této práce obsahoval řešení rozboru bezpečnosti poruchových stavů na zadaném obvodu kapacitního dekodéru. Zpočátku se tento úkol jevil jednoduše. Postupem času nastaly problémy se správným zapojením do obvodu pro vyzkoušení poruchových stavů z důvodu nedostatečného popisu způsobu realizování poruchových stavů na vytvořených modelech. V programu PSPICE se tento problém podařilo celkem snadno vyřešit, ale v ICAP nastaly problémy při zapojení. Simulace stejně zapojených modelů zobrazovaly jiné výsledky v porovnání s programem PSPICE. Bylo nutné prostudovat řešení modelu CHATTER v ICAP a porovnat jej s programem PSPICE. Nakonec tento problém byl zjištěn a podařilo se jej vyřešit. Výsledné chování poruchových stavů kapacitního dekodéru bylo zhodnoceno

v kapitole 3. Obvod může být považován za bezpečný z hlediska zabezpečovací techniky v železniční dopravě.

Závěrem bych chtěl říci, že snad tato práce pomůže při řešení rozboru bezpečnosti poruch zabezpečovacích zařízení a možnosti porovnávat výsledky simulací v obou programech. Byly splněny všechny body zadání. V případě navazování na tuto práci doporučuji dokončit převod prvků L a C s časově proměnnými parametry do programu ICAP.

Literatura

[1] ROZTOČIL, L.: *Diplomová práce*. ZČU fakulta elektrotechnická, Plzeň, 2011.

[2] *IsSpice4 User´s Guide*, Build 3090. Intusoft, 2007.

[3] Intusoft; *Pspice2IsSpice* [online]; 2012; [2012-05-08];

URL <http://www.intusoft.com/Pspice2IsSpice.htm>

[4] PINKER, J., KOUCKÝ, V.: *Analogové elektronické systémy*, 1. a 2. část. ZČU, Plzeň, 2006

[5] KOLKA, Z.: *Analýza elektronických obvodů programem OrCad PSpice*. Elektronické učební texty, ÚREL FEKT VUT Brno, 2004.

[6] BIOLEK, D.: *Modelování a simulace v mikroelektronice*. Elektronické učební texty, ÚMEL FEKT VUT Brno, 2005.

[7] Intusoft; *Pspice_to_IsSpice_Converter* [online]; 2012; [2012-05-08];

URL <http://www.intusoft.com/products/Pspice_to_IsSpice_Converter.htm>

[8] BIOLEK, D., BIOLKOVÁ, V.: *Modelování akumulačních prvků s proměnnými parametry v PSpice*. Článek v odborném periodiku, Slaboproudý obzor, 2010

[9] BIOLEK, D., KOLKA, Z., BIOLKOVÁ, V.: *Modeling time-varying storage components in PSpice*. Dept. of EE, FMT, University of Defence, Brno 2007

[10] TUHÁČEK, I.: *Diplomová práce*. ZČU fakulta elektrotechnická, Plzeň, 1998.

[11] TŘÍSKA, R.: *Diplomová práce*. ZČU fakulta elektrotechnická, Plzeň, 1999.

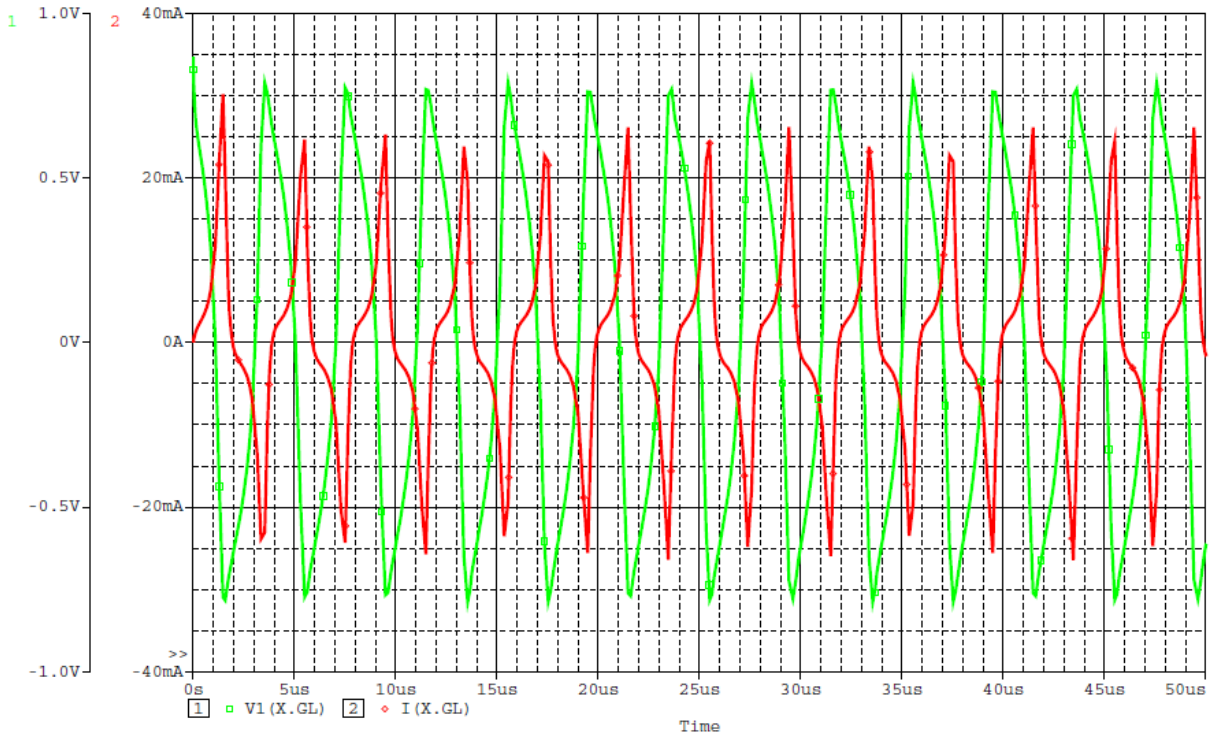
[12] CHUDÁČEK, V. a kol.: *Železniční zabezpečovací technika*. 2. přeprac. a doplň. vyd. Praha: VÚŽ, 2005.

[13] HLOUŠEK, P.: *Zabezpečovací technika v žel. dopravě 1*. Přednášky a cvičení, ZČU FEL, Plzeň, 2011.

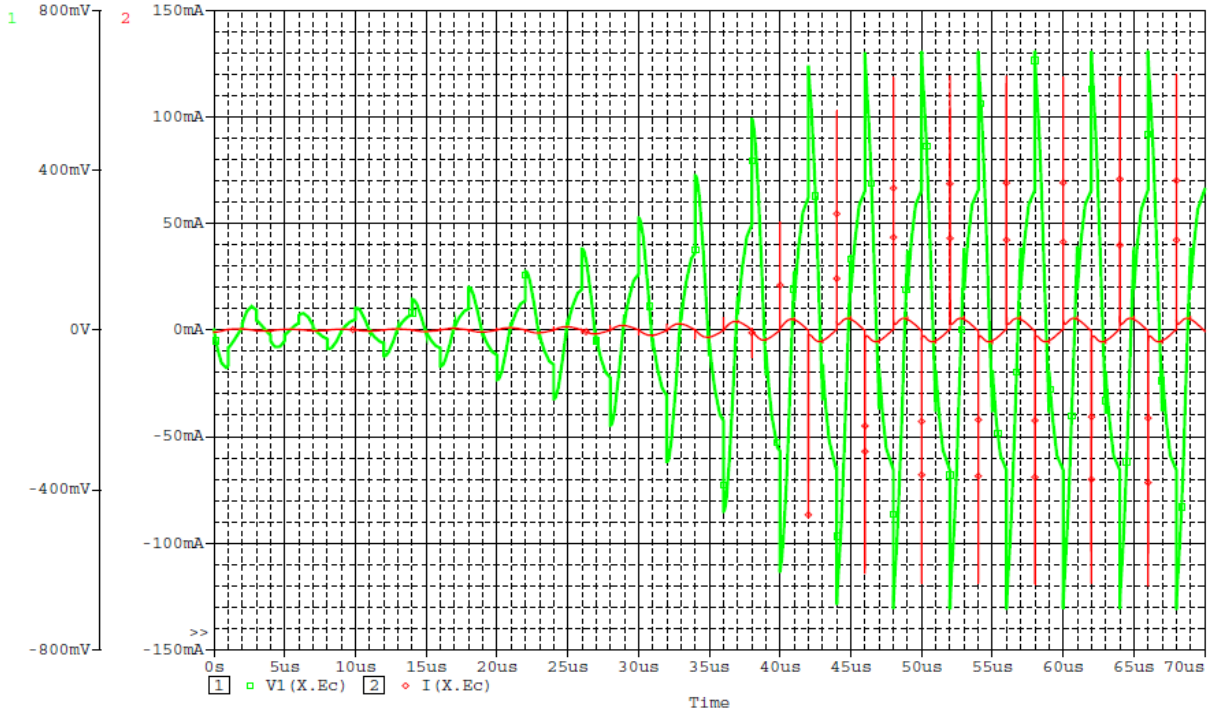
[14] BÜRMEN, A.: *AN INTRODUCTION TO ICAP/4*, Fakulteta za elektrotehniko. Rosenheim, 2000.

Přílohy

Příloha I: Výsledky simulací akumulačních prvků L a C

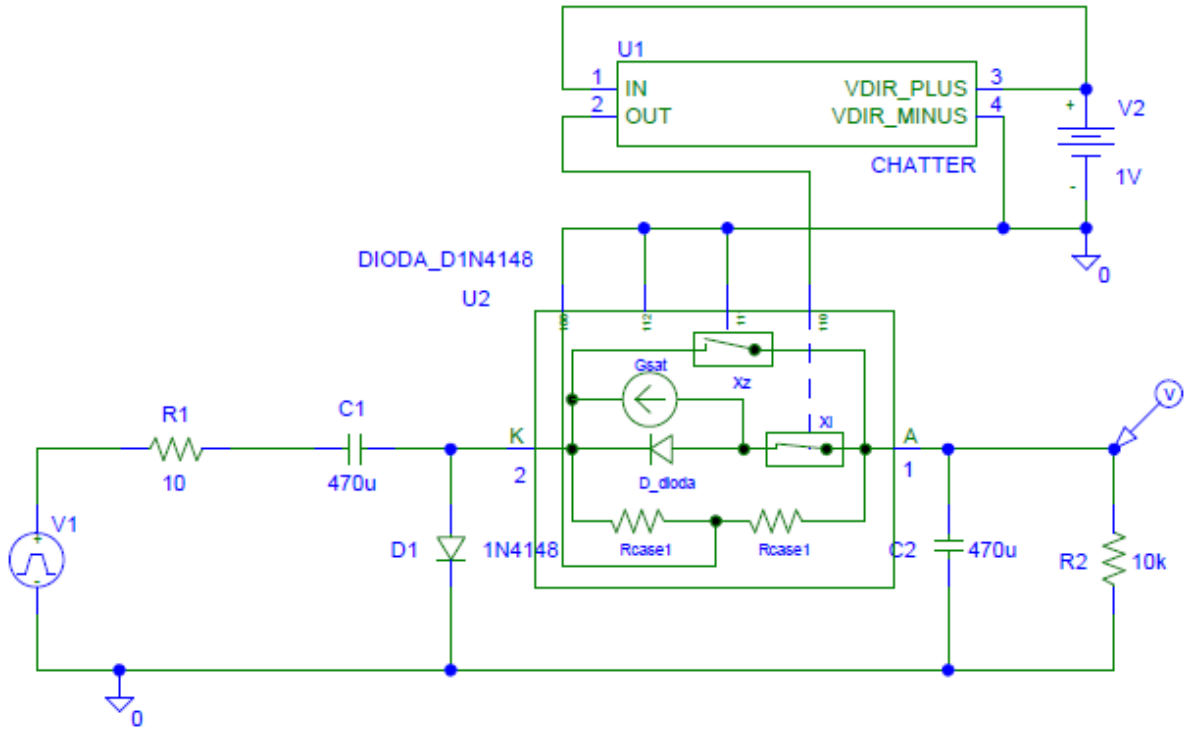


Obrázek 1 Periodický ustálený stav pro časově proměnnou L v PSPICE

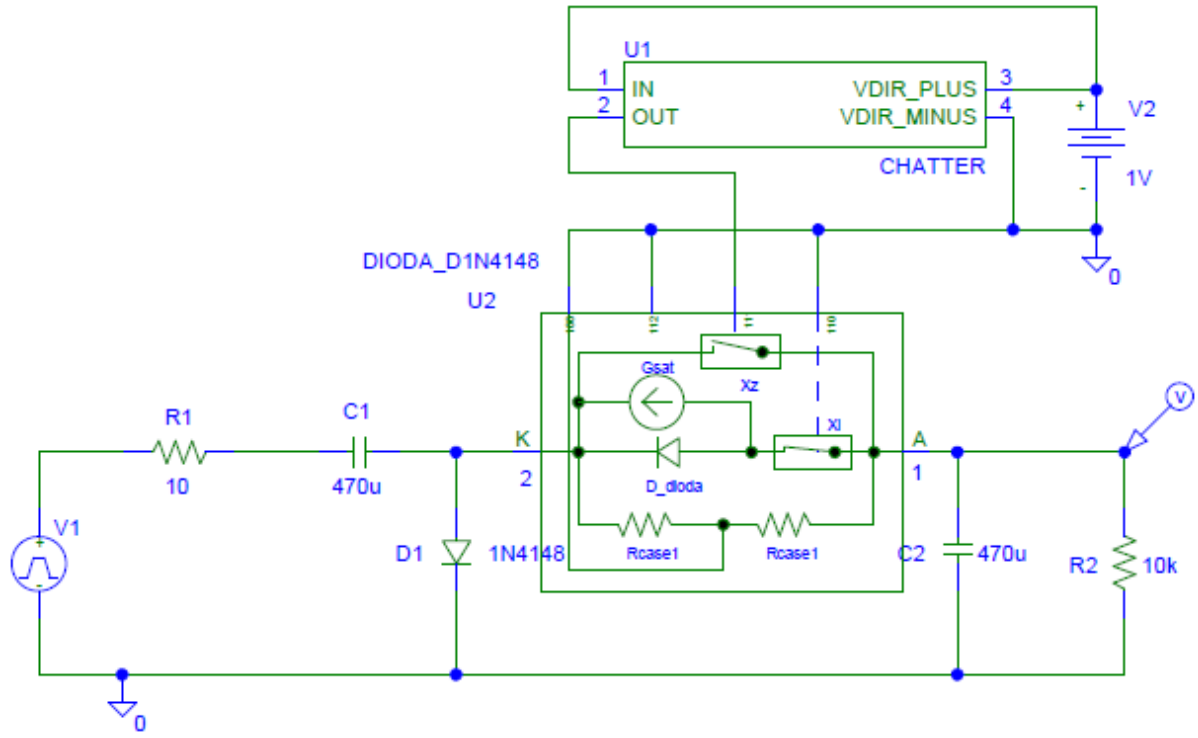


Obrázek 2 Periodický ustálený stav pro časově proměnnou C v PSPICE

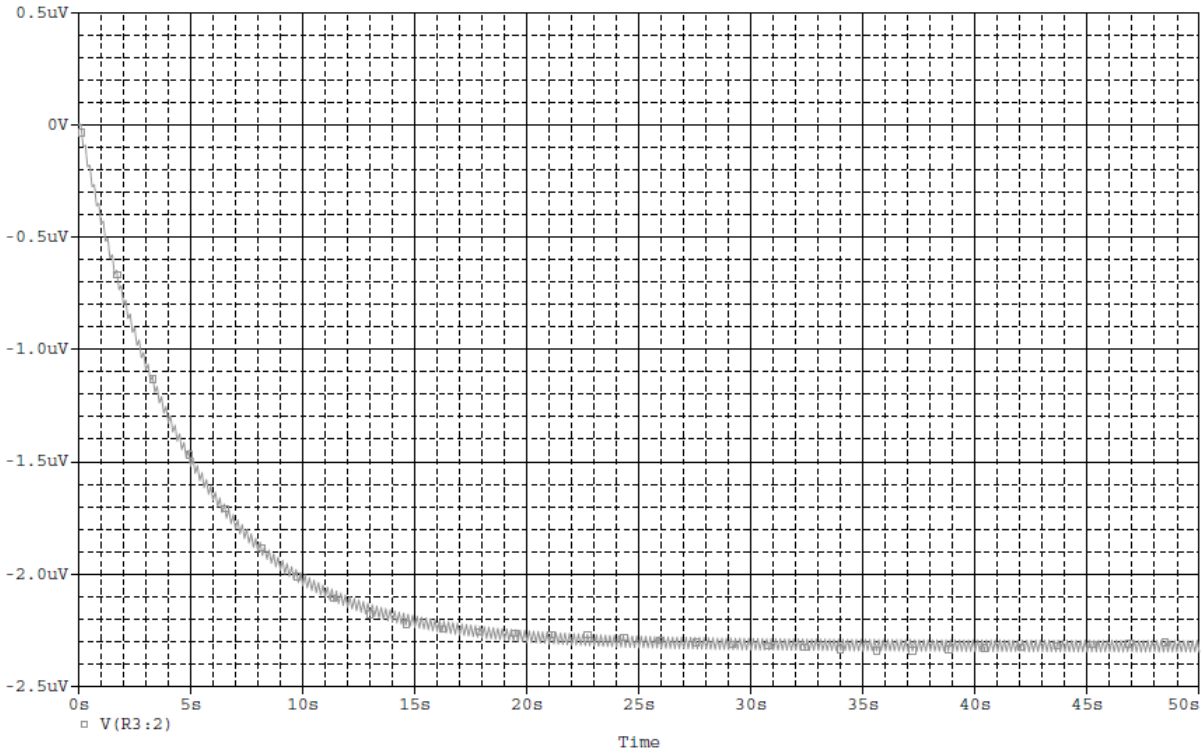
Příloha II: Zapojení a výsledky simulací poruchových stavů



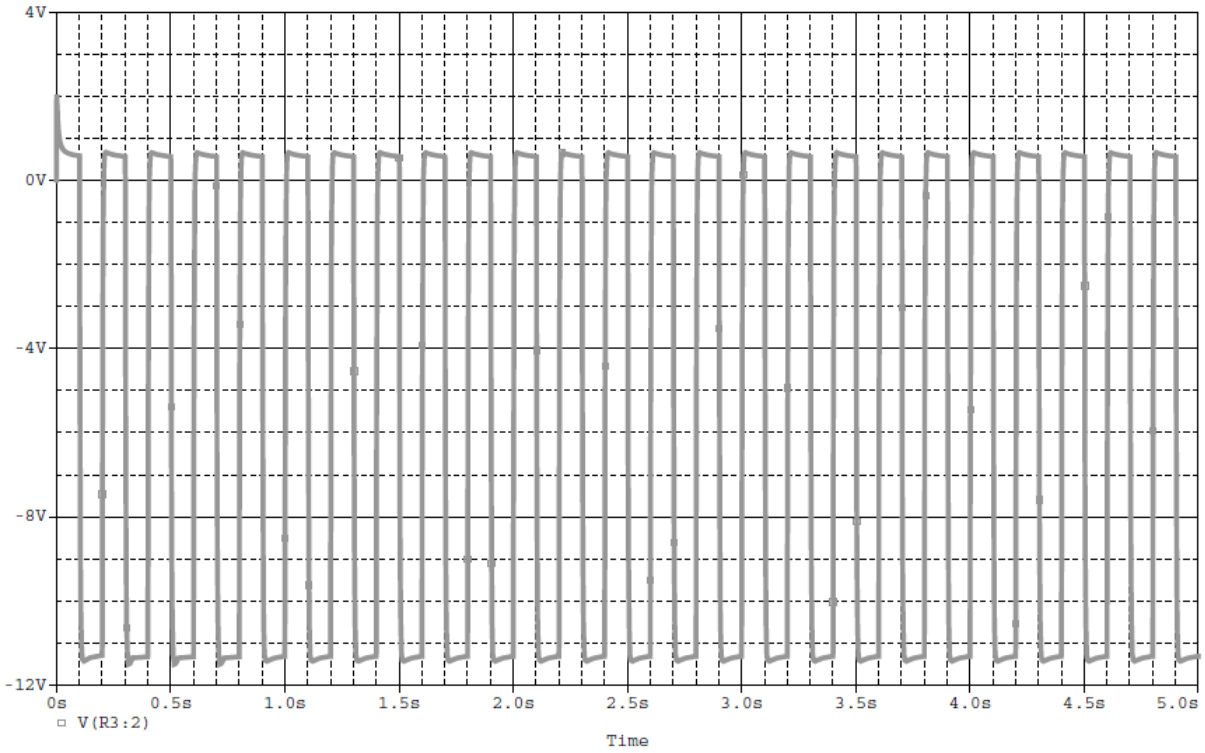
Obrázek 3 Zapojení kapacitního dekodéru v PSPICE, *přerušení* na D2



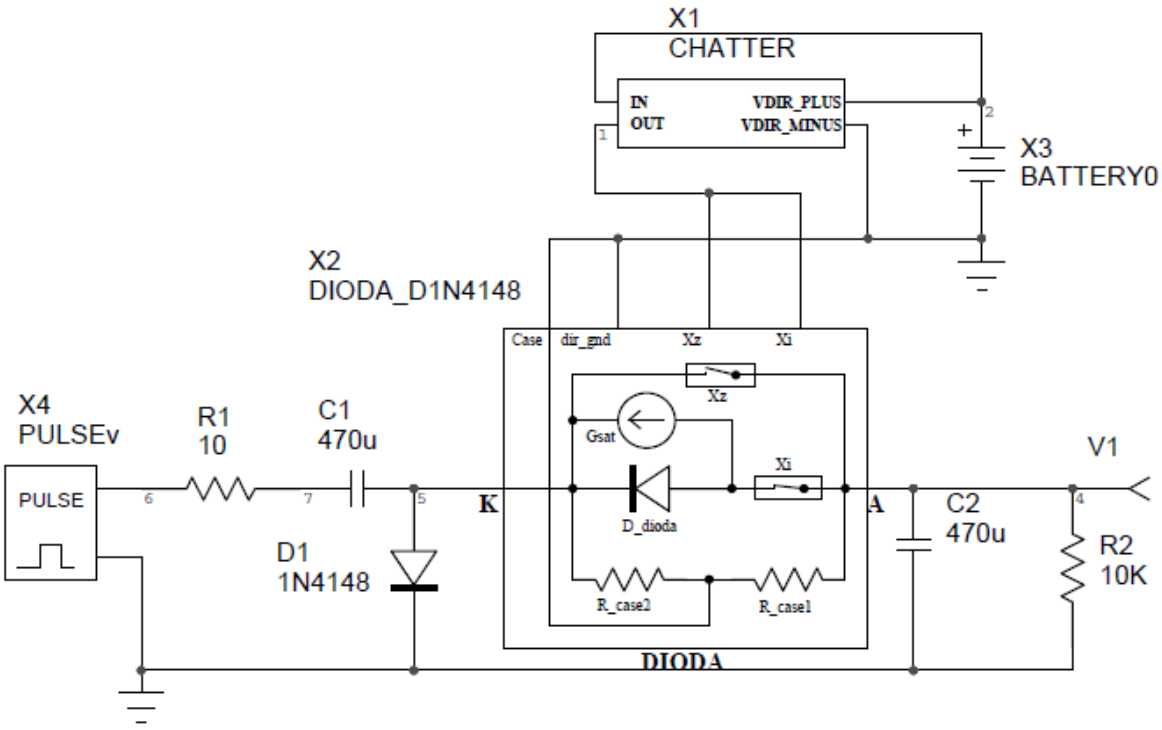
Obrázek 4 Zapojení kapacitního dekodéru v PSPICE, *zkrat* na D2



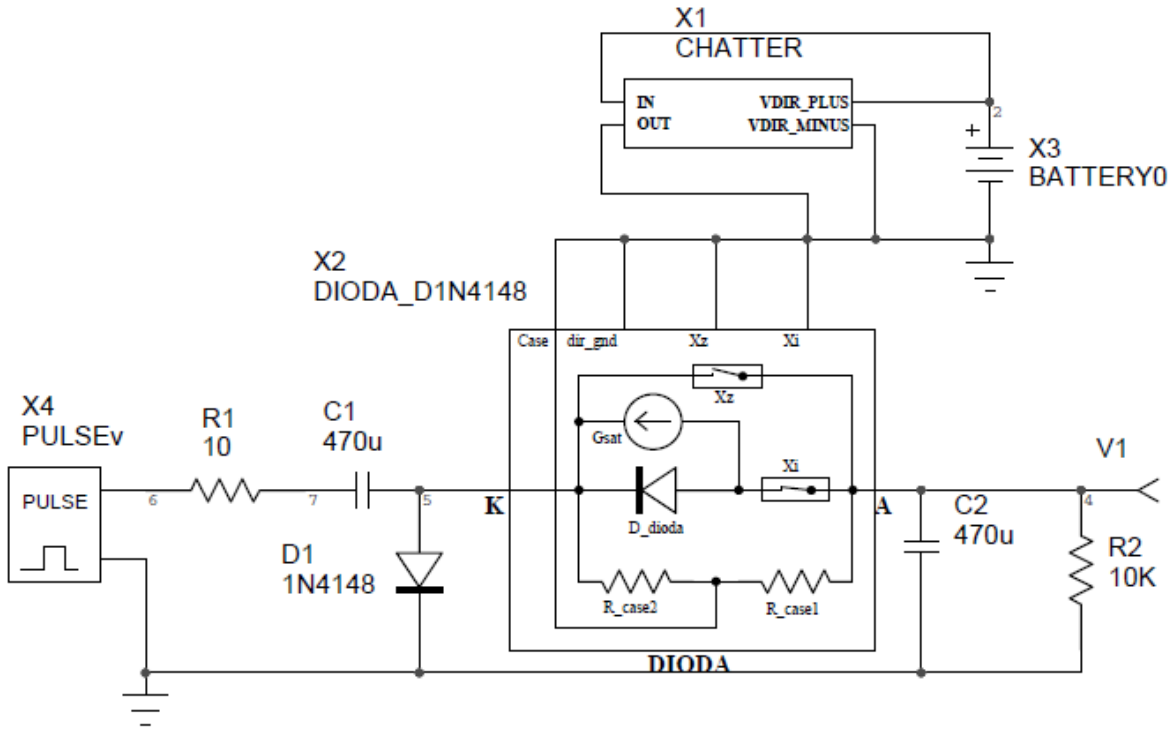
Obrázek 5 Poruchový stav *přerušení* na D2 v PSPICE



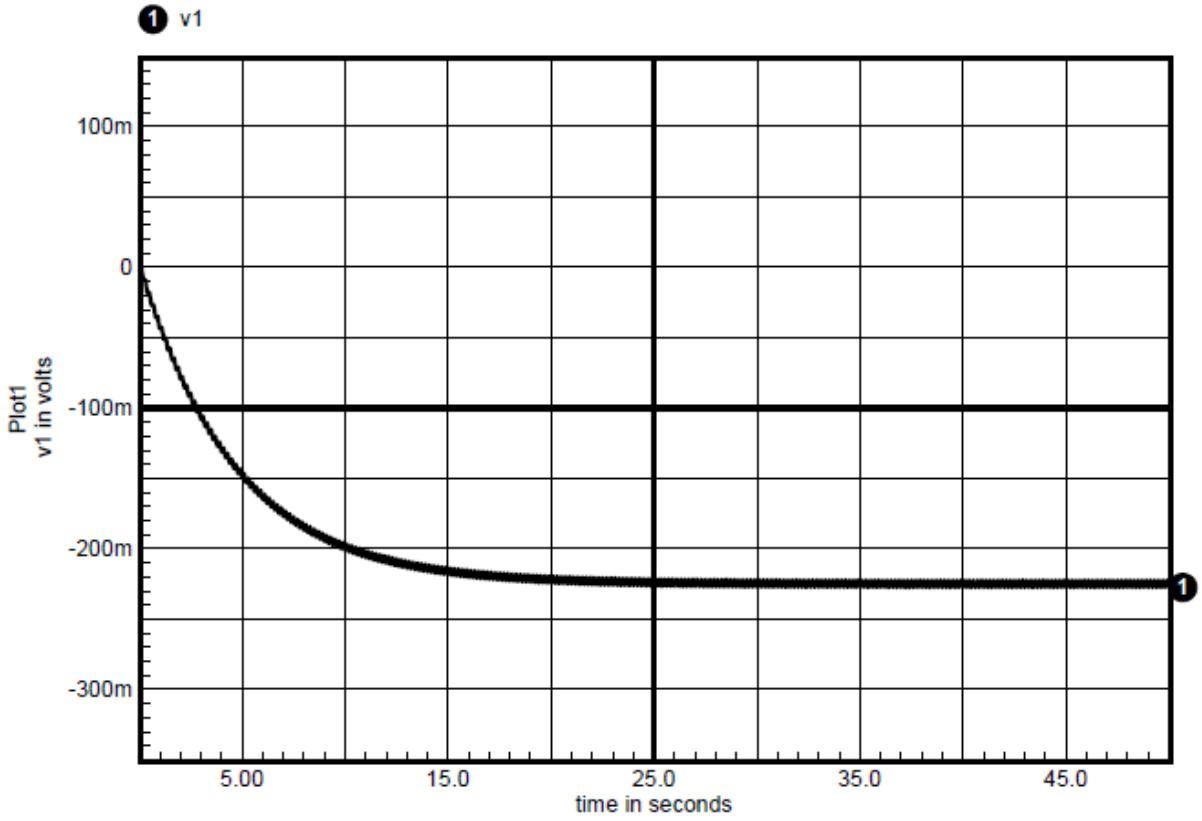
Obrázek 6 Poruchový stav *zkrat* na D2 v PSPICE



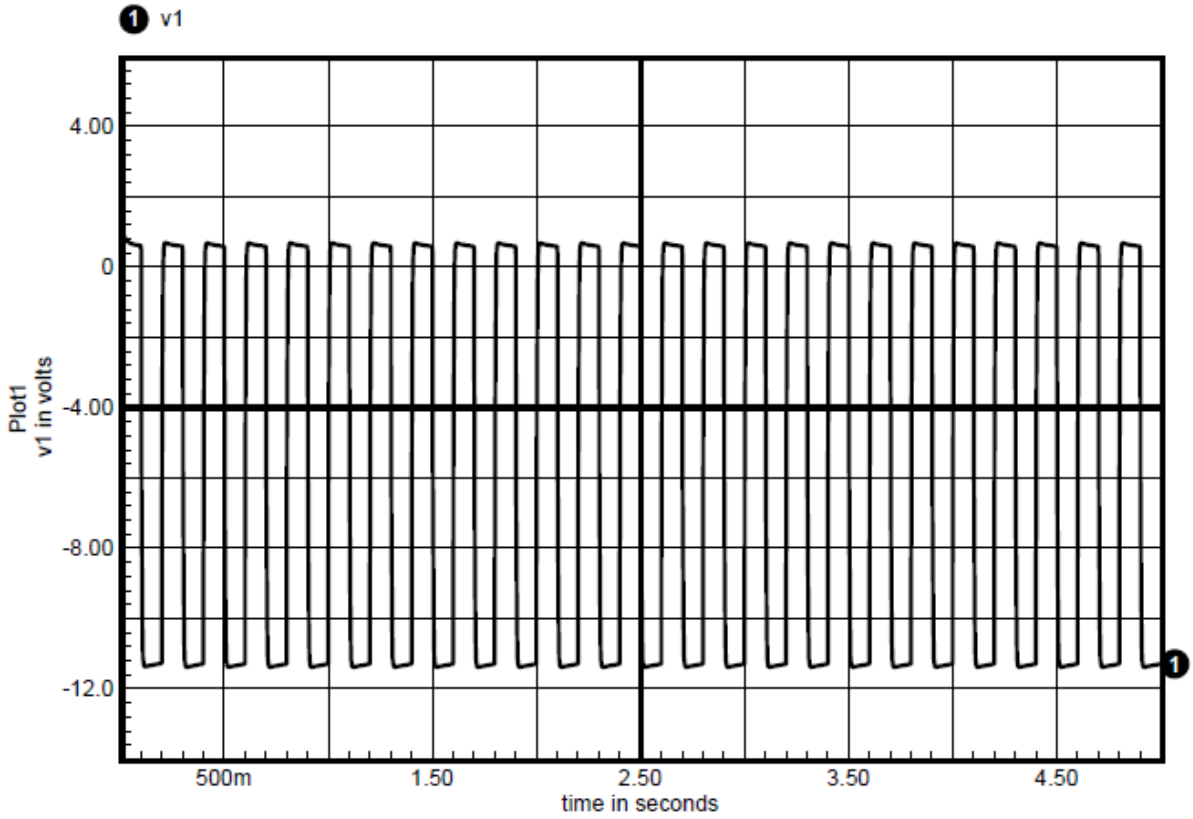
Obrázek 7 Zapojení kapacitního dekodéru v ICAP, *přerušení* na D2



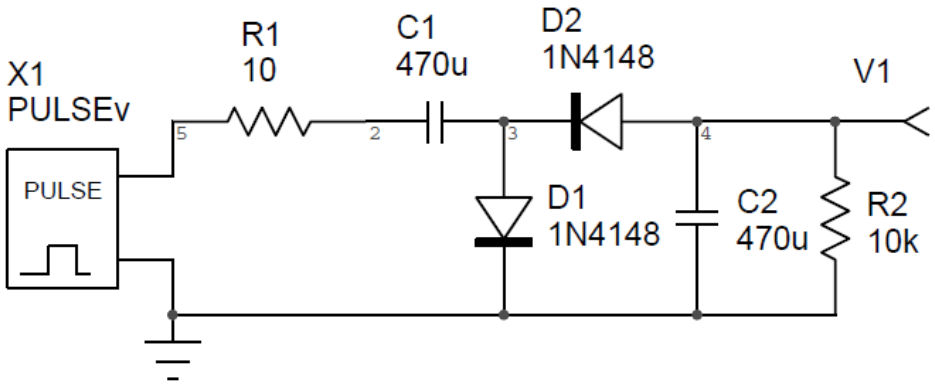
Obrázek 8 Zapojení kapacitního dekodéru v ICAP, *zkrat* na D2



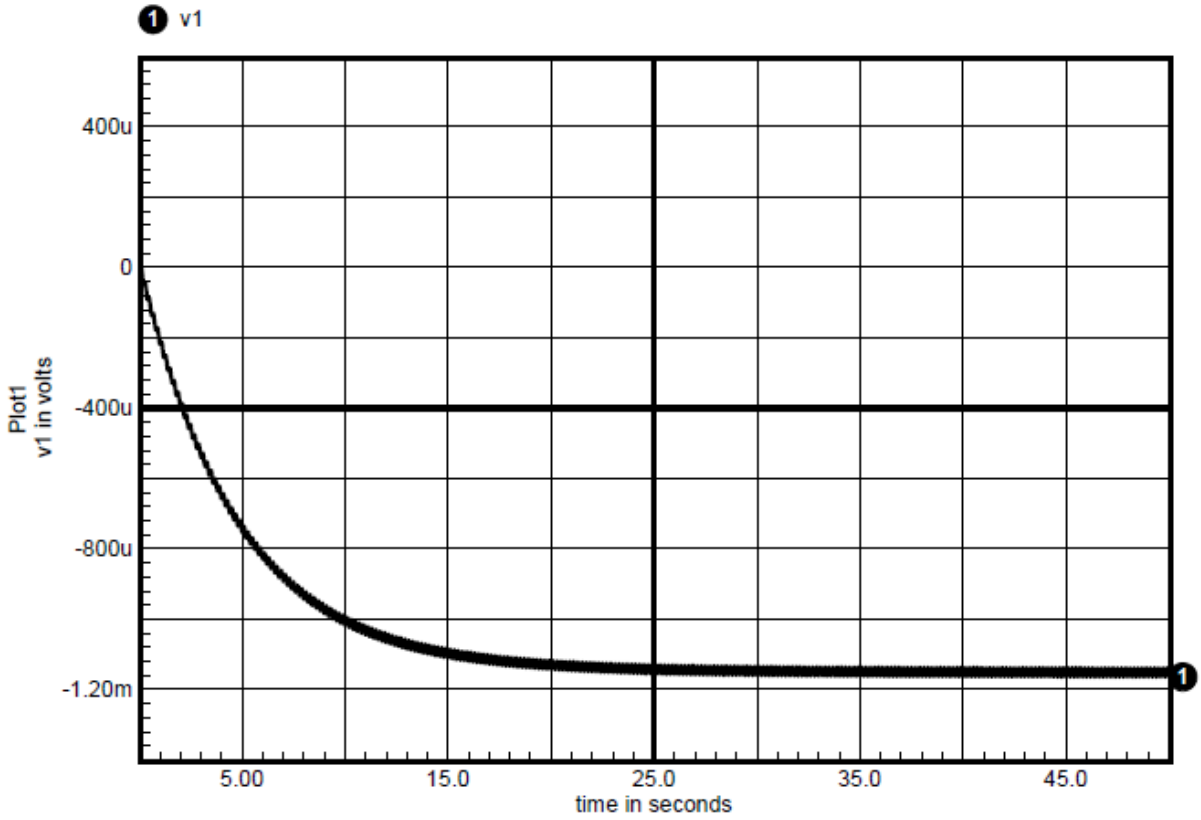
Obrázek 9 Poruchový stav *přerušení* na D2 v ICAP



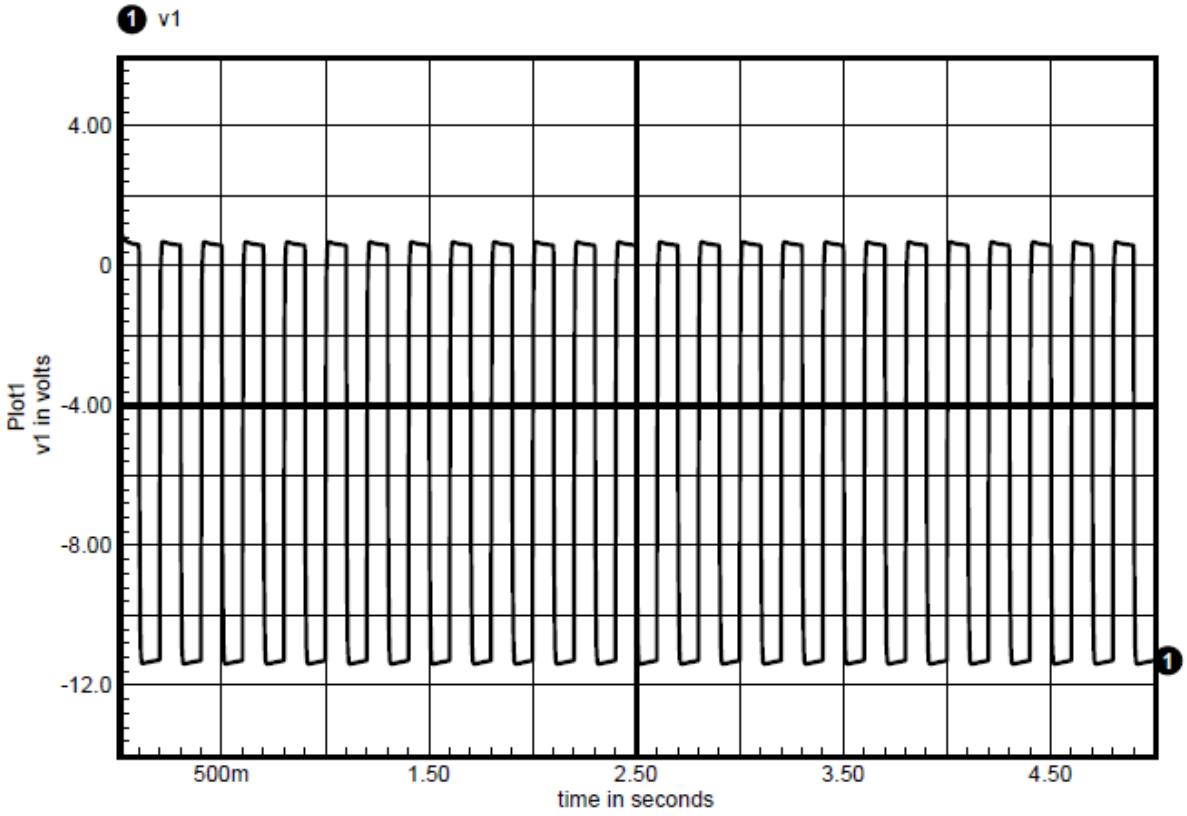
Obrázek 10 Poruchový stav *zkrat* na D2 v ICAP



Obrázek 11 Klasické zapojení kapacitního dekodéru v ICAP



Obrázek 12 Poruchový stav *přerušení* na D2 pro klasické zapojení v ICAP



Obrázek 13 Poruchový stav *zkrat* na D2 pro klasické zapojení v ICAP

Příloha III: Obsah přiloženého CD

Soubor	Obsah souboru
Grafický editor\ICAP*prvek_porucha*.dwg	Vytvořené schéma kapacitního dekodéru s modely poruchových stavů
Grafický editor\ICAP\klasik.dwg	Vytvořené schéma kapacitního dekodéru
Grafický editor\PSPICE*prvek_porucha*.dwg	Vytvořené schéma kapacitního dekodéru s modely poruchových stavů
Grafický editor\PSPICE\normal.dwg	Vytvořené schéma kapacitního dekodéru
Grafy\Kapacitní dekodér\ICAP\ *prvek_porucha*.pdf	Grafy výstupu kapacitního dekodéru při poruchách
Grafy\Kapacitní dekodér\ICAP*prvek_porucha_nazev*.pdf	Grafy výstupu kapacitního dekodéru při poruchách
Grafy\Kapacitní dekodér\PSPICE\ *prvek_porucha*.pdf	Grafy výstupu kapacitního dekodéru při poruchách
Grafy\ RLC\ *prvek*_promenna.pdf	Grafy výstupu časově proměnných prvků obvodu
ISED5\ *prvek*_promenna.cir	Vytvořený textový soubor časově proměnných prvků obvodu
Knihovny\ICAP\Originál\DIODA_OF.lib	Knihovna s modely diody
Knihovny\ICAP\Originál\Diplomka.sym	Značky všech vytvořených modelů poruchových stavů
Knihovny\ICAP\Originál\DIPLOMKA_PF.lib	Knihovna s obvodem CHATTER
Knihovny\ICAP\Originál\KAPACITOR_PF.lib	Knihovna s modely kondenzátorů
Knihovny\ICAP\Upravené\diody.lib	Upravená knihovna s modely diody
Knihovny\ICAP\Upravené\diplomka.lib	Upravená knihovna s obvodem CHATTER
Knihovny\ICAP\Upravené\kapacitor.lib	Upravená knihovna s modely kondenzátorů
Knihovny\PSPICE\DIODY.lib	Knihovna s modely diody
Knihovny\PSPICE\DIPLOM.lib	Knihovna s obvodem CHATTER
Knihovny\PSPICE\KAPACITY.lib	Knihovna s modely kondenzátorů

* *prvek* nahrazuje název příslušného prvku
** *porucha* nahrazuje název příslušné poruchy
*** *název* nahrazuje anglický název příslušné poruchy