

ZPŮSOB ŘEŠENÍ KONVERZAČNÍHO REŽIMU V SYSTÉMU SPDL/SCL

Ing. Aleš Kastner, CSc

1. Úvod

Systém SPDL/SCL, pojmenovaný podle dvou speciálních jazyků, které obsahuje, je automatizovaný systém určený k logické fázi návrhu elektronických obvodů vysokého stupně integrace. Při vývoji tohoto systému se počítalo s vysokou úrovní interakce člověk - stroj a zároveň s malou zkušeností návrháře s tímto způsobem práce.

Účelem konverzačního režimu systému SPDL/SCL je umožnit postupné drobné i větší modifikace "přeloženého programu" a jeho datových struktur, přičemž po každé změně lze spustit "výpočet" (tedy simulaci činnosti obvodu), sledovat na displeji jeho výsledky a kdykoli jej přerušit. V konverzačním režimu si může návrhář také vyžádat text vysvětlující tvar a funkci některého příkazu jazyka. Pro realizaci byl zvolen řádkový režim, který lze provozovat na všech dostupných komunikačních terminálech. Jediným dalším požadavkem kladeným na terminál je jeho schopnost vyžádat si z vlastní iniciativy pozornost počítačového systému (bývá zpravidla řešena tlačítkem označovaným ATTENTION).

Konverzační režim je zpravidla zahájen překladem a spuštěním plánu simulace sestaveného návrhářem obvodu. Systém je navržen tak, aby návrhář byl průběhu simulace přítomen, sledoval ji na svém terminálu a v případech, kdy je přerušena (ať z iniciativy systému nebo návrháře), mohl plán simulace doplnit nebo změnit konverzačním způsobem. Pak může být simulační běh opakován nebo může pokračovat z místa přerušení. Úseky simulačního běhu a konverzace návrháře se systémem se mohou střídát podle

potřeby i vícekrát.

Průběh celého experimentu nejlépe vyjadřuje sekvenční automat (obr. 1). Jsou v něm již rozlišeny určité druhy konstrukcí jazyka (deklarace, příkazy a povely), které byly zavedeny právě k pokrytí potřeb konverzačního režimu.

2. Požadavky konverzačního režimu na jazyk a překladač

Činnost překladače v průběhu experimentu se dělí do dvou etap, které jsou odděleny chodem simulátoru a interpretu (na obr. 1 "sim."). První etapu tvoří dávkový překlad (horní dva uzlyna obr. 1), druhou překlad konverzační (dolní dva uzly). Přejchod z první etapy do druhé může být automatický (např. časováním doby simulace) nebo vynucený zásahem návrháře. Přejchod z druhé etapy do první nebo přes chod simulátoru znovu do druhé etapy může provést pouze návrhář. Jazyk tedy musí obsahovat skupinu zvláštních příkazů - povelů - pro různé varianty přechodu. Tyto povely nemají smysl v dávkovém překladu a jsou zde zakázány.

2.1. Dávkový překlad

Překladač čte a zpracovává připravený text. V jazyku SCL obsahuje text v zásadě tyto celky:

- úvodní příkaz definující objekt simulace (model obvodu),
- deklarace pracovních struktur a objektů,
- výkonné příkazy měnící stav a strukturu modelu,
- příkaz START, kterým je text ukončen a spuštěna simulace.

Většina příkazů pracuje s objekty modelu, proto musí být povinně zadán úvodní příkaz. Z druhé strany ohraničuje text plánu simulace příkaz START. Zde je překlad ukončen, vyhodnocena jeho úspěšnost a vypsány případné chyby.

Pro dávkový překlad není důležité, jakým způsobem překladač pracuje, kolik má průchodů apod. Vytvořené datové struktury však musejí být sestaveny tak, aby se v konverzačním režimu daly měnit, rušit a doplňovat. V případě jazyka SCL je většina těchto struktur složena z pseudoinstrukcí, které jsou v průběhu simulace interpretovány (tzv. řídicí struktury).

2.2. Konverzační překlad

Práce návrháře v konverzačním režimu vykazuje značnou podobnost s prací programátora při interakčním ladění programu (viz např. ladící prostředek HILDA v operačním systému DOS-4/JS).

V této etapě návrhář "ledí" ve spolupráci se simulačním systémem, se kterým se dorozumívá jazykem řízení simulace (v našem případě jazykem SCL). Překladač tohoto jazyka musí být "on-line" spolu s určitou částí systému, která musí návrháři poskytovat informace o aktuálním stavu modelu a umožňovat jeho změny. Tato potřeba se v jazyku odrazí vznikem skupiny příkazů dotazovacího charakteru s okamžitou reakcí. Překladač musí tedy být schopen práce v rámci dílčích syntaktických konstrukcí. Jsou to:

- povely,
- dotazovací (výpisové) příkazy,
- příkazy pro okamžitou změnu hodnot objektů modelu,
- definiční (např. řídicí) struktury.

Jazyk musí být koncipován tak, aby syntaktická stavba jeho struktur (jak deklarací, tak zejména struktur příkazů) jednoznačně odpovídala vygenerovaným datovým resp. instrukčním strukturám. Pak může překladač rozhodnout, že jistá vložená část textu tvoří celek, který lze odděleně zpracovat, vyhodnotit jeho správnost a generovat reprezentaci ve výstupním jazyku.

Důležitým požadavkem konverzačního režimu je okamžitá indikace chyby a možnost její opravy. Při dodržení předchozí pod-

mínky strukturovanosti se problém indikace chyby redukuje na okamžitý výpis chybového hlášení. Dále musí být zaručeno, aby si uživatel chyby všiml a opravil ji. V překladači SCL je použit jednoduchý mechanismus opravy. Po hlášení chyby musí návrhář nejprve syntakticky ukončit rosepsanou strukturu (jde-li o chybu syntaxe elementů jazyka) nebo příkaz. Většina chyb se hlásí většinou po ukončení příkazu. Struktura se ukončí potřebným počtem příkazových závorek END.

K signalizaci opravy chyby je určen povel REDSPINE (RDP). Má smysl pouze v konverzačním režimu a vykonává několik funkcí. Pokud nastala v průběhu konverzačního překladače závažná chyba, nemůže být vydán povel START nebo CONTINUE k pokračování simulace, dokud nebyl vydán povel REDSPINE. Tím je návrhář nucen opravit chybu novým zadáním příkazu nebo struktury, nebo potvrdit, že hlášení chyby vzal na vědomí. Povel zároveň nuluje indikaci chyby a povoluje přechod od příkazu k deklaraci. (tato funkce je specifická vzhledem k řešení překladače SCL).

2.3. POZNÁMKY O REALIZACI PŘEKLADAČE SCL

Konec konverzačního režimu určují povel START, CONTINUE, GO a STOP. První dva spouštějí simulační běh (START znovu od začátku, CONTINUE pokračuje). Povel GO ukončuje práci návrháře s jedním modelem a zahajuje práci s jiným, přechází tedy do etapy dávkového překladače. Povel STOP ukončuje činnost simulačního systému.

Překladač SCL je vytvořen systémem CGS pro tvorbu kompilátorů. Podmínkou jeho použití je, aby jazyk měl gramatiku typu LL(1). Překladač je jednopřechodový. V konverzačním režimu byly jisté problémy vyplývající z činnosti analyzátoru syntaxe, který pracuje vždy o jeden řádek zpět za řádkem právě zadáním. Protože čtený text je získáván komunikačním modulem (viz dále), dodává tento modul v konverzačním režimu překladači vždy o jeden řádek navíc. Tento řádek je označen tak, aby jej překladač igno-

roval.

Celý překladač s výjimkou konverzačních a některých pomocných procedur je napsán v jazyku PASCAL VUT. Konverzační a pomocné procedury jsou soustředěny do jednoho komunikačního modulu, který je napsán v Assembleru JSEP. Touto koncepcí je usnadněn přenos systému SFDL/SCL na jiné typy počítačů (ADT, PC, SMEP), který v současnosti probíhá.

S0429SL SCL10
MH 3212

*** SCL SYSTEM ***

N	CAS	NS1	S2	SB	S	MD	NR	D	REG	R	NIT	PKO
0	0	0	0	0	0	0	1	00	11	22	X	1
1	1	0	0	0	0	0	1	00	11*ZZZZZZZZ		1	1
2	100	0	0	1	0	0	1	00	11*ZZZZZZZZ		1	1
3	140	0	0	1	0	0	1	00	00*ZZZZZZZZ		1	1
4	200	0	1	1	1	0	1	B3	00*ZZZZZZZZ		1	1
5	230	0	1	1	1	0	1	B3	B3*ZZZZZZZZ		K	1
6	231	0	1	1	1	0	1	B3	B3*ZZZZZZZZ		0	1
7	241	0	1	1	1	0	1	B3	B3	B3	0	1
8	300	0	1	0	1	0	1	B3	B3	B3	0	1
9	400	1	1	0	0	0	1	B3	B3	B3	0	1
10	430	1	1	0	0	0	1	B3	B3	B3	R	1
11	431	1	1	0	0	0	1	B3	B3*ZZZZZZZZ		1	1
12	500	1	1	0	0	0	0	B3	B3*ZZZZZZZZ		1	1
13	555	1	1	0	0	0	0	B3	00*ZZZZZZZZ		1	1
14	600	1	1	0	0	1	1	52	00*ZZZZZZZZ		1	1
15	601	1	1	0	0	1	1	52	00	00	1	1
16	700	1	1	1	0	1	1	52	00	00	1	1
17	800	0	1	1	1	1	1	52	00	00	1	1
18	830	0	1	1	1	1	1	52	00	00	K	1
19	831	0	1	1	1	1	1	52	00	00	0	1
20	840	0	1	1	1	1	1	52	52*0R0R00R0		0	1
21	841	0	1	1	1	1	1	52	52	52	0	1
22	900	0	1	0	1	1	1	4A	52	52	0	1
23	930	0	1	0	1	1	1	4A	4A*010KR010		0	1
24	931	0	1	0	1	1	1	4A	4A	4A	0	1
25	1000	0	1	0	1	1	0	4A	4A	4A	0	1

Tab. 1 Příklad tabulky výsledků simulace

3. VLASTNOSTI PROCEDUR KONVERZAČNÍHO REŽIMU VE STYKU S NÁVRHÁŘI

V konverzačním režimu je pro návrháře výhodné, aby mohl vydávat pomocné povely organizačního charakteru. Tvar a funkce těchto povelů je do jisté míry závislá na možnostech operačního systému a na způsobu řešení simulačního systému. Je proto účelné nezahrnovat tyto jednoduché povely do simulačního jazyka, ale zpracovávat je samostatnou částí simulačního systému. Tato část se skládá z několika oddělených procedur, které tvoří mezičlánek mezi zdroji povelů, textů a zpráv přijímaných a produkováných složkami simulačního systému a návrhářem u terminálu. Potřebné povely mohou být rozděleny do těchto skupin:

- stanovení příštího zdroje textu, příkazů a povelů.
- směrování zpráv a výsledků produkováných sym. systémem,
- příjem požadavku návrháře na přerušeni běžící simulace,
- žádost o vyavětlující výpis (HELP).

Povely by měly být syntakticky co nejjednodušší a snadno rozpoznatelné. Protože vstupní text je čten po řádcích, měly by být řádkově povely orientované; řádek s povelem není předáván překladači SCL. Výhodné schéma povelů lze převzít z některých editorů (LUISA): povel s operandem má výkonný charakter, povel bez operandu poskytne informaci o aktuálním stavu.

3.1. URČENÍ ZDROJE VSTUPNÍCH INFORMACÍ

Vstupní text může přicházet ze systémového zařízení vstupu (SYSIPT), z terminálu návrháře nebo z knihy uložené na zdrojové knihovně. Volbu a přepnutí vstupu zajistí procedura vstupního souboru na povel XINPUT s příslušným parametrem. Po ukončení připraveného textu na vstupním zařízení nebo v knihovně je vstup automaticky přepnut na terminál, o čemž je vydána zpráva. Je důležité, aby se nastavený zdroj týkal všech procedur simul. systému, které mohou požadovat vstupní text (např. data modelu paměti).

3.2. SMĚROVÁNÍ ZPRÁV O PRŮBĚHU SIMULACE

K těmto zprávám patří zejména chybová a varovná hlášení vypisovaná jak při překladu, tak při vlastní simulaci. Výstupním zařízením je standardně terminál, alternativním zařízením tiskový výstup (SYSIST). Je možné a účelné zapojit oba výstupy současně. Při dávkovém zpracování experimentu bez terminálu se volí pouze tiskový výstup. Ke směrování slouží povel %OUTPUT.

3.3. SMĚROVÁNÍ VÝSLEDKŮ

Výsledky simulace jsou soustředěny do přehledné tabulky, jejíž formát si návrhář volí deklarací TABLE jazyka SCL. (příklad viz tab. 1). Standardně vystupuje tato tabulka do souboru výsledků pro možnost jejich pozdějšího zpracování. Do tiskového výstupu se opisuje po ukončení experimentu. Povel %TABLE lze duplicitně směřovat řádky tabulky, tak, jak během simulace vznikají, na terminál, takže jsou okamžitě viditelné a návrhář na ně může okamžitě reagovat.

3.4. PŘERUŠENÍ SIMULACE

Jestliže návrhář sledoval průběh výpočtu (tedy simulace) a její výsledky na terminálu, může přirozeně nastat situace, kdy další pokračování běhu je zbytečné, nebo kdy je nutno ručně zasáhnout do stavu objektů modelu a pokračovat. Záseh spočívá v použití tlačítka ATTENTION (nebo obdobného) na terminálu. Další postup závisí na operačním systému počítače. (V systému DOS-4/JS je nutno napsat dohodnutý asynchronní příkaz s kódem SCL). Výsledkem jeho činnosti musí být aktivace čekající rutiny simulačního systému, která naplánuje zastavení na vhodný okamžik (nebo ihned) a opět se uvede do stavu čekání. Simulaci lze regulérně zastavit jen ve specifickém bodě simulačního cyklu,

anebo jej nouzově přerušit, pokud vinou chybného modelu dojde k nekonečné době simulačního kroku.

Systém SPDL/SCL upozorní po zásahu uživatele, že je nutno vyčkat do konce kroku simulace. Nouzové přerušení je realizováno trojnásobně opakovaným zásahem. Vstup textu (příkazů, povelů) se přepíná automaticky na konverzační zařízení - terminál.

3.5. ČASOVÉ OMEZENÍ KONVERZAČNÍHO REŽIMU

Simulační, ladící nebo obdobný interakční systém bývá rozsáhlý a obsazuje nezanedbatelné množství jednotek operační paměti, úseků disku i jiných prostředků. Proto je vhodné chránit systém před nekázní uživatele časovým omezením komunikační pauzy. Jestliže návrhář příliš dlouho "přemýšlí", je vypsáno varovné hlášení. Nereaguje-li návrhář jakoukoliv (i prázdnou) odpovědí na tři varovná hlášení, je simulační systém ukončen.

Problémem bývá stanovení vhodné časové konstanty. Je možné, aby se základní doba před každým dalším varovným hlášením zvýšila na vhodný násobek. Základní (první) prodleva musí být podle našich zkušeností alespoň 40 sekund, aby hlášení nepřerušilo zápis delšího textového řádku před jeho odesláním.

V simulačním systému SPDL/SCL je využita možnost systému DCC-4 JS zrušit požadavek čtení (TYPE) a očekávat současně více událostí (makro WAIT#).

4. POMOCNÉ FUNKCE KONVERZAČNÍHO REŽIMU

Nejdůležitější pomocnou funkcí je výpis vysvětlující funkcí a tvar zadaného příkazu jazyka SCL nebo konverzační procedury. Výpis je řešen povelom ZHELP obvyklým i v jiných konver-

začnících systémech. Je-li příkaz zadán bez parametru, vypíše seznam příkazů a jejich případných zkratek. Při uvedení parametru, kterým je jméno příkazu nebo jeho zkratka, vypíše text s údaji o syntaxi, sémantice a funkci příkazu na terminál. Jde v podstatě o část příručky uživatele, která je zpracována editorem, textu PÉS a speciálním programem upravena.

Za zmínku stojí způsob uložení těchto textů. Popis každého příkazu tvoří větu složenou z několika řádků textu. Věta je opatřena klíčem odvozeným ze jména nebo zkratky příkazu a uložena do indexsekvenčního souboru s proměnnou délkou věty. Vyhledání tak trvá velmi krátkou dobu. Pokud se změní vlastnosti příkazů dalším vývojem systému SFDL/SCL, bude popis v příručce i vysvětlující text aktualizován současně a jen jednou.

Pomocnou funkci mají i ostatní povely konverzačních procedur, které po realizaci vypisují aktuální stav nastavených vstupů a výstupů.

K informovanosti návrháře přispívají také příkazy jazyka SCL pro výpis informací o objektech modelu, pracovních strukturách definovaných v plánu simulace a jejich hodnotách. Návrhář k nim má přístup pomocí jejich symbolických názvů, tak jak byly deklarovány popisem modelu (jazyk SFDL) a plánu simulace (jazyk SCL). Adresář objektů je součástí modelu obvodu.

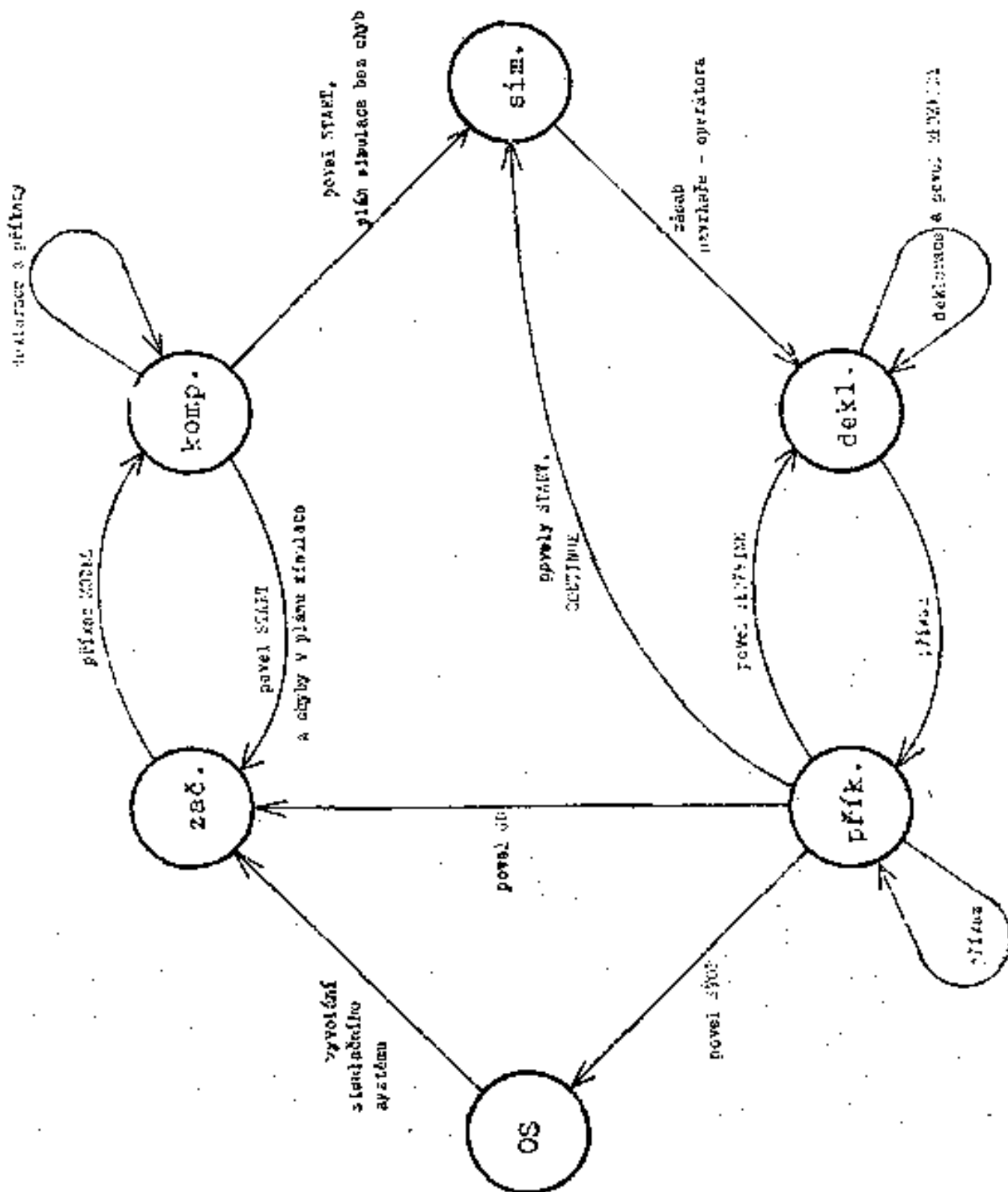
5. ZÁVĚR

Systémy automatizovaného návrhu (CAD - computer aided design) se v současnosti stále více uplatňují v nejrůznějších oborech technické tvůrčí činnosti. Důležitým měřítkem jejich kvality je schopnost účinné spolupráce stroje s člověkem, která není myslitelná bez existence konverzačního režimu. V tomto směru mají největší zkušenosti právě programátoři a je tedy na nich, aby je uplatnili v tvorbě systémů, které budou návrháři stále více potřebovat a uplatňovat.

Popisované konverzační procedury systému SPDL/SCL předpokládají konverzaci v řádkovém režimu. S příchodem dokonalejších komunikačních terminálů, zvláště obrazovkových displejů, se budou konverzační procedury dále vyvíjet. Očekává se rozvoj panelového typu konverzace s rozdělením obrazovky na okna se skupinami vzájemně vázaných údajů, které jsou průběžně aktualizovány. Příkladem může být simulační systém ISIS vyvíjený ve VÚMS k.ú.o.

6. LITERATURA

- /1/ Zendulka J., Hruška T., Kastner A., Prágelová M.:
Simulační systém SPDL/SCL. Definice jazyků, uživatelská příručka. Katedra SAPO FE VUT Brno. 1984, 91 s.
- /2/ Dolák J.:
Komunikační modul interpretu SCL. Diplomová práce FE VUT Brno. 1984
- /3/ Kastner A.:
Prostředky pro automatizovaný návrh číslicových obvodů na logické úrovni. Kandidátská disertační práce FE VUT Brno. 1985, 114 s.
- /4/ Blatný J., Zendulka J., Hrbáčková M.:
Údajové struktury a algoritmy pod systému řízení simulace simulačního systému SPDL/SCL. Podkladová zpráva úkolu SPZV III-8-3/11. Katedra SAPO FE VUT Brno, 1986



Obr. 1 Průběh simulačního experimentu