

ZRIMEC Tatjana, KOSER Dušan, SIMIČ Nikola, JAGODIČ Marko
 Iskra, Elektrovezje, TTS
 Tržaška 2
 61000 Ljubljana

PROGRAM ZA SIMULACIJO IN DIAGNOSTIKO NAPAK V LOGIČNIH
 VEZJIH Z UPORABO METOD UMETNE INTELEGENCE

A PROGRAM FOR THE SIMULATION AND FAULT-DIAGNOSIS OF
 LOGIC CIRCUITS USING METHODS OF ARTIFICIAL INTELLIGENCE

POVZETEK- Uporaba metod umetne inteligence je omogočila drugačen pristop k problemu losične simulacije in diagnostike digitalnih vezij. V referatu je opisan program, ki se razvija za preprosto preverjanje delovanja vezij v razvoju ter odkrivanje napak pri realiziranih vezjih. Program ima lastnosti ekspertnega sistema. Z uporabo znanja na področju digitalne tehnike in izkušenj strokovnjakov bo program v pomoč razvijalcem pri opravljanju njihovih nalog.

ABSTRACT- A different approach to the problems of logic simulation and the diagnosis of the logic circuits was enabled by the use of methods of artificial intelligence. A program for simple testing of circuits under development, and debugging of circuits in production is being developed and is described in the paper. The program has the characteristics of an expert system. By the use of knowledge on digital engineering techniques the program will be of great help to the design engineers.

UVOD

Uporaba računalniške simulacije je elegantna metoda preverjanja pravilnosti delovanja električnega vezja, ko je še v razvoju in je realizirano samo na papirju. Za takšno delo potrebujemo računalnik in program za simulacijo. Vendar, če sta oba posoja izpolnjena, še ne pomeni, da bo uporabnost takega programa zelo velika. Potreben je dodatni posoj, ki je udobna komunikacija s programom. Mnogi razvijalci čutijo odpor do računalnika prav zaradi težave komunikacije s sistemom ali programi. Zato raje realizirajo prototip vezja in na modelu odkrivajo napake, kot da bi jih na papirju že prej odpravili. Pri tem je izsuba časa in denarja lahko občutna.

Z uporabo metod umetne inteligence [1,10] ter z razvojem programskih jezikov [3,11] lahko gradimo programe, ki so komunikativni, razumljivi ter imajo inteligentno obnašanje. Na ta način lahko premasamo odpor uporabnika do računalnika, da začne uporabljati računalnik kot orodje pri izvrševanju konkretnih nalog.

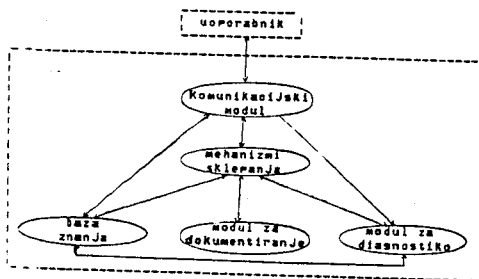
Danes obstaja množica inteligentnih računalniških programov na različnih aplikacijskih področjih, ki so z uporabo metod umetne inteligence svetovalci pri reševanju problemov. Pogosto jih imenujemo ekspertni sistemi [2,5,12,14,15]. Na specifičnem problemskem prostoru delujejo podobno kot človek-strokovnjak, ki zna sklepati, svetovati in razlagati svoje odločitve. Za vse te programe je značilno, da vsebujejo specializirano bazo znanja, mehanizme sklepanja, s katerimi uspešno uporabljajo znanje pri reševanju konkretnih problemov ter komunikacijski modul za enostavno komunikacijo s sistemom [4,7].

Tudi električna vezja so privlačna tema za snovalce sistemov z uporabo metod umetne inteligence. Klasični pristop obravnavanja električnih vezij temelji na natančni numerični analizi. Strokovnjaki za električna vezja pa uporabljajo različne strategije pri razmišljanju o vezjih in dosegaajo s tem hitrejšo in bistveno boljše rezultate. Z uporabo takega znanja so nastala v M.I.T. v ZDA, znamenita dela. Prvi dosežek je bila implementacija intuitivnih idej, ki jih ljudje uporabljajo pri kvalitativni analizi vezja [6]. Tako je nastala serija programov pod imenom EL. Kmalu nato sta nastala še dva sistema, in sicer WATSON za lociranje napak in DESI za sintezo vezij. Naslednji uspešni program je SYN [4], uporaben za reševanje različnih problemov pri sintezi vezij. Nazadnje je nastal zelo učinkovit sistem za kvalitativno analizo vezij, imenovan QUAL [9]. Oba sistema EL in QUAL sta zaradi učinkovitosti, novosti in moči uporabljenih idej vzbudila zanimanje javnosti tudi izven okvira umetne inteligence. Tudi pri nas je bil razvit podoben ekspertni sistem DRES [8] za analizo vezij z diodami in upori.

Programi kot so EL in QUAL obravnavajo analoška električna vezja. Namen našega programa pa je analiza in diagnostika digitalnih vezij.

PROGRAM

Program, ki smo ga razvili, se razlikuje od standardnih programov za simulacijo. Zgradba programa je podobna klasičnemu ekspertnemu sistemu. Sestavljen je iz več modulov prikazanih na sliki 1.



Sl. 1- Zgradba programa

Komunikacijski modul omogoča enostavno komunikacijo z uporabnikom. V tem modulu podajamo podatke vezja in izbiramo sledeče možnosti uporabe programa: simulacija vezja, diagnostika vezja, pojasnila o

Podatkih v bazi in dokumentiranje. Za vpisovanje podatkov vezja imamo na izbiri vodeni in direktni način vpisa. Pri vodenem vpisu program z podprašanji pomaga uporabniku kako in kaj naj vpiše (sl 2). Direktni vpis omogoča hitrejšo komunikacijo.

Opis vezja je zelo preprost. V poljubnem vrstnem redu oštevilčimo elemente v shemi in jih vpisujemo z imenom in vhodi v računalnik (sl 3). Imena so izbrana tako, da je mogoče iz njih takoj prepoznati funkcijo elementa. Vhode podamo s tem, da povemo na kateri izhod je vhod priključen, če uporabnik ne poda začetnih vrednosti elementov, si jih sistem določi in postavi sam (na primer števece postavi na nič). Za simulacijo vezja je potrebno, da povemo, kaj želimo razen izhodov, zasledovati in na kakšen način bodo podani rezultati: tabelarično ali s časovnim diagramom. Časovni diagram je primerna oblika prikazovanja analize sekvenčnih vezij in je tudi udomačena pri razvijalcih.

Na sliki 2 podajamo del dialoga med računalnikom in uporabnikom pri opisovanju vezja.

```
Vpis elementov vezja je v poljubnem vrstnem redu.
Če je vezje že opisano, vpiši številko vezja ali 0
! : 0.
Zanima te te vodila ? "da-ne"
! : da.
Vpiši ime funkcijskega sklopa ali "konec"
! : and.
Vpiši [vhodi,vhod2] ter od kod sta (v/2)
! : [v1/3,+3/3].
Vpiši začetne vrednosti za element -and- :
! : 0!.
Vpiši ime funkcijskega sklopa ali "konec"
! : d.duel.
Tega elementa ne poznam !
Vpiši ime funkcijskega sklopa ali "konec"
! : deal.
Vpiši [v1,vhod,vhod] ter od kod sta (v/2)
! : [v1/1,+2/3].
Vpiši začetne vrednosti za -deal- :
! : 0!.
Vpiši ime funkcijskega sklopa ali "konec"
.....
```

Sl. 2-Del komunikacije

Komunikacijski modul omogoča pojasnila o tem kaj program lahko dela, kako izberemo določeno delovanje ter kateri elementi so v bazi znanja.

Baza znanja vsebuje osnovne funkcijske elemente: in, ali, nein, ex.ali, neali, d.celice, števece z binarno sekvenco,uro kot generator impulzov ter sestavljene elemente kot n.pr. in-ali element. Baza bo razširjena z drugimi spominskimi celicami, registri ter spominskimi enotami(RAM). Vsak funkcijski element je opisan v naslednji obliki:

```
losel(Ime_elementa,[Vhodil],[Izhodil],Zakasnitve).
```

Losišne funkcije pa v obliki pravil. Na primer funkcija "neali" je podana z pravilom :

```
fncor([Vhodil],Izhod):-or([Vhodil],Izhod1),invert(Izhod1,Izhod).
```

(losišna vrata "neali" ima več vhodov "Vhodil" in en izhod "Izhod". Funkcijo tega elementa podamo z "ali" vrati in invertorjem izhoda).

Baza znanja vsebuje tudi opis delovanja, ki jo ima vsak element:

```
funk(dcel,zakasnilni_element_za:N_ciklov).
funk(inverter,nesacija_vhoda).
```

Za hitreje delovanje programa smo podali relacijo med binarnimi in desetiški številki.

Med posovorom z izkušenimi razvijalci električnih vezij smo dobili pravila o poteku testiranja vezij pri iskanju napak. Ta pravila se nahajajo v bazi znanja in bodo uporabljena kot nasveti ali vodila pri iskanju napak v vezjih.

Modul z mehanizmi sklepanja se losišno deli na več delov:

- priprava vezja za simulacijo,
- simulacija,
- usotavljanje pravilnosti oz.nepravilnosti delovanja vezja.

Program pripravi vezje za simulacijo tako, da poišče povezave med elementi, si ustvari sliko o vezju ter razdeli elemente v nivoje glede na to, kako potujejo signali v vezju. Poišče tudi vse povratne povezave, zlasti direktne. Program sam določa čas oz. število korakov, potrebnih za določanje funkcije vezja, lahko pa te podatke vpiše uporabnik.

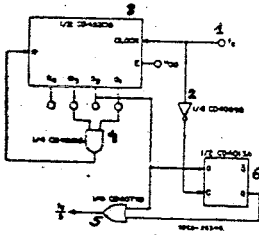
Simulacija poteka na zelo enostaven način. To, da smo elemente v vezju uredili v nivoje, zadostuje, da zasledujemo spremembe vhodnih signalov iz nivoja v nivo. Začnemo z najvišjim nivojem, v katerem so elementi odvisni od krmilnih signalov in končamo z najnižjim nivojem, v katerem so izhodi. Direktne povratne vezave upoštevamo v istem časovnem trenutku, ali koraku, ko se pojavi karakteristična sprememba, ki spremeni časovni potek vezja. V tem trenutku ponovimo simulacijo od tistega nivoja dalje, v katerem pride povratna vezava. Ostale povratne povezave upoštevamo vedno v naslednjem koraku. S takim potekom simulacije lahko dobimo realno sliko delovanja vezja upoštevajoč še vse lastne zakasnitve elementov.

Modul za dokumentiranje hrani vezja, s katerimi imamo namen še delati. Vse popravke na shemi z lahko prenesemo v računalnik brez ponovnega opisa vezja. Ta modul je uporaben tudi pri odkrivanju napak v realiziranih vezjih.

Modul za odkrivanje ali lociranje napak v vezju je še v razvoju. Mišljeno je, da bi pomagal razvijalcem pri testiranju vezij z simulacijo in nasveti.

PRIMER DELOVANJA PROGRAMA

Izbrali smo vezje prikazano na sliki 3-a in ga simulirali z programom. Oštevilčili smo elemente v vezju in jih direktno vpisali računalnik (sl 3-b).



Sl. 3-a Vezje

```

(ura,(s/0)).
Einvert,(col/1)).
Cstevec,(col/1,(s/4)).
Cand,(col/3,(s/3)).
Cor,(c/2/3,(s/5)).
Cdeci,(c/2/3,(s/2)).

```

Sl. 3-b Vpis vezja
v računalnik

Nismo podali začetne vrednosti. Program je postavil vse začetne vrednosti "0" razen, vrednosti na elementu 4, ki je reset impulz in je dobil vrednost "1". S tem smo omogočili, da števec začne šteti od "0" naprej. Program je poiskal povezave med elementi in uredil vezje v štiri nivoje. Dobil je povratno povezavo med elementi 3 in 4. Program je izračunal stevilo korakov za prikaz funkcije vezja. Na sliki 4 podajamo umesne rezultate programa pri urejevanju vezja.

Vse povezave
[1+2,1+3,4+3,3+4,3+5,5+6,2+8]

Potrebne povezave
[1+2,1+3,4+3,3+4,5+5,3+6,2+8]

Povratne povezave
[4+3]

nivo(1,(1)).
nivo(2,(2,3)).
nivo(3,(8)).
nivo(4,(4,5)).

Je primer vezja e/

```

vezje(1).
f(ura,(1/1)-col(1)...1).
f(invert,(1/1),(1)...2).
f(stevec,(1/1,(s/4),(s+2,-s+4)...3).
f(and,(1/3/3/3),(1)...4).
f(or,(2/3,(s/3),(1)...5).
f(deci,(1/2/2/3),(1)...8).
mnozeli((1,2,3,4,5,8)).

```

Sl. 4-Prikaz umesnih rezultatov in dokumentiranje vezja

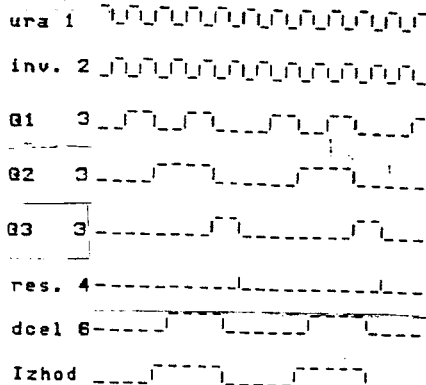
Izbrali smo simulacijo vezja, zato je program izpisal vprašanje:

```

Podaj množico izhodov :
1:(1,2,3,4,8,5).
Zelis prikaz - časovni diagram ali tabelo ?
1:časovni_diagram.

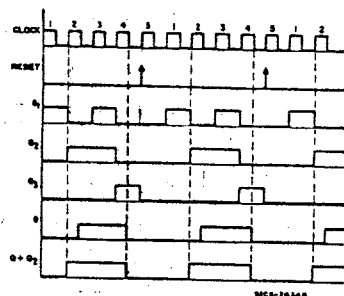
```

Zeleli smo časovni diagram izhodov vseh elementov, zato smo dobili diagram prikazan na sliki 5-a. Za primerjavo smo podali na sliki 5-b še predpisani časovni diagram za to vezje.



Sl. 5-a časovni diagram dobljen z simulacijo

ZAKLJUČEK



Sl. 5-b željeni časovni diagram

Program je napisan v programskem jeziku Prolog in je implementiran na računalniku DEC-10. Prolog je zelo uporaben za različne simulacije vezij ter tudi predstavitel tovrstne baze znanja. Možnost predstavitve znanja v obliki relacij omogoča sklepanje v obeh smereh. Lahko postavimo na primer tudi vraščanje:

"Kakšni so vhodni pri določnem izhodu"

Ti že vsajani mehanizmi Prologa so zelo uporabni tako pri logični analizi vezja, kot tudi pri odkrivanju narek v vezju.

Program omogoča analizo decizijskih in sekvenčnih vezij. V Prologu ne obstajajo nikakšne omejitve glede podatkov, zato je možno analizirati vezja s poljubnim številom elementov. Možnost simulacije realnega delovanja vezja z upoštevanjem fizikalnih lastnosti elementov je lahko zelo uporabna za odkrivanje občutljivosti vezja na spremenjene lastne zakasnitve elementov. To pa je zelo važen podatek pri izbiri elementov za realizacijo vezja.

S hitrim in preprostim načinom spreminjanja baze znanja lahko razširimo uporabnost programa.

Uporaba programa bo omogočala razvijalcem lažje in hitrejšo reševanje nalog.

LITERATURA

- [1] Bratko I. : Inteligentni informacijski sistemi; Univerze E. Kardelja v Ljubljani, Fakulteta za elektrotehniko, 1981.
- [2] Bratko I., Lavrač N., Mozetič I., Zimec T., Horvat M., Rode P. : The Development of an Expert System for Intensive Medical Treatment; Proceedings of Technical Conference: Theory and Practice of Knowledge - Based, London, 1982.
- [3] Clocksin W.F., Mellish C.S. : Programs in Prolog; Springer Verlag, 1981.
- [4] Stefik M. et al. : The organisation of Expert Systems, A Tutorial; Artificial Intelligence 18, pp 135-173, 1982.
- [5] Duda R. et al. : Development of the Prospector Consultation System for Mineral Exploration; Final Report SRI International, 1978.
- [6] Stallman R. M., Sussman G. J. : Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuits Analysis; Artificial Intelligence 9, 1975.
- [7] Gams M., Lavrač N. : Ekspertni sistemi; Delovno poročilo IJS - DP - 1867; Institut Jožef Stefan, Ljubljana, 1978.
- [8] Gams M. : Predstavitev in uporaba znanja v sistemih za interakcijo z uporabnikom; Magistrska naloga; Univerza E. Kardelja v Ljubljani, Fakulteta za elektrotehniko, 1981.
- [9] de Kleer, J. : Causal and Theoretical Reasoning in Circuit Recognition, doktorat M.S.T., 1978.
- [10] Nilsson N.J. : Principles of Artificial Intelligence; Tracy Publishing Co. 1980.
- [11] Pereira F., Pereira L.M. : User's Guide to DEC-10 Prolog; Dept. of Artificial Intelligence, University of Edinburgh, 1978.
- [12] Poppel H.E. Jr., Miller R. A., Myers J.D. : INTERNIST-1, an Experimental Computer-based Diagnostic Consultant for General Internal Medicine; The New England Journal of Medicine, 1982.
- [13] Shortliffe E.H. : Computer-Based Medical Consultation: MYCIN; Elsevier Scientific Publishing Co., 1976.
- [14] Lavrač N., Mozetič I., Zimec T., Bratko I., Grad A., Čerček B., Horvat M., Rode P. : Ekspertni Sistem za Diagnostiko MotenJ Srčnega Ritma; 7 Simpozij iz Informatike, Jahorina, 1983.