

GENERIRANJE SCHEME IZ TEKSTOVNEGA OPISA LOGIČNEGA VEZJA S PROGRAMOM SCHEMGEN

S. Gruden, M. Leban, B. Zajc

KLJUČNE BESEDE: računalniško načrtovanje vezij, integrirana vezja, logična vezja, generiranje shem, SCHEMGEN, računalniški program

POVZETEK: Opisali bomo program, ki na osnovi tekstovnega opisa logičnega vezja generira shemo, ki si jo lahko ogledamo na zaslonu ali pa shranimo za nadaljnjo obdelavo. Opisali bomo postopke za razmeščanje simbolov celic in njihovo medsebojno povezovanje.

SCHEME GENERATION FROM TEXTUAL DESCRIPTION OF THE LOGICAL CIRCUIT WITH THE PROGRAM SCHEMGEN

KEY WORDS: computer aided circuits design, integrated circuits, logic circuits, sheme generation, SCHEMGEN, computer program

ABSTRACT: A program for scheme generation from textual description of the logical circuit is described. The resulting scheme can be viewed immediatly or saved for further use. Algorithms for placement and routing of cell symbols are shown.

1. UVOD

Eden od korakov računalniškega načrtovanja integriranih vezij je vnos opisa vezja v računalnik. Opis vezja lahko vnesemo grafično tako, da na zaslon narišemo shemo vezja, ali tekstovno tako, da opišemo vse celice vezja in njihove medsebojne povezave. V drugem primeru imamo shemo vezja narisano samo na papirju ali pa še to ne. Tudi če dobimo tekstovni opis vezja kot rezultat dela katerega drugega programa (programi za minimizacijo logičnih vezij, programi za generiranje vezja iz matematičnega opisa vezja itd.), sheme nimamo nikjer narisane. Le-to pa velikokrat potrebujemo zaradi nepreglednosti tekstovnega opisa vezja. Ročno risanje sheme je posebno za velika vezja zamudno opravilo, pri katerem lahko naredimo veliko napak. Naslednji prispevek opisuje program in postopke za avtomatsko generiranje sheme iz tekstovnega opisa vezja. Program deluje za logična vezja, ki so sestavljena iz standardnih celic. Opisi teh celic morajo biti v knjižnici, ki jo program prebere na začetku. Program rabi podatke o velikosti in obliki celic ter podatke o vrsti in pozicijah priključkov celic. Priključki se lahko nahajajo na vseh štirih robovih celic. Datoteka z opisom vezja, ki naj ga program nariše, mora vsebovati vrstico, v kateri je ime vezja in imena vozlišč, ki so vhodi in izhodi celotnega vezja. Vsaka celica, ki se nahaja v vezju pa mora imeti svojo vrstico, v kateri je opisan tip celice in vozlišča, kamor so njeni priključki vezani. Program loči tri vrste celic: vhodne, izhodne in aktivne celice vezja. Vhod vhodne celice je zunanji (vhodni) priključek vezja, izhod vhodne celice pa je vezan v vezje. Podobno je izhod izhodne celice zunanji (izhodni) priključek, vhod izhodne celice pa je vezan v vezje. Aktivne celice imajo vse svoje vhode (vsi

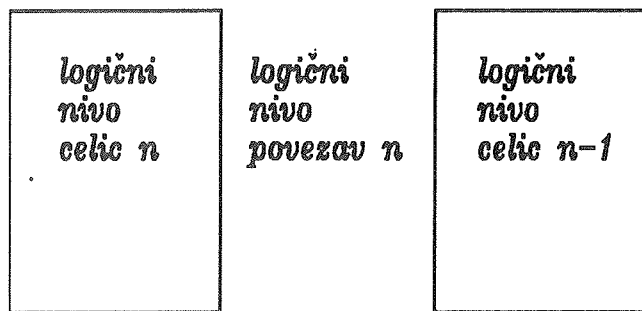
vhodi morajo namreč biti definirani) in nekatere ali vse svoje izhode vezane v vezje. Izhodi celic med sabo ne smejo biti povezani. Program vse vhodne celice postavi skrajno levo, vse izhodne celice pa skrajno desno na shemi. Aktivne celice vezja porazdeli po nivojih. Povezave, ki povezujejo celice na dveh sosednjih nivojih, postavi v vmesni prostor med tema dvema nivojema, povezave, ki povezujejo celice na bolj oddaljenih nivojih, pa postavi nad ali pod vse celice.

2. OPIS DELOVANJA PROGRAMA

Program SCHEMGEN je sestavljen iz treh večjih delov. Prvi del prebere vhodne datoteke (opis vezja in knjižnico elementov). Drugi del programa generira shemo, tretji del pa je namenjen obdelavi dobljenih rezultatov (prikaz sheme na zaslon in tvorjenje izhodne datoteke s podatki o shemi). Glavni del programa SCHEMGEN je drugi del. Ta del sestavljajo podprogrami za določanje nivojev, funkcije za razmestitev celic, funkcije za povezovanje celic med sabo in funkcije za namestitev spojišč na mesta, kjer je to potrebno.

2.1 DOLOČANJE NIVOJEV CELIC

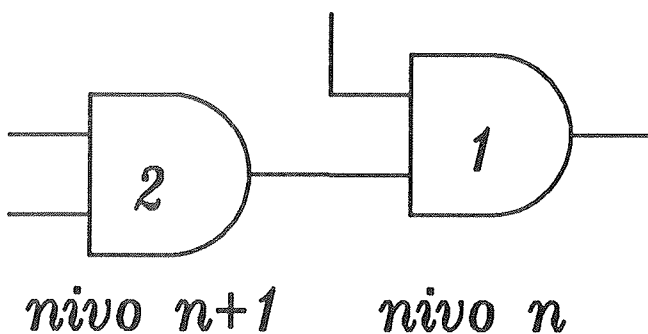
Preden celice vezja dejansko razmestimo na določene koordinate, je potrebno določiti nivoje, na katerih se posamezne celice nahajajo. Nivo celic v bistvu pove, kako daleč od izhoda (oz. v katerem stolpcu glede na izhod) se določena celica nahaja. Skrajni levi nivo celic ima številko 0, nivoji pa naraščajo od desne proti levi.



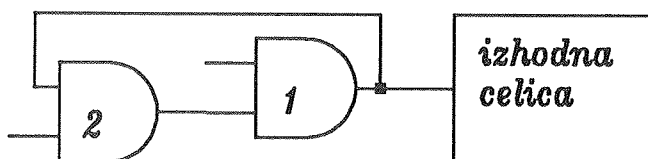
Slika 1: Ponazoritev definicije nivojev celic in nivojev povezav

Nivo povezav s številko n pa pomeni tisti vmesni prostor (kjer se nahajajo povezave), ki se nahaja med nivojema celic $n-1$ in n (slika 1).

V programu SCHEMGEN so torej nivoji celic definirani tako, da so vse izhodne celice postavljene na nivo 0. Celica, ki je bolj oddaljena od izhoda, je na višjem nivoju. Pravilo za določanje nivojev je v grobem tako: Imejmo dve celici. Če vhod prve celice, ki se nahaja na nivoju n , krmili druga celica, potem lahko to drugo celico postavimo na nivo $n+1$ (slika 2). Za vezja brez povratnih vezav so nivoji celic enolično definirani, medtem ko za vezja s povratnimi vezavami nivoje lahko določimo na več različnih načinov. Tak primer je prikazan na sliki 3. Izhodna celica je na nivoju 0. Po pravilu za določanje nivojev postavimo celico št. 1 na nivo 1 in celico št. 2 na nivo 2. Vhod celice št. 2 pa krmili celica št. 1, ki bi jo zato lahko postavili tudi na nivo 3.



Slika 2: Določanje logičnih nivojev

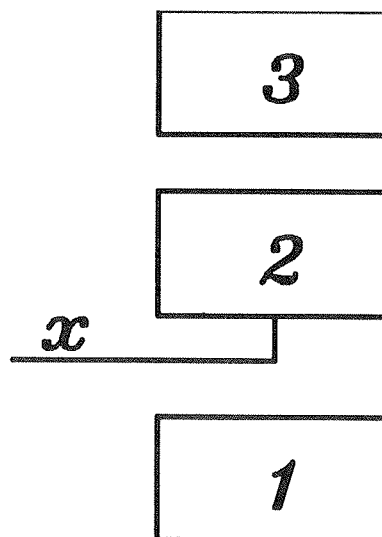


Slika 3: Primer določanja nivojev za vezje s povratnimi povezavami

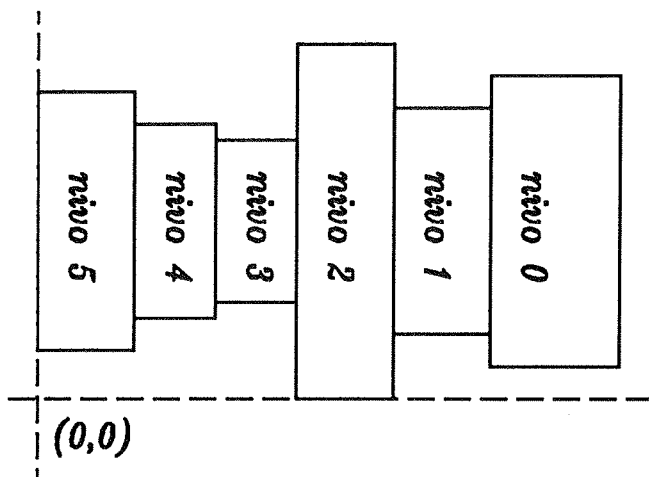
Da lahko sploh določimo nivoje celic, moramo najprej za vsako vozlišče ugotoviti, katera celica ga krmili. Nivoje nato poiščemo na sledeči način. Začnemo z eno od izhodnih celic, ki jo postavimo na nivo 0. Za en vhod te celice poiščemo krmilno celico in jo postavimo na nivo 1. Nato zopet za en vhod nove celice poiščemo krmilno celico ter jo postavimo na naslednji višji nivo celic. Postopek ponavljamo, dokler ne pridemo do ene od vhodnih celic ali do celice, ki ima nivo že določen. Pri tem smo za vsako celico izkoriščali le po en vhod. Isti postopek rekurzivno ponovimo tudi za ostale vhode celic ter za vse izhodne celice. Problem, da je isto celico mogoče postaviti na različne nivoje, rešimo tako, da celico pač pustimo na tistem nivoju, kamor smo jo prvič postavili. Zatem poiščemo najvišji nivo celic, ki je pri tem nastopil, vsem vhodnim celicam celotnega vezja pa priredimo še za ena višji nivo. Vse vhodne celice se bodo torej nahajale na najvišjem nivoju (skrajno levo na shemi), vse izhodne celice pa na nivoju 0 (skrajno desno na shemi).

2.2 OPIS RAZMEŠČANJA CELIC

Ko je določeno, katerim nivojem bodo celice pripadale, je potrebno izračunati še dejanske koordinate, na katerih se bodo celice v shemi nahajale. Celice na skrajnem levem nivoju postavimo na določeno začetno x koordinato. Nivoje postavljamo po vrsti od leve proti desni. Na vsakem nivoju poiščemo najširšo celico, naslednji nivo pa postavimo na desni rob te celice. Nivoje torej postavimo enega poleg drugega, saj zaenkrat še ne vemo, koliko prostora za povezave bomo rabili med celicami. Na posameznih nivojih celice postavimo eno nad drugo. Če ima celica priključke na zgornji ali spodnji strani, je potrebno celice razmakniti za toliko, da naredimo prostor za povezave, ki bodo priključene na te priključke (slika 4, povezava x). Nato celice še uredimo tako, kot so urejene na sliki 5. Nivo z največjo skupno višino postavimo torej na y koordinato izhodišča, vse ostale nivoje pa simetrično glede na linijo, ki teče skozi sredino tega najvišjega nivoja.



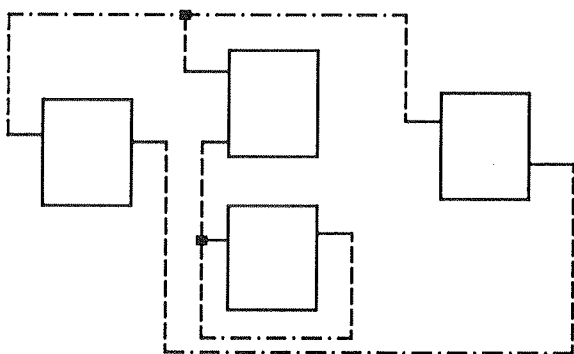
Slika 4: Primer, ko ima celica priključek na spodnji strani



Slika 5: Končna razporeditev celic vezja

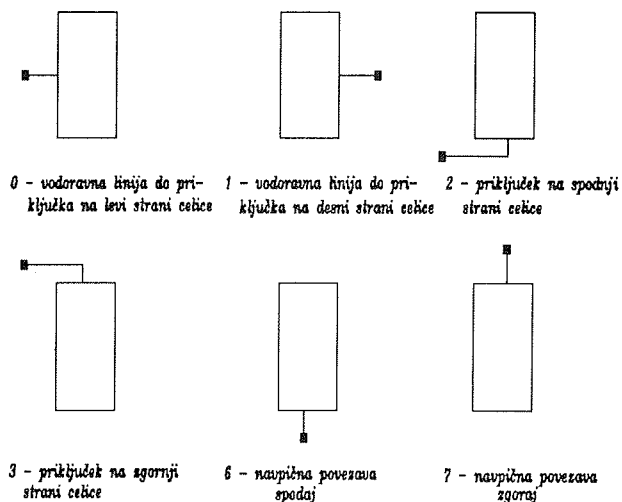
2.3 POVEZAVE MED CELICAMI

V programu so za povezovanje uporabljene tri vrste povezav. Prva vrsta so povezave, ki povezujejo skupaj priključke na različnih nivojih povezav (linije šrafirane z - - - - na sliki 6). Te povezave v celoti tečejo nad ali pod celicami, nikoli pa med celicami določenega nivoja celic. V nadaljnjem tekstu se te povezave na kratko imenujejo vodoravne povezave. Druga vrsta povezav so povezave, ki med sabo povezujejo priključke na istem nivoju povezav, poleg tega pa povezujejo te priključke še z ustreznimi vodoravnimi povezavami (linije šrafirane z --- na sliki 6). Te povezave se v nadaljnjem tekstu imenujejo vertikalne povezave.



Slika 6: Prikaz vseh treh vrst povezav vezja

Tretja vrsta povezav pa povezuje priključke celic z ustreznimi vertikalnimi povezavami (linije šrafirane s polno črto na sliki 6). Te povezave se v nadaljnjem tekstu na kratko imenujejo horizontalne povezave, čeprav to niso vedno le vodoravne črte. Možne oblike horizontalnih povezav so prikazane na sliki 7. Poleg tega pa se izkaže, da je potrebno včasih vodoravne dele teh povezav še dodatno nalomiti, da ne pride do prekrivanja z drugimi povezavami.



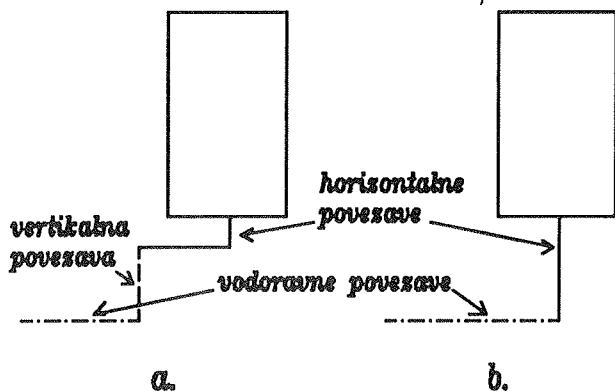
Slika 7: Možne oblike horizontalnih povezav

Vsakemu vozlišču lahko pripada največ ena vodoravna povezava, zato podatke o teh povezavah hranimo kar med podatki o vozliščih. Podatki, ki jih rabimo so: začetni in končni nivo povezav, med katerima povezava teče, odmika na obeh nivojih, kjer se vodoravna povezava točno konča in še višina, na kateri se povezava nahaja. Lahko se zgodi, da vodoravna povezava za določeno vozlišče ni potrebna, ker so vsi priključki med sabo povezani na enem in istem nivoju povezav. Podatki o vertikalnih povezavah so: y koordinata spodnjega in zgornjega roba vertikalne povezave, nivo povezav, na katerem se povezava nahaja, odmik od izhodišča nivoja in vozlišče, ki mu povezava pripada. Podatki o horizontalnih povezavah pa so: nivo povezav, na katerem se povezava nahaja, koordinate začetka in konca horizontalne povezave, vrsta horizontalne povezave (slika 7) in vozlišče, ki mu pripada povezava.

2.4 POVEZOVANJE CELIC

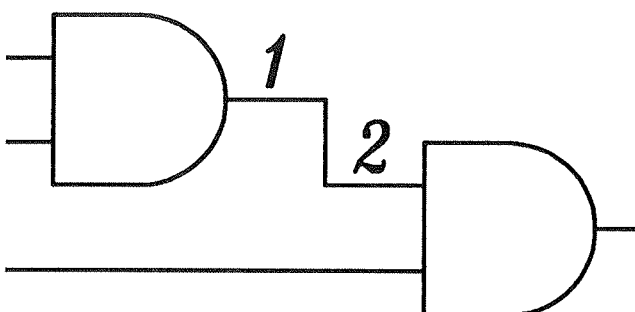
Najprej v grobem izračunamo koordinate vseh vrst povezav vezja. Vhodi in izhodi celotnega vezja ne potrebujejo nobenih povezav, zato jih ni potrebno obdelati. Za ostale priključke pa je treba povezave izračunati. Najprej za horizontalno povezavo določimo njeno vrsto (glej sliko 7). Zaenkrat se vpiše le to, na kateri strani celice se horizontalna povezava nahaja (primeri 0, 1, 2 in 3). Vsaki horizontalni povezavi določimo nivo, kjer se nahaja, x koordinato začetka, višino, na kateri se nahaja, in vozlišče, kateremu pripada povezava. Označimo še, da to zaenkrat ni lomljena povezava. Horizontalno povezavo, ki se nahaja nad, oz. pod celico (slika 7, tip povezave 2 in 3) postavimo en razdelek nižje ali višje glede na priključek celice. Na vodoravne povezave so običajno priključene ustrezne vertikalne povezave. Če pa ima npr. neka celica priključek spodaj in se hkrati nahaja čisto na dnu nivoja celic, vodoravna povezava pa tudi teče pod celico, bi bilo škoda, da bi bila povezava dvakrat nalomljena (slika 8.a). Zato rajši potegnemo povezavo kar navpično do vodoravne povezave (slika

8.b). Vse povedano velja tudi, če se priključek nahaja na vrhu celice, celica na vrhu nivoja celic in vodoravna povezava nad celico. Ker še ne vemo, ali se bo ustrezna vodoravna povezava res nahajala na ustrezni strani celice, si zaenkrat samo zapomnimo ustrezno x koordinato priključka. Začetke in konce vseh vodoravnih povezav določimo tako, da bodo le-te dosegle vse priključke, ki jih morajo povezati. Nato obdelamo še vertikalne povezave. Te povezave določimo tako, da segajo od najnižjega do najvišjega priključka na danem nivoju povezav za ustrezno vozlišče.



Slika 8: Horizontalna povezava, ki teče navpično do ustrezne vodoravne povezave

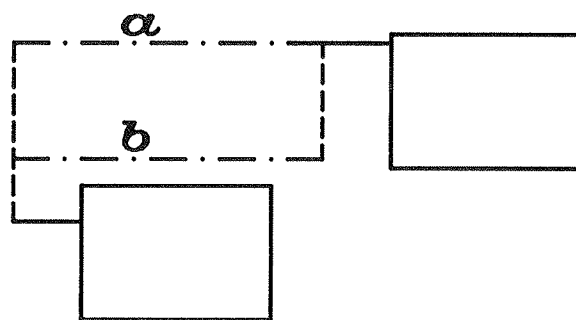
Vertikalne povezave zatem razporedimo po vmesnih prostorih med nivoji celic. Preštejemo, koliko priključkov z leve in koliko z desne povezujejo vertikalne povezave danega nivoja povezav. Povezave razporedimo tako, da tiste z najmanjšo razliko med številom priključkov na levi in desni (lahko je negativna) postavimo najbolj desno, ostale pa z naraščanjem razlike postavljamo proti levi. S tem dosežemo, da so povezave z velikim številom priključkov na eni strani bolj pomaknjene na to stran (skupna dolžina povezav je tako manjša). Nato vse nivoje celic (ki so bili do sedaj skupaj) pomaknemo proti desni za toliko, kolikor je vrinjenih vertikalnih povezav na pripadajočem nivoju povezav.



Slika 9: Primer, ko vodoravna povezava ni potrebna

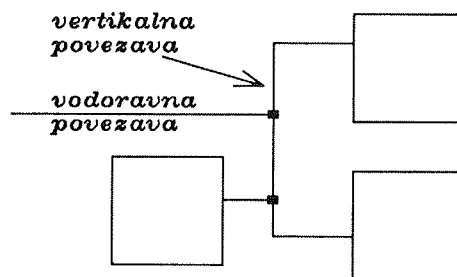
Velikokrat se zgodi, da povezave med različnimi nivoji sploh niso potrebne (povezava med priključkoma 1 in 2 na sliki 9), ker so vsi priključki povezani med sabo na istem nivoju s pomočjo vertikalnih in horizontalnih povezav. Take povezave zato izločimo. Nato za vse nivoje celic, prek katerih gre vodoravna povezava, izračunamo na zgornji strani najnižji in na spodnji strani najvišji

položaj, kamor bi vodoravno povezavo lahko postavili, ne da bi prišlo do prehajanja prek celic. Upoštevati moramo še dve stvari: če je na isti višini že postavljena neka druga vodoravna povezava, potem nove povezave na tem mestu ne moremo postaviti. Lahko pa se zgodi, da preglednost sploh ne bi trpela, tudi če bi povezavo oddaljili od celic. S tem bi namreč pustili prostor za povezave, za katere je bolj smiselno, da so celicam čim bližje. Povezavo lahko oddaljimo kljub temu, da imamo prostor zanj, če se skupna dolžina vertikalnih povezav, ki so vezane na vodoravno povezavo, pri oddaljevanju ne poveča (primer na sliki 10). Vodoravno povezavo poskusimo postaviti nad in pod celicami. Povezavo višamo (na zgornji strani), oz. nižamo (spodaj), dokler je to potrebno zaradi morebitnega prekrivanja, oz. dokler to smemo zaradi tega, ker se skupna dolžina vertikalnih povezav ne večja. Nato še pogledamo, na kateri strani bi bila vsota dolžin novo nastalih vertikalnih povezav manjša in vodoravno povezavo postavimo na to stran.



Slika 10: Skupna dolžina vertikalnih povezav (---) je neodvisna od tega, ali vodoravno povezavo (- · - · -) postavimo na mesto a ali b

Ko imamo vodoravno povezavo postavljeno na znano višino, do nje podaljšamo pripadajoče vertikalne povezave. Podaljšamo spodnji ali zgornji rob vertikalne povezave, ali pa celo nobenega (primer na sliki 11).

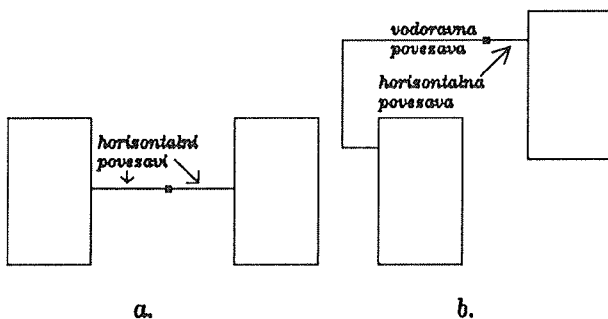


Slika 11: Podaljševanje vertikalne povezave do vodoravne ni potrebno

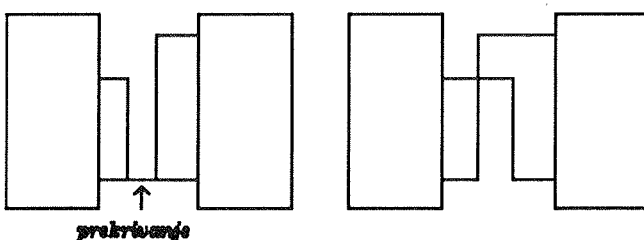
Nato se natančno izračunajo vse horizontalne povezave. Pri tem se lahko zgodi, da nekatere vertikalne povezave ne bodo več potrebne. Preverimo, če lahko horizontalno povezavo potegnemo kar navpično navzdol (tip povezave 6, glej sliko 7) ali navpično navzgor (tip povezave 7, glej sliko 7). To lahko storimo, če gre za spodnjo celico, ki ima ustrezni priključek spodaj, pripadajoča vodoravna povezava pa teče pod celicami, oz.

če gre za zgornjo celico na nivoju, ki ima priključek zgoraj, medtem ko vodoravna povezava teče nad celicami. V teh dveh primerih pripadajoče vertikalne povezave ne rabimo. Če pa je horizontalna povezava ena od običajnih (tip povezave 0, 1, 2 ali 3, glej sliko 7), ji določimo x koordinato konca tam, kjer se nahaja pripadajoča vertikalna povezava. Ker imamo določene x koordinate vseh vertikalnih in horizontalnih povezav, lahko izračunamo, kje točno se vodoravne povezave končajo. Nato pregledamo, če obstajajo vodoravne povezave z dolžino 0 in jih izločimo, vertikalno povezavo za ustrezno vozlišče na istem nivoju povezav (če obstaja) pa nazaj skrajšamo. Pobrismo tudi vse tiste vertikalne povezave, ki imajo dolžino 0. Take povezave ne vodijo niti do drugega priključka na istem nivoju, niti do ustrezne vodoravne povezave (to se lahko zgodi, če npr. izhod neke celice ni prav nikamor vezan). Do vertikalne povezave z dolžino 0 pa lahko pride tudi, če se horizontalni povezavi dveh sosednjih celic nahajata na isti višini (slika 12.a). Tudi v tem primeru je vertikalna povezava nepotrebna. V primerih, ko brišemo vertikalno povezavo, moramo včasih pobrisati tudi horizontalno povezavo, ki pelje do določenega priključka. V ta namen preštejemo, koliko priključkov vseh celic je sploh vezanih na ustrezno vozlišče. Če je tak priključek le eden, potem lahko horizontalno povezavo pobrišemo, če pa je teh priključkov več pa morajo med sabo biti nekako povezani, zato horizontalnih povezav ne smemo izbrisati (dva primera, ko vertikalna povezava ni potrebna, horizontalna pa je, sta na sliki 12).

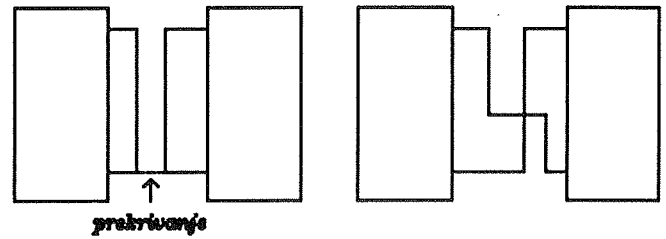
Program do tu že skoraj pravilno poveže med seboj vse celice. V dokaj pogostih situacijah pa lahko še vedno pride do neželenega prekrivanja povezav (glej sliko 13.a). V tej konkretni situaciji bi se lahko prekrivanju izognili že tako, da bi enostavno zamenjali med sabo obe vertikalni povezavi (dobili bi situacijo na sliki 13.b). Obstajajo pa tudi primeri, ko problem tudi na ta način ni rešljiv (glej sliko 14.a). V tem primeru moramo eno od povezav na mestu, kjer se prekrivata, tako zlomiti, da se



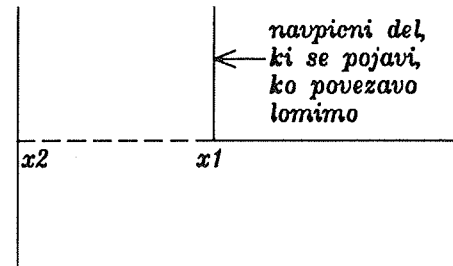
Slika 12: Primeri, ko vertikalna povezava ni potrebna



Slika 13: Neželeno prekrivanje povezav, ki ga je možno preprosto odpraviti

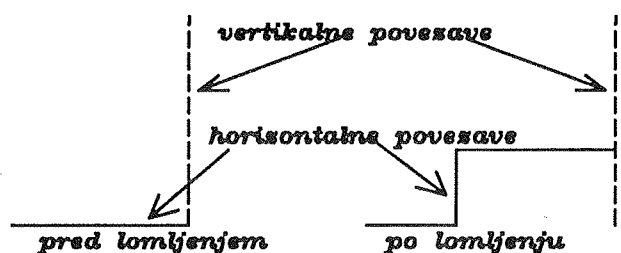


Slika 14: Neželeno prekrivanje povezav, ki ga je možno odpraviti le z lomljenjem povezav



Slika 15: Ponazoritev razmer, ko nalomimo horizontalno povezavo

izognemo prekrivanju (slika 14.b). V programu najprej ugotovimo, če se horizontalne povezave prekrivajo s kako drugo povezavo. Če prekrivanje obstaja, povezavo nalomimo. Pri tem pa se pojavi kos navpične črte (slika 15). Če bi horizontalno povezavo nalomili na katerikoli x koordinati, bi se lahko ta kos prekrival s katero od obstoječih vertikalnih povezav. Da se temu izognemo, razširimo razmik med nivojema celic za en razdelek. S tem pridobimo potrebno prosto mesto za ta kos navpične črte. Vse nivoje celic, ki se nahajajo desno od te povezave, moramo premakniti za eno mesto v desno. Če je horizontalna povezava, ki jo hočemo nalomiti, vezana na celico na desni strani, bomo prostor za kos navpične črte naredili na skrajni desni strani nivoja povezav (tipi povezav 0, 2 in 3, glej sliko 7), sicer pa na skrajni levi (tipi povezav 1, glej sliko 7). Nato vzamemo vodoravni del linije, ki se začne na x koordinati, kjer smo naredili prostor za navpični kos črte (X1 na sliki 15) in konča na x koordinati pripadajoče vertikalne povezave (X2 na sliki 15). Temu vodoravnemu delu linije nato tako dolgo višamo y koordinato, dokler prekrivanja z drugimi povezavami ni več in zato ta vodoravni del lahko pustimo na tem mestu. Zaključki popravljenih horizontalnih povezav se sedaj ne nahajajo več na isti višini kot prej. Če je bila taka horizontalna povezava prej vezana na enega od robov pripadajoče vertikalne povezave, ta vertikalna povezava sedaj ni več pravilno zaključena (primer na sliki 16). Zato pregledamo pripadajočo vodoravno pove-



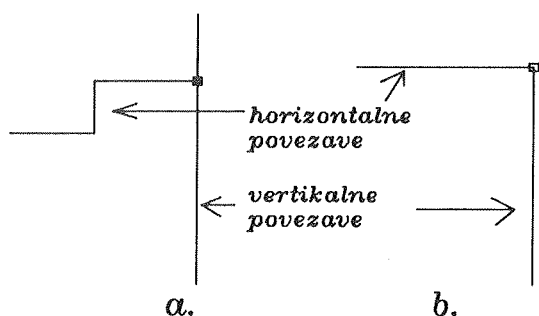
Slika 16: Nepravilna zaključitev vertikalne povezave po lomljenju horizontalne povezave

zavo in vse horizontalne povezave, ki so vezane nanjo ter na novo določimo spodnji in zgornji rob povezave.

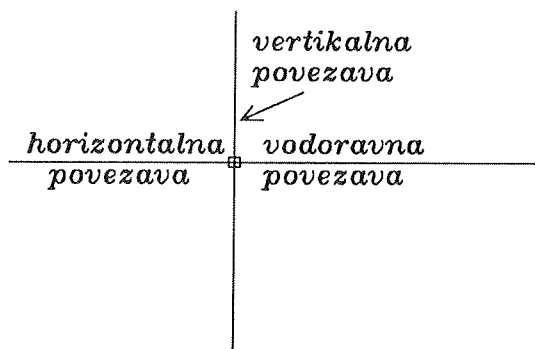
2.5 NAMESTITEV SPOJIŠČ NA POVEZAVE

Na koncu generiranja sheme vezja moramo še postaviti spojišča na tista mesta, kjer je to potrebno. To pa je tam, kjer se v eni točki stika tri ali več linij, ki so med sabo električno povezane. Podatki o spojiščih so pridruženi podatkom o vertikalnih in horizontalnih povezavah. Za vertikalne povezave se spojišče nahaja na stičišču z vodoravno povezavo. Za horizontalne povezave pa se spojišče nahaja na stičišču z ustrezno vertikalno povezavo (oz. vodoravno povezavo za tip povezave 6 ali 7, glej sliko 7).

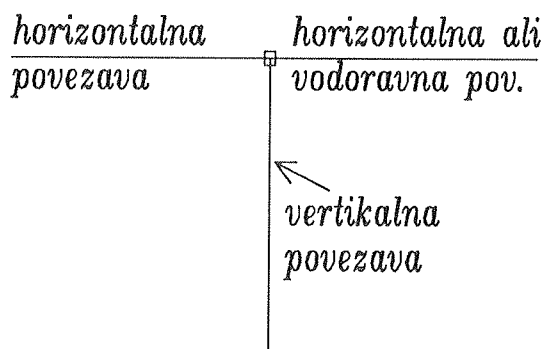
Najprej se določijo spojišča za horizontalne povezave. Za tipe horizontalnih povezav 0, 1, 2 in 3 (glej sliko 7)



Slika 17: Postavitev spojišč za horizontalne povezave

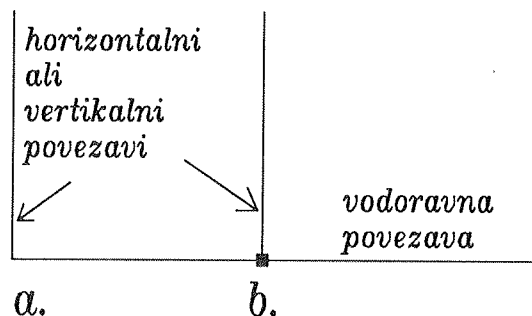


Slika 18: Poseben primer, ko spojišča ne priredimo horizontalni povezavi



Slika 19: Primer, ko je na konec vertikalne povezave poleg horizontalne priključena še ena povezava

poiščemo y koordinato, kjer se horizontalna povezava končuje (tu bo pozneje mogoče spojišče). Spojišče postavimo v primeru, ko se horizontalna povezava dotika ustrezne vertikalne povezave nekje v sredini (slika 17.a). Če se horizontalna povezava dotika vertikalne na robu, spojišča ni treba postaviti (slika 17.b). Obstaja možnost, da se na isti višini, kjer se nahaja horizontalna povezava, priključi še vodoravna povezava (slika 18). V tem primeru bi morali spojišče tudi narisati, vendar pustimo, da se bo le-to narisalo, ko se bodo obdelovale vertikalne povezave. Na konec vertikalne povezave, kamor je priključena horizontalna povezava, je lahko priključena še katera druga povezava (slika 19). Če je to vodoravna povezava, moramo spojišče narisati, ker se le-to ne bo narisalo, ko bomo obdelovali vertikalne povezave. V primeru, da je to horizontalna povezava pa spojišče narišemo le, če le-to še ni narisano na drugi horizontalni povezavi (spojišče torej dobi tista povezava, ki pride prva na vrsto). Za horizontalne povezave, ki so direktno priključene na vodoravno povezavo (tip 6 in 7, slika 7), narišemo spojišče le, če se horizontalna povezava dotika vodoravne nekje v sredini (slika 20.b), na robovih pa spojišča ne postavimo (slika 20.a).

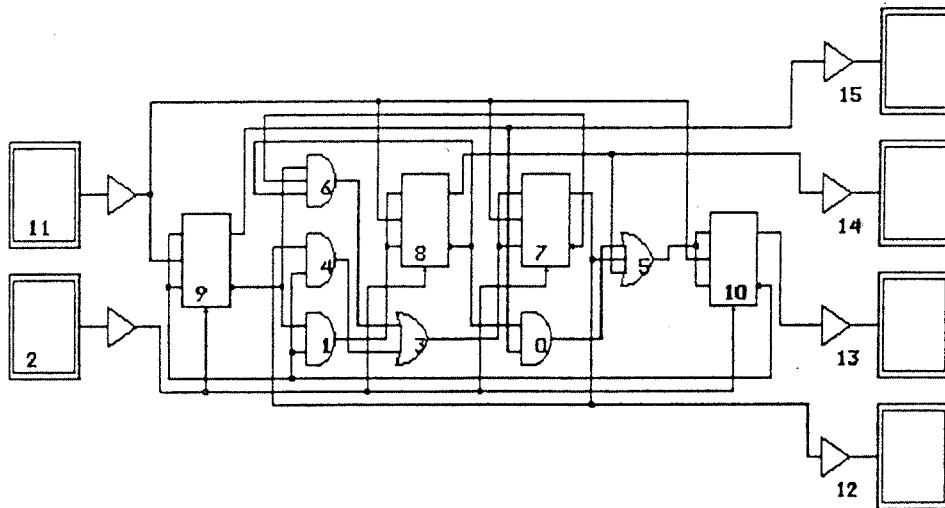


Slika 20: Določitev spojišč za vertikalne povezave

Nato določimo še spojišča za vse vertikalne povezave. Spojišča postavimo ravno tako, kot za horizontalne povezave, ki so direktno priključene na vodoravne (slika 20), spojišče postavimo, če se vertikalna povezava dotika ustrezne vodoravne nekje v sredini.

3.ZAKLJUČEK

Na sliki 22 je izrisan primer sheme vezja, ki jo generira program SCHEMGEN. To vezje predstavlja številnik po modulu 9. V datoteki opisa vezja (slika 21) se nahaja vrstica MODULE, v kateri so opisani vsi izhodi in vhodi celotnega vezja ter ime vezja. Med vrsticama BEGIN in END pa se nahajajo opisi celic vezja: vozlišča, kamor so vhodi in izhodi celic priključeni in tipi celic. Številke ob celicah na sliki pomenijo zaporedno št. celice v vhodni datoteki (le-te tečejo od 0 naprej). Program je posebno uporaben za kombinacijska vezja. Problem pa nastane pri vezjih, ki imajo povratne povezave, ker razporeditev celic vezja ni več enolično določena. Razporeditev celic je močno odvisna od vrstnega reda opisov celic v vhodni datoteki. Primer je ravno vezje na sliki 22. Ker je to



Slika 22: Shema vezja, ki jo je program generiral (opis na sliki 21)

```

MODULE QA QB QC QD .STEVNIK9 RES CLK;
BEGIN 16 .AA025 4 5;
9 .AA025 6 8;
RES1 .IB01C5 RES;
22 .OR025 21 10;
10 .AA025 1 8;
15 .OR035 16 1 3;
21 .AA035 6 2 4;
1 2 .JK115 22 22 CLK1 RES1;
3 4 .JK115 9 9 CLK1 RES1;
5 6 .JK115 8 8 CLK1 RES1;
7 8 .JK115 15 15 CLK1 RES1;
CLK1 .IB01C5 CLK;
QD .OB03C5 1;
QA .OB03C5 7;
QC .OB03C5 3;
QB .OB03C5 5;
END;

```

Slika 21: Tekstovni opis vezja na sliki 22

števniki, bi bilo idealno, če bi bili flip-flopi na shemi postavljeni tako, kot si sledijo biti po svoji pomembnosti. Vidimo, da to ne velja. Ne obstaja splošen postopek, po katerem bi lahko računalnik vedno našel tako razporeditev celic, kot si jo želimo mi. Razporeditev je namreč močno odvisna tudi od funkcije vezja. Ta problem bi bilo možno rešiti tako, da bi v programu omogočili, da nekatere celice vezja razmestimo ročno.

LITERATURA:

/1/ Yehuda Shiran: A Deterministic Approach to Netlist Display, COMP EURO 89, Proceedings of the 3rd Annual European Computer Conference, Hamburg, May 8-12, 1989, str. 5-13 do 5-34.

Stanislav Gruden, dipl. ing.
 Marijan Leban, dipl. ing.
 prof. dr. Baldomir Zajc, dipl. ing.
 Fakulteta za elektrotehniko in računalništvo
 Tržaška 25, 61000 Ljubljana

Prispelo: 08.02.91 Sprejeto: 01.03.91