

Denis Tomašević
Baldomir Zajc

Fakulteta za elektrotehniko in računalništvo
Tražaka 25
Ljubljana

"SPIDER" : Učinkovit program za avtomatsko
povezovanje celic integriranega vezja

"SPIDER": An Efficient Program for
Channel Routing

Povzetek:

Enega ključnih problemov pri načrtovanju geometrije VLSI predstavlja povezovanje celic (*routing*). Predstavljen bo program, ki temelji na algoritmu Yoshimura (*channel routing*) algoritem za povezovanje celic. Program je napisan v programskem jeziku Turbo Pascal version 5.0 in testiran na IBM PC. Omogoča tudi prikazovanje opravljenega dela na zaslonu in tiskalniku.

Abstract:

One of the key problems in VLSI layout design is connecting the cells - routing. The represented program is based on channel routing developed by Yoshimura. The program was coded in Turbo Pascal ver.5.0 and tested on IBM PC. It enables the user to see the results of the program on the screen or printer.

1. Uvod

Pri izdelavi integriranih vezij težimo k čim večji integraciji oziroma čim manjši površini celotnega vezja. Geometrijsko podobo vezja (*layout*) določamo v dveh korakih: V prvem koraku določamo razpored celic (*placement*),

v drugem pa razpored povezav (*routing*). Razmeščanje odločilno vpliva na dolžino povezav in obliko vezja. Povezovanje določa končno dimenzijo vezja in dolžino povezav.

že od leta 1960 dalje, ko se je pokazala potreba po večji stopnji integracije, lahko v literaturi zasledimo različne pristope k razmeščanju in povezovanju celic. Tako je bil prvotni problem povezovanja podan z dvema vrsticama vozlišč, povezave pa so bile narejene v prostoru (*channel*) med vrsticama v dveh plasteh. Kasneje se je pojavil tudi problem povezovanja v skatli, ki jo omejujejo štiri vrstice vozlišč prostorsko razmeščenih celic. Iskale so se tudi možnosti uporabe tri in več plastnih povezav. Kljub temu ostaja osnovni problem še vedno pomemben.

Obstajata dva bistveno različna pristopa k reševanju problema povezovanja dveh vrstic z dvoplastnimi povezavami: Pri prvem ležijo vsi horizontalni segmenti v eni plasti, vsi vertikalni segmenti pa v drugi. Drugi pristop dovoljuje vsakem segmentu, da leži tako v eni kot drugi plasti, ali pa v obeh. Karakteristično za prvi pristop je dejstvo, da dva sorodna segmenta (vertikalna oz. horizontalna) ne moreta ležati eden čez drugega, ker bi nastal električni stik med povezavama. Pri upoštevanju tega dejstva dobimo iz vertikalnih segmentov vertikalni usmerjeni graf. Vsako vozlišče tega grafa predstavlja eno povezavo, kazalec od vozlišča *a* proti vozlišču *b* pa pomeni, da eden od vhodov povezave *a* leži nad enim od vhodov povezave *b*. Velja: če obstaja direktna pot od vozlišča *a* do vozlišča *b*, mora

povezava a ležati nad povezavo b. Glede na to kako se zgoraj omenjeno dejstvo uporablja na horizontalnih segmentih, ločimo dve metodi : *Maze algoritmi* postavijo čez kanal najprej koordinatno mrežo : stolpce mreže ob vhodih celic in toliko vrstic mreže kot je največja gostota kanala (*maxdensity*). Vsako presežno točko te mreže predstavijo kot eno komponento matrike .Potem v to matriko vpisujejo povezave, upoštevajoč vertikalni usmerjeni graf. če kakšna povezava ni izvedljiva, vrinejo novo vrstico.

Pri metodi *channel routing* na osnovi horizontalnih segmentov kanal razdelijo v cone. Cona predstavlja področje kanala v katerem so vsi stolpci vsebovani v enem stolpcu. Bistvo te metode je združevanje vozlišč v vertikalnem usmerjenem grafu. Ko je združevanje končano, priredimo vsakem preostalem vozlišču en trak kanala. S to metodo v bistvu dosežemo boljše rezultate v krajšem času, problemi pa nastanejo ,če vertikalni usmerjeni graf vsebuje sklenjene poti (*cyclic conflict*). Takrat moramo tako sklenjeno pot pretrgati, problematični segment realiziramo posebaj (npr. z *Maze*). Minimalna širina, ki jo lahko dosežemo z tema metodama je podana z gostoto kanala (*maxdensity*).

Pri drugem pristopu dosežemo v večini primerov manjšo širino kanala od njegove gostote. Slaba stran tega pristopa so parazitne kapacitivnosti zaradi prekrivanja povezav, razpadanje povezave na veliko število segmentov (bolj zapletene maske) ter tehnološko zahtevnejši postopek. Zato se ta pristop uporablja le takrat, ko je glavni cilj minimalna površina vezja .

2. Opis programa

Predstavljeni program "SPIDER" je osnovni program za avtomatsko povezovanje dveh vrstic z dvoplastnimi povezavami. Temelji na metodi *channel routing* Yoshimura [1], napisan v programskem jeziku Turbo Pascal ver. 5.0 in testiran na IBM PC na FER. Razdeljen je v tri dele: V prvem delu program prebere podatke iz vhodne datoteke, opravi lomljenje povezav, ter ustvari vertikalni usmerjeni graf in cone. Drugi del programa izbira pare povezav in jih združuje do končnega rezultata. V tretjem delu lahko rezultate opazujemo na zaslonu ali tiskalniku.

2.1. Vhodni del

Program najprej opravi vnos podatkov. Vhode bere po parih - zgornjega (označi kot +x) in spodnjega (-x) ob stolpcu x. Istočasno ustvarja vertikalni usmerjeni graf, pri čem vsako povezavo razdeli na segmente - en segment omejujeta dva sosednja vhoda povezave. Za vsako vozlišče (segment) izračuna najdaljšo pot do začetnega oz. končnega vozlišča in potem poizkuša združiti sosednje segmente iste povezave tako, da najdaljša pot ostane nespremenjena. V idealnem primeru bi bila vsaka povezava predstavljena z enim samim horizontalnim segmentom. Po opravljenem združevanju, program določi še cone kanala in njegovo gostoto.

2.2. Glavni del

Bistvo tega dela je združevanje vozlišč vertikalnega

usmerjenega grafa. Potencialne pare za združitev program izbira na osnovi con in nato izloči tiste, ki niso dovoljeni z vertikalnim usmerjenim grafom (da ne obstaja direktna pot med vozliščema). Z uporabo uteznostnih funkcij [1] izloči en par, združi vozlišča in ponovno izračuna najdaljše poti za vsako vozlišče. Slednji korak se v programu "SPIDER" ne izvrši na klasičen način (z rekurzijo), ampak posebej v eno smer (začetno vozlišče -> končno vozlišče), potem pa še v obratno smer. Ta način se izkaže pri kompliciranih primerih kot hitrejši, poleg tega pa nam omogoči enostavnejše združevanje vozlišč. Izkušnje so pokazale, da najboljše rezultate dosežemo , če se združevanje začne pri coni z največjo gostoto. Z dodatnim parametrom uporabnik lahko izbira smer združevanja.

2.3. Grafika

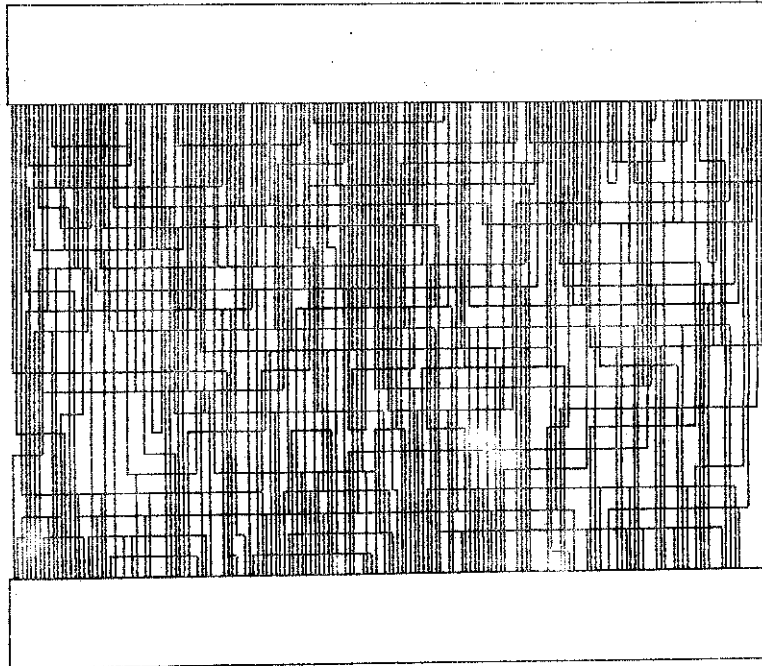
Tretji del programa je namenjen vizuelni predstavitvi opravljenega dela. Program sam izbere merilo v x in y smeri tako, da se na začetku celoten kanal prikaže na zaslonu. Včasih to ni najbolj pregledno, zato program omogoča 32-kratnega povečavo . V vsakem trenutku lahko sliko , ki je na zaslonu prenesmo tudi na tiskalnik. Primer podan v naslednjem poglavju je narisan prav na ta način.

3. Primeri

Tabela 1. podaja rezultate za primere , ki jih najdemo v literaturi [1]. Primer diff je znani Deutch-ov problem. Ob tem naj poudarimo, da je program "SPIDER" še vedno v fazi

razvoja. Z izbiro novih utežnostnih funkcij lahko pričakujemo še boljše rezultate.

primer	maxdensity	št. trakov
3a	15	15*
3b	17	17*
3c	18	18*
4b	20	20*
diff	19	22



Slika 1. Rešitev Deutchovega problema

4. Sklep

Predstavljeni program je del programskega paketa, ki se razvija na FER. V večini primerov doseže optimalne

rezultate, manjša odstopanja se pojavijo le pri izredno zapletenih primerih. Pričakujemo da bomo z razvojem novih utežnostnih funkcij dosegli še boljše rezultate.

5. Literatura

- 1.) T.Yoshimura and E.S.Kuh, "Efficient Algorithms For Channel Routing", IEEE Trans CAD, vol.cad1 no.1,pp 25-35, January 1982
 - 2.) D.N.Deutch, "A "Dogleg" Channel Router ", Proc 13th Design Automation Conf, pp 425-433, 1976
 - 3.) Y.L.Lin et al, "SILK:A Simulated Evolution Router", IEEE Trans CAD, vol.8 no.10,pp 1108-1114, october 1989
 - 4.) F.J.Kurdahi and A.C.Parker, "Techniques for Area Estimation of VLSI Layout", IEEE Trans CAD, vol.8 No.1,January 1989
 - 5.) J.M.Kleinhanz, "Efficient Algorithms for Two- and Three-Layer Channel Routing", Proc Comp EURO 87 VLSI and Computers, pp 629-632, 1987
 - 6.) J.Criado et al, "CANAL : An Efficient Program For Channel Routing", Proc Melecon 89, pp 320-323, 1989
-
