XXXIV JUGOSLOVENSKA KONFERENCIJA ETAN-a, ZAGREB, 4-8. JUNA 1990.

Roman Kužnar
Baldomir Zajc
Andrej Žemva

Fakulteta za elektrotehniko
Tržaška 25
Ljubljana

AVTOMATSKO RAZMEŠČANJE STANDARDNIH CELIC INTEGRIRANEGA VEZJA NA OSEBNEM RACUNALNIKU IBM PC

AUTOMATIC PLACEMENT OF STANDARD CELLS FOR INTEGRATED CIRCUIT ON IBM PC PERSONAL COMPUTER

POVZETEK: Pričujoči članek opisuje iskanje optimalne razporeditve celic znotraj bloka digitalnega integriranega vezja. Osnova optimizacijskega postopka je minimizacija vsote dolžin povezav med celicami. Celoten postopek je v celoti avtomatiziran. Doseženi so zelo dobri rezultati.

ABSTRACT: The searching of optimal cell placement inside the blocks of digital integrated circuit is presented. The basis of the optimisation procedure is minimisation the total length connection between the cells. The whole process is computerized and very good results are achieved.

1. UVOD

Pri projektiranju digitalnih integriranih vezij zelo visoke integracije (VLSI) predstavlja razmestitev celic vezja enega pomembnejših korakov načrtovanja. Problem je spričo velikega števila komponent zelo obsežen in ga razdelimo na več manjših korakov. Današnja topologija VLSI integriranega vezja temelji na hierarhični zasnovi čipa. Celica je najmanjša logična enota, ki vsebuje do 50 vrat. Blok je osnovna enota logičnega načrtovanja vezja in vsebuje od 500 do 1000 vrat. Integrirano vezje je lahko sestavljeno iz več deset blokov. Dimenzioniranje, razporeditev ter medsebojno povezovanje blokov so postopki, ki so znani pod imenom "Floorplannning" in "Block placement and routing". Cilj teh postopkov je dobiti obliko čipa, ki je čim bolj kvadratna ter ima čim manjšo površino.

V tem članku je opisana razporeditev celic znotraj bloka integriranega vezja. Celice so razvrščene v sodo število vrstic ter so med seboj povezane s povezavami, ki potekajo v povezovalnih kanalih. Kanali ležijo med vsako sodo in liho vrstico. Končni cilj je dobiti takšno razporeditev celic, ki minimizira vsoto dolžin povezav med celicami. Smatramo, da je to

zapleten kombinatorični problem, ki ga ni mogoče analitično rešiti. V praksi se poslužujemo algoritmov, ki temeljijo na hevrističnih metodah.

Program, ki omogoča avtomatsko razmestitev celic, temelji na AMI-jevi knjižnici standardnih celic. Napisan je v programskem jeziku Turbo Pascal 5.0 in služi kot dopolnilo programskemu paketu SuperSCEPTRE [4].

2. ALGORITEM ZA AVTOMATSKO RAZMESTITEV CELIC

Celoten algoritem za razmestitev celic je smiselno razdeljen v dve fazi:

1. faza: generiranje m-začetnih razmestitev (m= število celic) in izbira n-najboljših začetnih razmestitev.
2. faza: iterativno izboljševanje n-začetnih razmestitev ter izbira tiste razmestitve, ki rezultira v minimalni dolžini vseh povezav.

V prvi fazi je generirano toliko razporeditev celic, kolikor je celic vezja. Vsaka celica je enkrat izbrana kot začetna celica pri izračunu funkcije vhodno izhodne povezanosti. Število n-začetnih razmestitev, ki jih iterativno izboljšujemo je vnaprej določeno. Pri majhnem številu celic je n sorazmeren številu celic, pri večjem številu celic pa je n konstanta. Mejno število celic je 100.

Končna razvrstitev je močno odvisna od začetne postavitve. Izbira celice, ki je izhodiščna pri izračunu funkcije vhodno izhodne povezanosti, vpliva na začetno postavitev. Izbiro celice, ki daje najboljši izhodiščni položaj, ne moremo vnaprej predvideti. Prav tako ni nujno, da najboljša začetna postavitev vodi k najboljšemu končnemu rezultatu, vendar obstaja velika verjetnost, da ena od začetnih postavitev vodi v globalni minimum celotne dolžine povezav. Tudi če iterativno izboljšujemo vse m-začetne postavitve se izkaže, da v večini primerov rezultati niso boljši kot pri uporabljenemu algoritmu.

2.1 GENERIRANJE ZAČETNE RAZMESTITVE

Začetna razmestitev celic je sestavljena iz dveh medsebojno neodvisnih korakov. V prvem koraku določimo vrstni red razvrščanja celic v vrstice glede na funkcijo vhodno izhodne povezanosti. V drugem koraku razvrščamo celice v proste vrstice tako, da je dolžina povezav do že postavljenih celic minimalna.

A. Definicija funkcije vhodno izhodne povezanosti

Utežna mera celice, ki jo izberemo za začetno je ena, ostale celice imajo utežno mero nič. Vsaki neizbrani celici povečamo utežno mero za ena pri vsaki povezavi s celicami, ki so že izbrane. Izmed celic, ki imajo največjo utežno mero, izberemo takšne, ki imajo najmanj povezav do še neizbranih celic. V kolikor je takšnih celic več, izberemo tisto, ki ima največjo

širino. Postopek ponavljamo toliko časa, dokler niso izbrane vse celice.

B. Začetna razmestitev celice

Celice razvrščamo v vrstice po principu rasti kristala. Celico razvrstimo v tisto vrstico, ki ima najmanjšo dolžino vseh povezav do že postavljenih celic. V vrstici je celica razmeščena na prvo prosto mesto. Pri tem moramo upoštevati, da ne presežemo dopustne dolžine vrstice. Celico, ki ima daljšo širino, kot je prostora v izbrani celici, postavimo v drugo vrstico.

2.2 ITERATIVNI POSTOPEK

Uporabili smo algoritem, ki hkrati zamenja dve celici. Lega celic, ki ju želimo zamenjati je poljubna. Glavni kriterij za zamenjavo dveh celic je zmanjšanje celotne dolžine povezav.

1. korak: Izbira primarne celice. Izberemo celico z največjo povprečno dolžino povezav.

2.korak: Za primarno celico izračunamo novo optimalno lego [1]. Vse celice, ki ležijo v okolici optimalne lege označimo za sekundarne.

3.korak: Med sekundarnimi celicami poiščemo celice, katere pri zamenjavi s primarno celico zmanjšajo celotno dolžino povezav. Razporedimo jih po padajočem zaporedju glede na učinek zmanjševanja dolžine povezav. Pri tem naredimo aproksimacijo, da imajo sekundarne celice enako širino kot primarna celica.

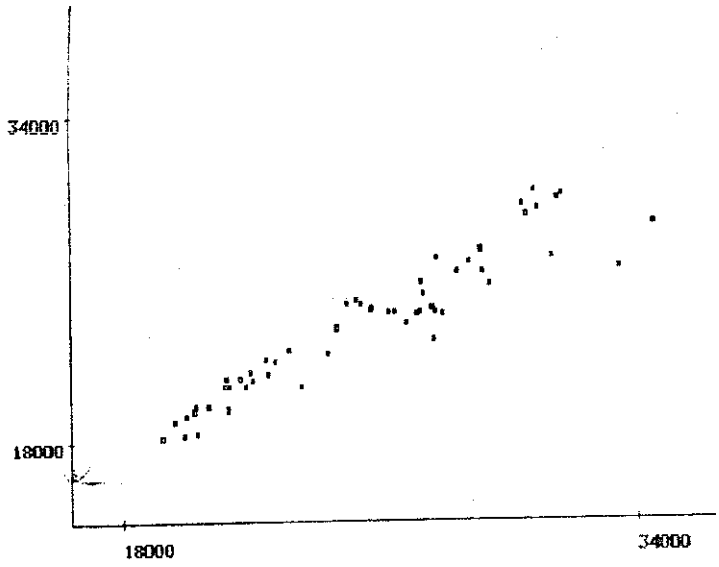
4.korak: V enem ciklu primarna ter sekundarna celica zamenjata legi. V primeru, da imata celici različni širini, izvedemo popravek za vse celice, ki ležijo desno od celic, ki zamenjata legi. Upoštevamo tisto končno zamenjavo, ki daje najmanjšo vsoto dolžin povezav. V primeru, da nobena zamenjava ne prispeva k zmanjšanju dolžine povezav, upoštevamo prvotno razporeditev celic.

3. REZULTATI

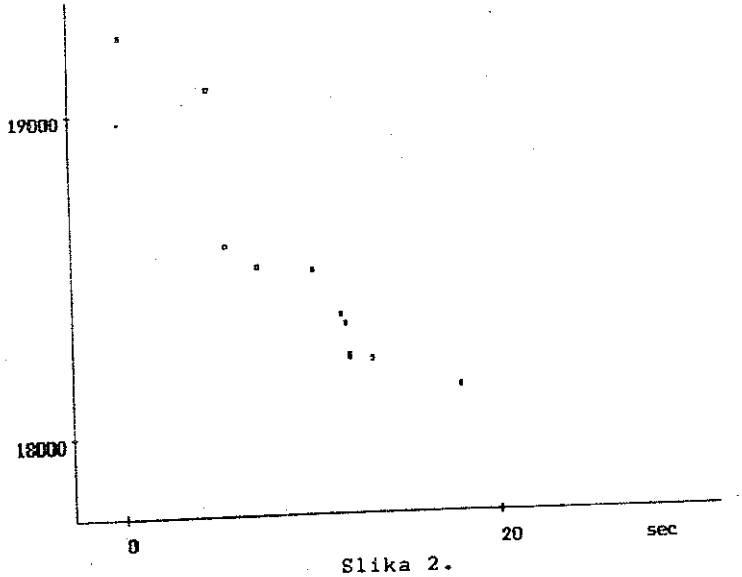
Uporabili smo takšna testna vezja, da smo lahko dobili rezultate, ki smo jih primerjali z rezultati objavljenimi v literaturi. Izkazalo se je, da so rezultati primerljivi z dosedanjimi objavljenimi rezultati.

Na sliki 1. je prikazana odvisnost celotne dolžine povezav začetne razmestitve celic po izvedbi iteracijskega postopka.

Časovno odvisnost iterativnega postopka ene razmestitve kaže slika 2.



Slika 1.



Slika 2.

4. ZAKLJUČEK

Program je namenjen razvrščanju standardnih celic znotraj

bloka integriranega vezja srednje stopnje integracije (do 500 celic), zato smo uporabili metodo, ki temelji na ponavljanju enostavnih korakov. Algoritem je zasnovan tako, da se za to stopnjo integracije približamo optimalni rešitvi v zelo kratkem času. Sam program je že v fazi razvoja in bo skupaj s programom za avtomatsko povezovanje celic omogočil programsko načrtovanje layouta s programskim paketom SuperSCEPTRE.

5. LITERATURA

1.) H. Terai, et al, "Automatic Placement and Routing program for Logic VLSI Design Based on Hierarchical Layout Method", Proc. IEEE Int. Conf. Circuit Computers, 1982, pp. 415-418.

2.) S. Goto, "An Efficient Algorithm for the Two-Dimensional Placement Problem in Electrical Circuit Layout", IEEE Trans. on Circuits and Systems, Vol.CAS-28, No.1.,pp 12-18.

3.) M.Hanan, P.K.Wolff, and B.J.Anguli, "Some experimental results on placement techniques," in Proc 13th Design Automation Conf.,(San Francisco, CA), pp. 214-224, 1976.

4.) V.Jurkas, B. Zajc, "Primer načrtovanja integriranih vezij na osebnih računalnikih IBM PC, XXXIII Jug. konf.ETAN, Novi Sad, 1989, pp. 137-142.