

# Optimizacija digitalnih gradnikov na nivoju tranzistorjev

Janez Puhan, Iztok Fajfar, Tadej Tuma, Árpád Bűrmen

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija

E-pošta: janez.puhan@fe.uni-lj.si

**Povzetek.** Izdelovalci integriranih vezij za svoje procese podajajo knjižnice osnovnih digitalnih gradnikov, ki se uporabljajo pri načrtovanju digitalnih integriranih vezij. Izvedbe posameznih gradnikov na tranzistorskem nivoju so s tem določene in se med načrtovanjem vezja ne spreminjajo več. Gradniki, kot npr. medpomnilniki, logična vrata, seštevalniki, flip-flopi itd., se uporabljajo brez sprememb, kjerkoli jih načrtovalec potrebuje. Zaradi tega posamezen gradnik ni povsem prilagojen na specifično okolje v vezju, kjer se nahaja. To odpira prostor optimizaciji gradnika točno na zahteve v katerih deluje. članek opisuje primer optimizacije digitalnega gradnika na tranzistorskem nivoju. Zaradi šumnosti kriterijske funkcije je bila uporabljena robustna globalna optimizacijska metoda. Rezultati pokažejo znatno izboljšanje želenih lastnosti glede na splošni gradnik izdelovalca integriranih vezij.

**Ključne besede:** načrtovanje digitalnih integriranih vezij, splošni osnovni gradniki, sinteza digitalnih vezij, optimizacija na tranzistorskem nivoju

## 1 UVOD

V nalogi je obravnavan postopek 3D rekonstrukcije objektov na podlagi večih zajetih slik. Za rekonstrukcijo je uporabljena DLT (angl. *Direct linear transform*) metoda. Temelji na rekonstrukciji 3D oblika iz slik, ki so zajete pri znanih kotih zasuka objekta. Pri postopku rekonstrukcije ima zelo veliko vlogo sistem s katerim zajemamo sliko. Bolje kot je zastavljen, bolj kontrastne slike dobimo. Dobro zajete slike nam olajšajo delo pri kasnejši obdelavi.

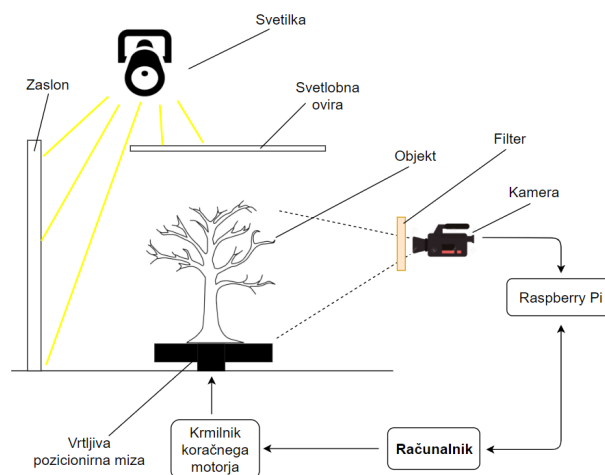
## 2 TEORETIČNO OZADJE

kalibracija in fbp algoritem (pseudo ali flowchart), poravnava

## 3 ZAJEM SLIK

Sistem s katerim smo izvajali rekonstrukcijo je blokovno prikazan na sliki 1. Objekt, ki ga želimo skonstruirati postavimo na vrtljivo pozicionirno mizo. Iz zgornje strani navzdol svetimo s homogeno, difuzno osvetlitvijo. Objekta direktno ne osvetljujemo, saj je namen osvetlitve predvsem osvetlitev ozadja z enakomerno intenziteto. Cilj je doseči čimvečjo razliko v intenziteti med opazovanim objektom in zaslonom. Za ta namen je med objekt in svetilko postavljena svetlobna ovira. Slike smo zajemali z Raspberry Pi kamero v osi na objekt in na osvetljeno ozadje. Pred kamero je dodan filter za namen filtriranja okoliške svetlobe.

Prejet 19. oktober, 2010  
Odobren 4. februar, 2011



Slika 1:: Semioperacijska shema sistema.

Celoten proces zajemanja krmilimo z računalnikom. Ta je žično povezan na krmilnik s katerim upravljamo rotirajočo pozicionirno mizo. Prav tako je računalnik brezžično povezan na Raspberry Pi. Ta zajete slike pošilja na računalnik. Objekt rekonstruiramo tako, da ga postavimo na predvideno mesto. Nato lahko pričnemo z zajemom slik. Najprej inicializiramo kamero na začetne vrednosti. Nastavimo ISO na 200, svetlost na 45 in resolucijo na 900x1000. Objekt obračamo po načrtanih zasukih. Po vsakem izvedenem zasuku posnamemo eno sliko, ki se nato brezžično prenese iz Raspberry Pi-ja na računalnik. Postopek ponavljamo dokler objekta ne poslikamo v rangu celotnega kroga. Zajete slike ob znanih kotih zasuka se kasneje uporabijo za 3d rekonstrukcijo.

V tabeli 1 so zapisane vse glavne komponente, ki so zajete v postavljenem sistemu. Manjše komponente kot so vijaki in nosilci v tabeli niso zajeti.

Tabela 1:: Komponente sistema.

| Komponenta        | Podrobnosti                  |
|-------------------|------------------------------|
| Raspberry Pi      | Raspberry Pi 3 Model B       |
| Svetilka          | Halogenska svetilka (12V)    |
| Zaslon            | Bela ravna plošča            |
| Krmilnik          | Fischertechnik stage control |
| Kamera            | RP Cam V2-8 MP,1080p         |
| Svetlobni filter  | Ozkopasovni filter           |
| Svetlobna ovira   | Kartonasta zavesa            |
| Pozicionirna miza | Fischertechnik rotary stage  |

Primer zajete slike obravnavanega objekta je prikazan na sliki 2. S slike lahko vidimo dober kontrast objekta in precejšnjo razliko v barvi in svetlosti objekta proti ozadju.



Slika 2:: Zajeta slika.

## 4 REZULTATI

### 4.1 Vpliv kompleksnosti in kontrasta

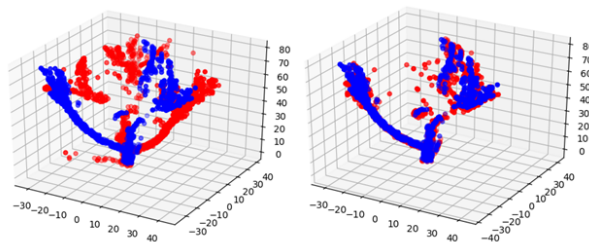
- primerjava kompleksnosti modela - Primerjaj slab in dober kontrast slike (primer slusalk na 30 slikah)

### 4.2 Število zajetih slik

- Ali res boljša z 90 proti 30 slikami (drevešček)

### 4.3 Poravnava oblakov točk

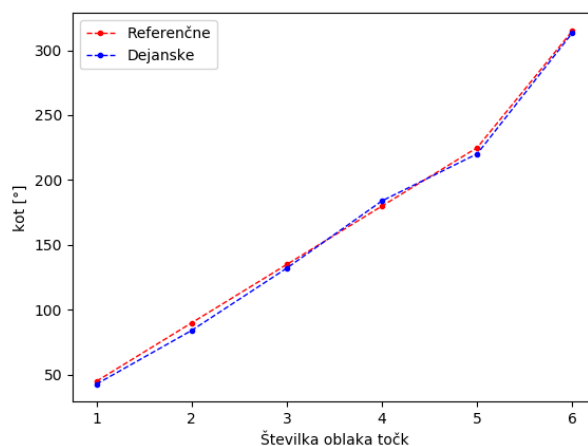
Za dodaten eksperiment smo izvedli medsebojno poravnavo oblakov točk. Referenčni oblak točk smo zajeli z 90 slikami in si zapolnili referenčno lego objekta na pozicionirni mizi. Nato smo izvedli še šest zajemov po 30 slik. Ob vsakem zajemu smo skeniran objekt zasukali za predpisan kot glede na referenčno lego. Tako smo zajeli oblake točk zasukane za 45, 90, 135, 180, 225 in 315 stopinj. Na sliki sta zaradi preglednosti prikazana le referenčni in zasukan oblak. Na levi strani sta oblaka prikazana pred poravnavo in na desni po poravnavi.



Slika 3:: a) neporavnana, b) poravnana.

Po vsaki izvedeni poravnavi smo iz transformacijske matrike izsluščili kot..

Na sliki 4 so z rdečo prikazani referenčni koti, z modro pa dejanski koti, ki jih dobimo s pomočjo poravnave.



Slika 4:: Razlika med referenčnimi in dejanskimi koti.

### 4.4 Vzvratno inženirstvo

- mesh v blenderju in 3d print (reverse engineering)

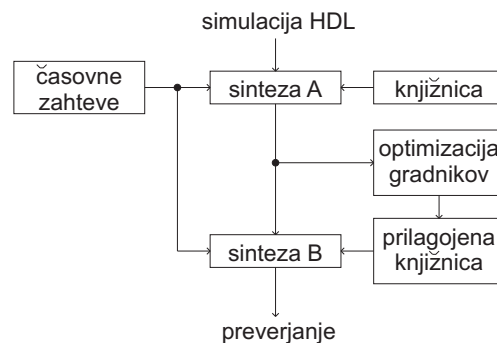
Pri načrtovanju digitalnega vezja načrtovalci uporabljajo osnovne gradnike ali bloke [1] kot so medpomnilniki, logična vrata, seštevalniki, zatiči, flip-flopi itd. Knjižnico z digitalnimi gradniki za vsak proces izdelave integriranih vezij priskrbi izdelovalec integriranih vezij. Navadno je podanih več različic posameznega gradnika, skupaj z opisom karakteristik posamezne različice. Med postopkom načrtovanja digitalnega integriranega vezja načrtovalec simulacij [2], [3], [4], [5] na tranzistorskem nivoju ne izvaja. Za izračun odziva vezja se uporabljajo opisi gradnikov na višjem nivoju [6], [7].

Posamezne različice nekega gradnika imajo v večini primerov enako topologijo. Razlikujejo se le v dimenzijah tranzistorjev, navadno v širinah in dolžinah kanalov. So rezultat prilagoditve gradnika na različne pogoje delovanja. V članku je opisana optimizacija gradnika iz knjižnice izdelovalca integriranih vezij. Zanima nas, ali je možno s pomočjo optimizacijskega postopka iz standardnega gradnika iztisniti še boljše lastnosti. S tem bi dobili tudi odgovor na vprašanje, ali je optimizacija gradnikov na tranzistorskem nivoju smiselna, oziroma ali bi kazalo v prihodnosti z avtomatizacijo optimizacijskega postopka gradnikov na tranzistorskem nivoju generirati poljubne gradnike prilagojene na točno določene razmere v vezju.

Proces načrtovanja digitalnega vezja ASIC (Application Specific Integrated Circuit) se začne z opisom specifikacij vezja na nivoju RTL (Register Transfer Level). Nato načrtovalec vezje opisano z opisnim jezikom HDL (Hardware Description Language) simulira in popravlja, dokler odziv vezja ni zadovoljiv glede na podane zahteve. Končen opis vezja v jeziku HDL se reducira na logični nivo. Vezje na logičnem nivoju se na nivoju osnovnih gradnikov optimizira, dokler ustreznost vezja glede na zahteve v časovnem prostoru ni dosežena. Redukcijo na logični nivo in optimizacijo na nivoju gradnikov imenujemo sinteza. Sintezo načrtovalec izvede s pomočjo sintezacijskega orodja, ki pri tem izbira med osnovnimi gradniki iz knjižnice proizvajalca integriranih vezij. Delovanje končnega vezja načrtovalec preveri pred in po razporeditvi (ang. layout) tranzistorjev na rezini. V kolikor med preverjanjem ugotovi napake, ali nezadovoljivo delovanje vezja, mora proces sinteze ponoviti s strožjimi zahtevami.

Idejo uvedbe optimizacijskega postopka na nivoju tranzistorjev v tok sinteze vezja prikazuje slika 5. Rezultat sinteze A je opis vezja na nivoju osnovnih gradnikov skupaj s časovnimi zahtevami za posamezen gradnik. Iz knjižnice so izbrane ustrezne različice posameznih gradnikov. Sledi predlagana optimizacija na nivoju tranzistorjev. Optimizacija s spreminjanjem velikosti tranzistorjev prilagodi posamezen gradnik točno na njegove časovne zahteve. Nekateri izmed izbranih gradnikov v koraku A komaj zadovoljujejo podane časovne zahteve, medtem ko imajo drugi veliko rezerve. Gradnike, ki komaj izpolnjujejo postavljene zahteve, bi lahko optimizirali na hitrost, gradnike z rezervo pa na porabo, brez

da bi pri tem poslabšali lastnosti vezja. Rezultat optimizacije na nivoju tranzistorjev so razmeram prilagojeni gradniki, iz katerih bi lahko sestavili novo knjižnico. Sinteza B bi prvotno knjižnico gradnikov zamenjala s prilagojeno. Uspešna sinteza B bi tudi potrdila pravilno delovanje vezja zgrajenega iz prilagojenih gradnikov.



Slika 5:: Uvedba optimizacijskega postopka na nivoju tranzistorjev v tok sinteze

## 5 KRITERIJSKA FUNKCIJA

Kriterijska funkcija (KF) je mera, ki meri kvaliteto predlaganega vezja. Bolj kot vezje ustreza postavljenim zahtevam, nižja je vrednost KF. Z določitvijo KF postane odločitev, katero vezje je boljše, nedvoumna [10].

Lastnosti predlaganega vezja so določene z vrednotenjem rezultatov ene ali več tranzientnih analiz. V našem primeru so zanimive naslednje lastnosti: površina na rezini (določena z dimenzijami tranzistorjev), obnašanje v časovnem prostoru (strmine, zakasnitve, itd.) in poraba.

K končni vrednosti KF prispeva vsaka izmerjena lastnost  $x_i$  svoj delež. Dokler želena vrednost  $g_i$  lastnosti ni dosežena je prispevek  $c_i(x_i)$  v KF proporcionalen oddaljenosti od želene vrednosti. Ko lastnost doseže, ali celo preseže želeno vrednost, postane njen delež negativen. Definicija prispevka lastnosti v KF je podana v enačbi (1).

$$c_i(x_i) = \begin{cases} \frac{t_i}{g_i}(x_i - g_i) & x_i \leq g_i \\ \frac{p_i}{g_i}(x_i - g_i) & x_i > g_i \end{cases} \quad (1)$$

Ker morajo biti vse zanimive lastnosti  $\mathbf{x} = [x_1, x_2, \dots]^T$  čimmanjše, potrebujemo le en način vrednotenja prispevka lastnosti. Enačba (1) zadošča. Končna vrednost KF je vsota vseh prispevkov podana v enačbi (2).

$$c(\mathbf{x}) = \sum c_i(x_i) \quad (2)$$

KF v enačbi (2) ima globalni minimum dolčen z enačbo (3), ki ustreza optimalnim parametrom vezja  $\mathbf{w}_{\text{opt}}$ . Z različnimi želenimi vrednostmi  $g_i$ , trgovalnimi ( $t_i$ ) in kazenskimi ( $p_i$ ) utežmi optimizacijski postopek privede do več globalnih minimumov KF. Torej do nabora vezij z različno uravnoteženimi lastnostmi.

$$c(\mathbf{x}(\mathbf{w}_{\text{opt}})) \leq c(\mathbf{x}(\mathbf{w})) \quad (3)$$

Nedoseganje posamezne želene vrednosti je lahko po definiciji KF izničeno s presežanjem drugih. Da se to ne zgodi in so želene vrednosti vseh lastnosti dosežene, mora za vsak par  $i \neq j$  veljati  $t_i \ll p_j$ . Tako bo prispevek lastnosti, ki ne dosega želene vrednosti, mnogo večji od absolutne vrednosti vsote negativnih prispevkov vseh drugih lastnosti, ki presežajo svoje želene vrednosti.

Predlaganim vezjem, za katera tranzientna analiza ne uspe, vektorja lastnosti  $\mathbf{x}$  ni mogoče določiti. V primeru, ko posamezna lastnost  $x_i$  ni znana, je njen prispevek v KF  $c_i(x_i)$  postavljen na neko veliko vrednost  $c_{\max,i}$ . Predlagano vezje, za katerega tranzientna analiza ne uspe, ima tako visoko vrednost KF in zato predstavlja slab poskus. Enako velja za predlagana vezja, za katera tranzientna analiza sicer uspe, vendar je odziv neuporaben in ene ali več lastnosti ni mogoče določiti (npr. v odzivu ni pričakovane naraščajoče fronte, zaradi česar ni mogoče izmeriti njene strmine).

Predlagano vezje z odzivom, ki ne izraža pričakovane funkcije vezja, je dodatno kaznovano s pomožnimi lastnostmi. Pomožne lastnosti, oziroma meritve preverjajo odziv v časovnih trenutkih, kjer je stanje digitalnega vezja določeno. Imajo velike kazenske uteži, kar vodi v visoke prispevke v KF za vezja z nepričakovanim odzivom. Pomožne meritve k KF ne prispevajo (trgovalne uteži so nič), ko ima predlagano vezje pričakovan odziv. S tem pomožne meritve silijo k pravilnemu odzivu vezja in tako pomagajo optimizacijskemu procesu.

## 6 TESTNO VEZJE

Kot testni gradnik je bil uporabljen D flip-flop prikazan na sliki ???. Vzet je iz knjižnice osnovnih digitalnih gradnikov podanih s strani proizvajalca integriranih vezij za 350nm CMOS proces. V kolikor optimizacijski postopek na tranzistorskem nivoju uspe izboljšati testni gradnik, je podobne izboljšave pričakovati tudi na drugih proizvajalčevih gradnikih.

Testni gradnik je vstavljen v testno vezje, ki zagotavlja vhodne signale, obremenitve izhodov in napajanje gradnika. Strmine vhodnih signalov, bremenske kapacitivnosti in napajalna napetost se spreminjajo od kota do kota.

Na sliki ?? so prikazani tudi časovni poteki vhodnih testnih signalov s ponazoritvami lastnosti vezja, oziroma meritev. Časovne lastnosti skupaj s površino na rezini in porabo predstavljajo lastnosti vezja  $x_i$ , ki prispevajo v KF iz enačbe (2).

Na primeru testnega gradnika je bilo merjenih dvajset časovnih lastnosti. Poleg zakasnitev in strmin so v KF pripevali še izhodni postavitveni časi (ang. setup time), vhodni zadrževalni časi (ang. hold time), zagonski čas po resetu, najmanjša perioda ure, ter najmanjša širina reset impulza. Na sliki ?? sta v ponazoritev označeni le ena zakasnitev in ena strmina.

Zakasnitve so bile merjene kot čas, ki preteče od trenutka, ko vhodni signal doseže 50% končne vrednosti, do trenutka, ko izhodni signal doseže 50% končne vrednosti. Strmine so bile merjene kot čas v katerem se opazovani signal spremeni od 10% do 90% končne vrednosti.

## 7 OPTIMIZACISKI POSTOPEK

Narejena sta bila dva optimizacijska teka. Cilj prvega je bil najti čimhitrejši gradnik, pri čemer ni bilo dovoljeno povečanje porabe glede na osnovni gradnik podan s strani proizvajalca integriranih vezij (optimizacija na hitrost). V drugem teku je bil cilj čimmanjša poraba, pri čemer so morale časovne lastnosti gradnika ostati najmanj enako dobre, kot pri osnovnem proizvajalčevem gradniku (optimizacija na porabo).

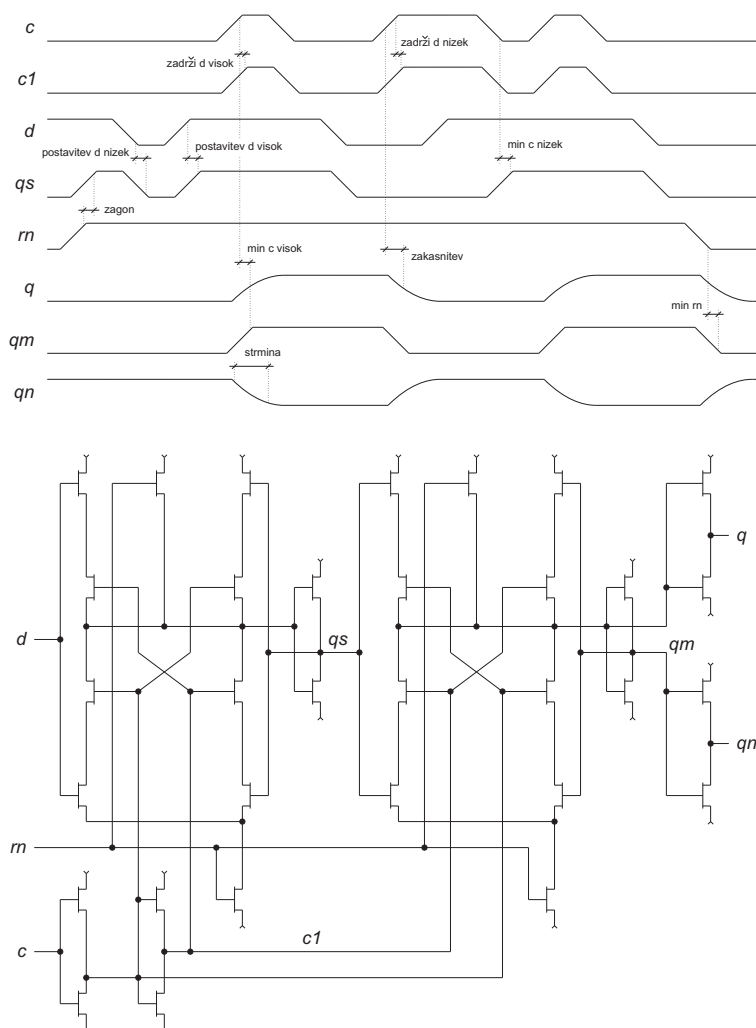
Lastnosti osnovnega proizvajalčevega gradnika so služile kot želene vrednosti  $g_i$ . Visoke kazenske uteži  $p_i$  so zagotovile, da so bile vse lastnosti osnovnega gradnika dosežene v obeh tekih. Tek sta se razlikovala v trgovalnih utežeh  $t_i$ , ki so v prvem teku poudarjale hitrost, v drugem pa nizko porabo.

Gradnik je bil optimiziran za različne pogoje procesa izdelave in delovanja. V poštrev so biti vzeti štirje procesni koti (ang. worst power, worst speed, worst one, worst zero), dve temperaturi ( $-25^\circ\text{C}$ ,  $105^\circ\text{C}$ ), dve napajalni napetosti (3V, 3.6V), dve izhodni bremen (10fF, 220fF) in dva naklona vhodnega signala (60ps, 4ns). K vsem 64-im skrajnim kombinacijam naštetih parametrov je bil dodan še tipični kot.

Za izračun najslabših vrednosti lastnosti gradnika v navedenem območju delovanja je bilo potrebno analize opraviti v štirih kritičnih kotih. Določevanje lastnosti gradnika v ostalih 61-ih kotih ni potrebno. To pomembno zmanjša število kotov na štiri, in sicer:  $\text{wp}/-25^\circ\text{C}/3.6\text{V}/220\text{fF}/4\text{ns}$ ,  $\text{ws}/105^\circ\text{C}/3\text{V}/10\text{fF}/60\text{ps}$ ,  $\text{ws}/105^\circ\text{C}/3\text{V}/220\text{fF}/60\text{ps}$  in  $\text{ws}/105^\circ\text{C}/3\text{V}/220\text{fF}/4\text{ns}$ . Lastnosti optimiziranega gradnika so bile preverjene preko vseh 65-ih kotov.

Neodvisne optimizacijske spremenljivke so bile širine in dolžina kanalov tranzistorjev. Dolžina kanala je bila za vse tranzistorje enaka in je predstavljala eno optimizacijsko spremenljivko. Vseh optimizacijskih spremenljivk je bilo 33. Eksplisitne meje optimizacijskega postopka so bile postavljene od 400nm do  $2\mu\text{m}$  za širine in od 350nm do  $2\mu\text{m}$  za dolžino kanalov. Ker je dolžina kanalov navadno najmanjša možna, je bilo pričakovati, da bo optimizirana dožina na spodnji dovoljeni meji. Navkljub pričakovanjem se je izkazalo, da temu ni vedno tako.

Uporabljeno je bilo vzporedno simulirano ohlajanje z diferencialno evolucijo (ang. Simulated Annealing with Differential Evolution, ali PSADE) [11]. Optimizacijska metoda je tekla na osmih procesorjih AMD Athlon 3GHz. Začetna točka metode je naključna. Metoda je bila ustavljena po 150000 določitvah KF.



Slika 6.: D flip-flop ( $d$  = podatek,  $c$  = ura,  $rn = \overline{\text{reset}}$ ,  $q$  = izhod,  $qn = \overline{\text{izhod}}$ )

## 8 REZULTATI

Lastnosti testiranega gradnika po obeh optimizacijskih tekih, ter lastnosti gradnika iz knjižnice proizvajalca so zbrane v tabeli 2. V stolpcu hitrost so lastnosti gradnika po optimizaciji na hitrost. Optimiziran gradnik je v vseh lastnostih vsaj tako dober, ali boljši, kot proizvajalčev gradnik. Časovne lastnosti gradnika se izboljšajo za do 77%. Zanimivo je, da ima optimiziran hitrejši gradnik celo za 5.8% manjšo porabo, kar pri optimizaciji na hitrost ni bilo pričakovano. Pričakovana pa je končna dolžina kanala tranzistorjev, ki je najmanjša možna (350nm). Za dosego minimuma je optimizacijski postopek potreboval 149689 določitev KF. Prvi gradnik z boljšimi lastnostmi od proizvajalčevega se pojavi v 3285-ti določitvi KF.

V stolpcu poraba v tabeli 2 so zbrani rezultati optimizacije na porabo. Optimiziran gradnik ima vse lastnosti zopet vsaj tako dobre kot proizvajalčev. Poraba gradnika je zmanjšana za 20%, pri čemer se časovne lastnosti izboljšajo za 43%. Zanimivo je, da dolžina kanala tran-

zistorjev optimiziranega gradnika ni najmanjša možna (390nm). Za dosego minimuma je bilo potrebnih 121406 določitev KF. Prvi gradnik z boljšimi lastnostmi od proizvajalčevega se pojavi v 3398-ti določitvi KF.

Kapacitivnosti vhodov gradnika na vrednost KF niso imele vpliva. Ne glede na to se v obeh optimizacijskih tekih zaradi manjših dimezij vhodnih tranzistorjev optimiziranih gradnikov kapacitivnosti vhodov vseeno zmanjšajo.

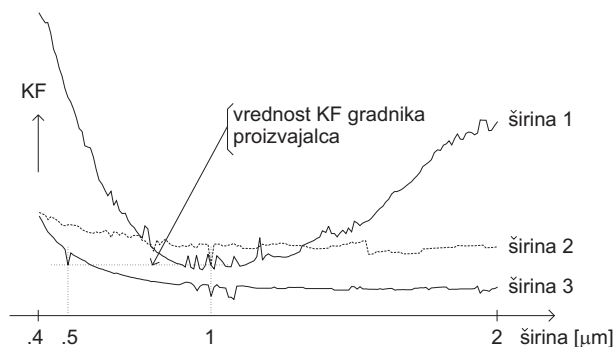
V obeh optimizacijskih tekih je bilo potrebnih le nekaj tisoč določitev KF, da je bil najden prvi gradnik boljši od proizvajalčevega. To pomeni, da je bila večina določitev KF porabljenih za fino določanje končne rešitve, za kar je navadno uporabljena hitra lokalna optimizacijska metoda, npr. [12]. Žal poskusi pospešitve optimizacijskega teka s pomočjo lokalne metode niso bili uspešni. Neuspeh gre pripisati negladkosti poteka KF [13]. Za ponazoritev so na sliki 7 narisani preseki KF po treh optimizacijskih parametrih, t.j. širinah kanalov tranzistorjev.

Na sliki 7 je jasno viden numeričen šum, ki je posle-

Tabela 2:: Rezultati optimizacije D flip-flopa

| lastnost                         |                     | original | hitrost | poraba |
|----------------------------------|---------------------|----------|---------|--------|
| površina                         | [ $\mu\text{m}^2$ ] | 59.1     | 59.1    | 48.5   |
| $c$ na $q\uparrow$ zakasnitev    | [ns]                | 5.17     | 2.95    | 4.19   |
| $c$ na $q\downarrow$ zakasnitev  | [ns]                | 6.71     | 2.78    | 5.97   |
| $c$ na $qn\uparrow$ zakasnitev   | [ns]                | 4.80     | 2.67    | 4.76   |
| $c$ na $qn\downarrow$ zakasnitev | [ns]                | 7.27     | 2.84    | 6.54   |
| $rn$ na $qn$ zakasnitev          | [ns]                | 4.19     | 1.82    | 3.28   |
| $rn$ na $q$ zakasnitev           | [ns]                | 6.03     | 1.60    | 4.17   |
| $c$ na $q\uparrow$ strmina       | [ns]                | 6.49     | 4.45    | 6.47   |
| $c$ na $q\downarrow$ strmina     | [ns]                | 9.96     | 2.61    | 7.88   |
| $c$ na $qn\uparrow$ strmina      | [ns]                | 6.41     | 3.03    | 6.38   |
| $c$ na $qn\downarrow$ strmina    | [ns]                | 9.95     | 3.11    | 9.60   |
| $rn$ na $qn$ strmina             | [ns]                | 6.41     | 3.03    | 6.37   |
| $rn$ na $q$ strmina              | [ns]                | 9.95     | 2.32    | 7.77   |
| $d\uparrow$ na $c$ postavitev    | [ns]                | 1.23     | 0.625   | 1.03   |
| $d\downarrow$ na $c$ postavitev  | [ns]                | 0.984    | 0.983   | 0.982  |
| $d\uparrow$ na $c$ zadrži        | [ns]                | 0.566    | 0.324   | 0.548  |
| $d\downarrow$ na $c$ zadrži      | [ns]                | 0.560    | 0.325   | 0.559  |
| $rn$ na $c$ zagon                | [ns]                | 1.27     | 0.665   | 1.08   |
| min. širina $c_{\text{high}}$    | [ns]                | 1.71     | 1.54    | 1.70   |
| min. širina $c_{\text{low}}$     | [ns]                | 1.30     | 0.970   | 1.30   |
| min. širina $rn$                 | [ns]                | 0.811    | 0.375   | 0.463  |
| poraba                           | [pAs]               | 6.22     | 5.86    | 4.96   |

↑ in ↓ pomenita naraščajočo in padajočo fronto



Slika 7:: Preseki kriterijske funkcije po treh širinah kanalov

dica končnega časovnega koraka v tranzientni analizi. Z zmanjševanjem časovnega koraka postaja numeričen šum manjši. Vendar fino določanje končne rešitve z lokalno optimizacijsko metodo ne uspe, neglede na to kako majhen je časovni korak. Zato je potrebno celoten optimizacijski tek izvesti z robustno globalno metodo. Za zagotovitev točnosti mora biti časovni korak kljub temu relativno majhen, kar pomeni dolge tranzientne

analize in posledično dolge optimizacijske teke. En optimizacijski tek traja pet dni na osmih paralelnih procesorjih.

## 9 ZAKLJUČEK

Osnovni digitalni gradniki podani v knjižnici proizvajalca predstavljajo nabor gradnikov, ki naj jih načrtovalec integriranih vezij na tranzistorskem nivoju ne bi spreminjal. Primer v članku pokaže znatno izboljšanje lastnosti osnovnega gradnika z uporabo optimizacijskega postopka na tranzistorskem nivoju. Doseženo je bilo do 77% izboljšanje časovnih lastnosti gradnika brez povečanja porabe. Oziroma na drugi strani 20% zmanjšanje porabe ob enakih časovnih lastnostih. Uporaba gradnikov prirejenih na specifične zahteve v vezju bi lahko vodila k hitrejšim digitalnim vezjem z nižjo porabo. Da bi predlagan postopek postal učinkovit, bi bilo potrebno optimizacijo na tranzistorskem nivoju vključiti v orodja za sintezo. Dolžina optimizacijskih tekov ostaja zaradi šumnosti poteka kriterijske funkcije glavna ovira. Ponovna preveritev optimiziranih gradnikov v sintezacijskem orodju zaradi nedostopnosti (plačljivosti) slednjih, kakor tudi zaradi nedostopnosti orodij za ekstrakcijo opisa gradnika iz tranzistorskega na višji nivo (proizvajalci integriranih vezij ekstrakcijskih orodij ne podajajo), ni bila narejena.

## ZAHVALA

Raziskavo je omogočilo Ministrstvo za visoko šolstvo, znanost in tehnologijo Republike Slovenije v okviru programa P2-0246 - Algoritmi in optimizacijski postopki v telekomunikacijah.

## LITERATURA

- [1] H. Kaeslin, *Digital integrated circuit design: from VLSI architectures to CMOS fabrication*, Cambridge University Press, 2008.
- [2] *HSPICE® simulation and analysis user guide*, Synopsys®, 2005.
- [3] K.S. Kunderth, *The designer's guide to SPICE and Spectre*, Kluwer Academic Publishers, 1995.
- [4] *Virtuoso® Spectre® circuit simulator user guide*, Cadence Design Systems, Inc., 2008.
- [5] T. Tuma, Á. Bűrmen, *Circuit simulation with SPICE Opus, theory and practice*, Birkhäuser, 2009.
- [6] D.E. Thomas, P.R. Moorby, *The Verilog hardware description language*, 5<sup>th</sup> ed. Kluwer Academic Publishers, 2003.
- [7] V.A. Pedroni, *Circuit design with VHDL*, Massachusetts Institute of Technology, 2004.
- [8] P. Kurup, T. Abbasi, *Logic synthesis using Synopsys®*, 2<sup>nd</sup> ed. Kluwer Academic Publishers, 1997.
- [9] H. Bhatnagar, *Advanced ASIC chip synthesis using Synopsys® Design Compiler™ Physical Compiler™ and PrimeTime®*, 2<sup>nd</sup> ed. Kluwer Academic Publishers, 2002.
- [10] Á. Bűrmen et al., "Automated robust design and optimization of integrated circuits by means of penalty functions," *AEÜ, International Journal of Electronics and Communications*, vol. 57, no. 1, pp. 47–56, 2003.



- [11] J. Olenšek, Á. Bűrmen, J. Puhon, T. Tuma, "DESA: a new hybrid global optimization method and its application to analog integrated circuit sizing," *Journal of Global Optimization*, vol. 44, no. 1, pp. 1–25, 2008.
- [12] R. Hooke, T. Jeeves, "Direct search solutions of numerical and statistical problems," *Journal of the Association for Computing Machinery*, vol. 8, no. 2, pp. 212–229, 1961.
- [13] Á. Bűrmen, I. Fajfar, T. Tuma, "Combined Simplex-Trust-Region optimization algorithm for automated IC design," *Proceedings of ECCTD07 European Conference on Circuit Theory and Design*, pp. 543–546, 2007.

**Janez Puhon** je leta 2000 doktoriral s področja elektrotehnike na Univerzi v Ljubljani. Je asistent na Fakulteti za elektrotehniko. Njegovo področje raziskovanja obsega modeliranje, simulacijo in optimizacijske postopke pri računalniškem načrtovanju vezij.

**Iztok Fajfar** je diplomiral (1991), magistriral (1994) in doktoriral (1997) s področja elektrotehnike na Fakulteti za elektrotehniko Univerze v Ljubljani. V letu 1991 je bil raziskovalec na Inštitutu Jožef Štefan v Ljubljani, nakar je konec leta zasedel raziskovalno mesto na Fakulteti za elektrotehniko Univerze v Ljubljani. Trenutno na fakulteti zaseda mesto izrednega profesorja. Poučuje več uvodnih in nadaljevalnih predmetov s področja računalniškega programiranja. Pri razvoju programske opreme je sodeloval v več industrijskih projektih s podjetji Adacta in ICE Telecom. Njegovo področje raziskovanja vključuje načrtovanje in optimizacijo elektronskih vezij.

**Tadej Tuma** je diplomiral (1988), magistriral (1991) in doktoriral (1995) na Fakulteti za elektrotehniko Univerze v Ljubljani. Na isti fakulteti je redni profesor, kjer poučuje štiri dodiplomske in tri podiplomske predmete. Njegovi raziskovalni interesi so predvsem na področju računalniške analize in načrtovanja vezij.

**Árpád Bűrmen** se je rodil leta 1976 v Murski Soboti. Leta 2003 je na Univerzi v Ljubljani doktoriral s področja elektrotehnike. Od leta 2005 dalje je docent na fakulteti za elektrotehniko. Njegovo raziskovalno področje vključuje zvezno in dogodkovno simulacijo vezij in sistemov, optimizacijske metode, njihovo konvergenco in uporabo, ter algoritme za vzporedno in porazdeljeno računanje. Je eden vodilnih razvijalcev simulatorja vezij SPICE OPUS. Objavil je preko 20 člankov v recenziranih revijah.