

Realizacija 8-bitnega polnega seštevalnika v dvojiški QCA logiki

Romana Grilj^{1,*,+}, Gregor Poročnik^{2,*,+}, Jernej Grosar^{3,*,+}, Vesna Novak^{4,*,+}, and Matic Pajnič^{5,*,+}

¹romana.grilj@gmail.com

²gp3964@student.uni-lj.si

³grosar.jernej@gmail.com

⁴0452452@student.uni-lj.si

⁵matic.pajnic@gmail.com

*Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

+ti avtorji so enakovredno doprinesli k delu

POVZETEK

Pri predmetu Nekonvencionalne platforme in metode procesiranja smo se spoznali z novimi tehnologijami, ki nam omogočajo nadaljnji razvoj in napredek na področju računalniške opreme in procesiranja. Ena izmed takih tehnologij je nanotehnologija. Namen tega dela je primerjava tehnologij na temo **razmeščanja ure za dvojiški kvantni celični avtomati** (angl. Quantum-dot Cellular Automata, QCA) in izbira ene tehnologije, v kateri bomo realizirali obsežnejšo strukturo. Naša naloga je realizirati 8-bitni polni seštevalnik **s kvantni celični avtomati** (angl. Quantum-dot Cellular Automata, QCA) celicami.

UVOD

Dvojiški kvantni celularni avtomati (v nadaljevanju QCA) so avtomati zgrajeni iz kvadratnih nano celic velikosti nekaj nanometrov katere so razporejene v mrežo. Vsaka taka celica hrani v sebi dva elektrona, ki se nahajata vsak v svoji kvantni piki, lahko pa sta tudi dva v eni piki, vendar pa to ni ravno pogost pojav. V posamezni celici so štiri take pike (vsaka v svojem kotu celice), med seboj pa so sosednje pike povezane s kvantnimi tuneli po katerih se lahko premikata elektrona. V dvojiškem QCA sistemu sta dve stabilni stanji, ko je en elektron v zgornjem levem kotu, drugi pa v spodnjem desnem kotu - to stanje predstavlja logično ničlo in ko je en elektron v zgornjem desnem kotu, drugi pa v spodnjem levem kotu - to stanje predstavlja logično enico.

Do takega stabilnega stanja v posamezni celici pride, ko je okolica celice (torej njene sosednje celice) v stanju, ki elektrona v celici zaradi odbojnih sil prisili v enega od dveh zgoraj opisanih fiksnih položajev. Ker se okolica posamezne celice v vezjih običajno spreminja, celica v določenem zaporedju preklaplja med svojimi stabilnimi stanji. Če gledamo več celic naenkrat, je jasno, da prihaja pravzaprav do procesiranja oz. prenosa informacij. V nasprotju s konvencionalnimi logičnimi vezji QCA tehnologija združuje procesa procesiranja in prenašanja informacij. Kot QCA kaže potencial v smislu časovne izkoriščenosti celotnega vezja, saj v idealnem primeru noben segment vezja ni neaktiven. Vsak kvantni tunel vsebuje pregrado katero zapremo, če želimo postaviti celico v "zaklenjen" položaj. V tem položaju sta elektrona zaklenjena vsak v svoji kvantni piki in se ne moreta premakniti iz nje. To zaklepanje uporabljamo za prenos "signala" iz ene celice v drugo (navaden prenos signala - analogno si lahko to predstavljamo kot prenos signala po žici), ter izvajanje logičnih operacij, ko signal spustimo skozi logična vrata zgrajena iz večih kvantnih celic.

Eden najbolj pomembnih elementov, katerega moramo upoštevati pri načrtovanju QCA vezij, je uporaba urinega signala oziroma bolje rečeno razdeljevanje celic (ali skupin celic) v posamezne faze urinega signala. Če pri načrtovanju ne upoštevamo te razdelitve lahko pride do nezaželenih motenj in posledično do nepravilnega rezultata operacij. Na primer: če imamo neko dolgo vodilo sestavljeno iz kvantnih celic, se pri prenosu informacije po tem vodilu zaradi motenj iz okolice zgodi, da se signal popači in tako lahko dobimo na izhodu vodila namesto logične enke, katero smo podali kot vhod v vodilo, logično ničlo. Da se izognemo temu nezaželenemu pojavu uporabimo urin signal in kvantne celice razdelimo na skupine. Vsaka skupina celic je v nekem trenutku v eni od štirih faz urinega signala, pri tem pa moramo paziti, da sta dve sosednji skupini v dveh različnih fazah urinega signala.

Poznamo tri vrste razporeditev celic v take skupine, ki se ločijo po tem, na kakšen način so te skupine fizično razporejene v prostoru: tile-based, SQUARES-based in gate-based razporeditve, ki so opisane v nadaljevanju.

PREGLED PRISTOPOV K RAZMEŠČANJU URE

Pri Gate based razporeditvi so kvantne celice razporejene v skupine tako, da posamezna logična vrata tvorijo eno skupino (slika 1).

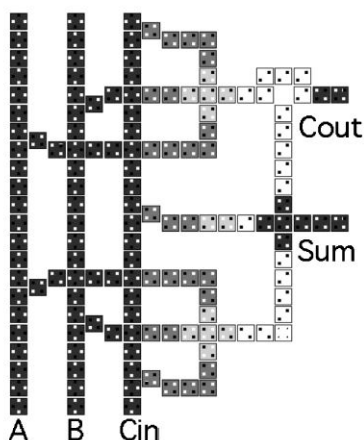


Figure 1. Gate-based razporeditev uriniga signala.

Squares razporeditev urinega signala upošteva, da posamezna urina perioda zajema celice, ki so v kvadratu velikosti $n \times n$ celic. Paziti moramo, da posamezne skupine urinih period postavimo tako, da levo in desno, ter pod in nad neko skupino celic z eno urino periodo ni druge skupine, ki bi imela isto urino periodo.

Tile based razporeditev je narejena tako, da so urine periode razdeljene na pasove. Izvajanje operacij pa si ponavadi sledi od leve proti desni, tj. vhodi so na levi strani, izhodi pa na desni strani posameznega pasa. Ta metoda je najbolj enostavna za implementacijo in je v večini primerov tudi najhitrejša med vsemi tremi metodami.

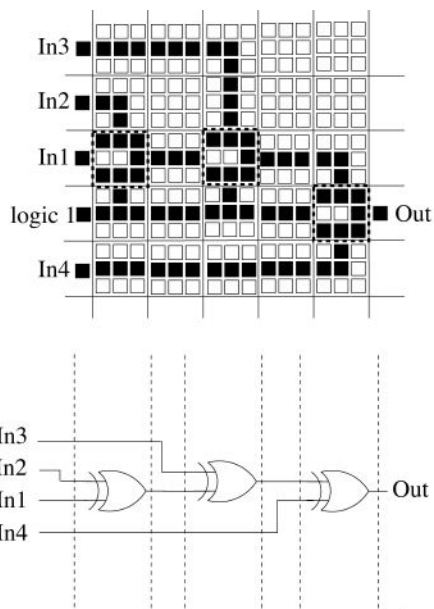


Figure 2. Tile-based razporeditev uriniga signala.

Huang¹ predlaga, da se QCA vezja sestavljajo iz segmentov celic (slika 3). Ti imajo 3×3 konfiguracijo celic, ki realizirajo funkcijo tega segmenta, ter vhodnih/izhodnih celic na srednji poziciji zunaj posamezne stranice segmenta. 3×3 konfiguracija naj bi zagotavljala fleksibilne logične funkcije, ter nudila primerna nivoja polarizacije in procesiranja. Huang zagovarja enodimenzionalno načrtovanje ure v QCA vezjih (tile-based razporeditev), to pomeni, da si skupine celic, ki so v zaporednih urinih fazah, v načrtu sledijo od leve proti desni, s tem da so vse celice v posameznem stolpcu v isti urini fazi. Za samo

implementacijo različnih logičnih funkcij naj bi zadostovala gradnika *Majority Voter* (MV) in *Inverter* (INV). MV je v 3x3 konfiguraciji funkcija s 3 vhodi in 1 izhodom ($Maj(A,B,C) = AB + BC + AC$). MV in INV Huang¹ združi v t.i. *MV-Like tile*, torej 3x3 konfiguracijo, podobno MV, s tem, da nudi še poljubno invertiranje vhodov. S to konfiguracijo naj bi bilo možno v QCA vezjih realizirati večino logičnih funkcij z dobro izkoriščenostjo prostora. Pri taki konfiguraciji naj bi pri tile-based pristopu celotno QCA vezje za implementacijo MV in INV gradnikov porabilo sicer več posameznih celic kot gate-based pristop, vendar se celotna površina vezja zaradi tega naj ne bi povečala napram gate-based pristopu.

Huang¹ kot najbolj primerno 3x3 konfiguracijo za načrtovanje QCA vezij obravnava t.i. ortogonalno konfiguracijo (Slika 3). Ta ima 3 vhode (A, B, C) in en izhod (F). Ta konfiguracija naj bi imela edinstvene procesirne lastnosti. Kot primer navede, da je s tako konfiguracijo s selektivnim odvzemanjem posameznih celic možno realizirati 2-vhodna NAND vrata ali 2-vhodna NOR vrata, če vhod B deluje kot stikalo med funkcijama.

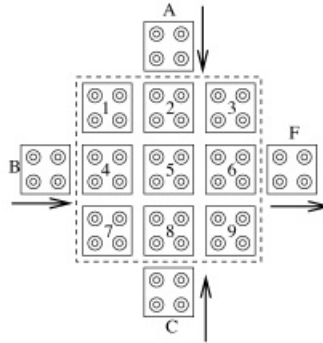


Figure 3. Ortogonalna konfiguracija.

Tabela 1 povzema rezultate testiranja 4 različnih QCA vezij po tile-based, SQUARES-based in gate-based pristopu. Testirana vezja so 2-4 dekodirnik, eno-bitni polni seštevalnik, preverjevalnik paritete, ter 2-1 multiplexer. Najbolj pomembna kriterija sta število celic in število urinih con, potrebnih za implementacijo vezja. Tile-based pristop se pri implementaciji vseh testiranih vezij izkaže kot najbolj cenovno učinkovit tako po zasedenem prostoru vezja kot po optrebnem številu urinih con. Pri tem pa je potrebno upoštevati, da rezultati za gate-based pristop močno izstopajo pri preverjevalniku paritete, torej pri 25% testiranih vezij. V tem oziru je popolnoma možno, da bi se ob testiranju implementacije več različnih vezij ta cena v povprečju zmanjšala in bi bila bližje ali celo nižja kot povprečna cena pri tile-based pristopu.

Circuit		2-to-4 Decoder	One Bit Adder	Parity Checker	2-to-1 MUX
Tile-based	# of Tiles	60	64	25	17
	Total # of cells	588	576	225	153
	# Clk zones	6	8	5	4
SQUARES	# of SQUARES	48	56	56	12
	Total # of cells	1200	1400	1400	625
	# Clk zones	11	15	19	5
Gate-based	Total # of cells	400	396	1479	234
	# Clk zones	5	5	22	5

Table 1. Vezja po tile-based, SQUARES-based in gate-based pristopu.

USE clocking scheme

Tehnologija kvantnih celični avtomatov (QCA) je novo nastajajoča tehnologija, katera bi lahko nasledila obstoječo CMOS tehnologijo v katere izhaja trenutna arhitektura za procesiranje podatkov. Kot ena izmed rešitev postopka izdelave je predlagana shema USE »universal, scalable and efficient clocking scheme« oziroma univerzalna nadgradljiva in učinkovita shema. S predlagano shemo je možno realizirati rešitev na manjši platformi in veliko manjšim zaostankom kot pri trenutno obstoječih shemah. Urina shema USE je razširljiva, regularna in dovolj fleksibilna rešitev, ki jo je možno realizirati z obstoječo produkcijsko tehnologijo, da omogoča enostavno usmerjanje, možnost povratnih poti, realizacijo z manjšim številom celic, poleg tega pa preprečuje težave kot je termodinamika katero se rešuje z realizacijo daljših vezi oziroma poti. Cilj tehnologije QCA je, da z se

z novo tehnologijo realizira boljšo procesno moč na manjši površini, pri tem pa se poraba energije, ki jo moramo dovajati v sistem še zmanjša.²

Prehod med dvema logičnima stanjema se pojavlja adiabatno. Nenadne spremembe v stopnji polarizacije nastavijo sistem v metastabilno stanje, kar lahko povzroči nezaželene zamude ali napačno logično obdelavo. Tako se morajo kvantne pike v QCA celici postopno prilagajati na morebitne ovire, da se omogoči ali zavrne elektronsko tuneliranje. To je mogoče doseči z zunanjo uro, ki zagotavlja urine signale, kateri nadzirajo postopek adiabatnih sprememb. Ura mora biti sposobna zagotoviti svoje signale v vsako celico. Da bi dosegli te zahteve, jo je treba natančno namesti pod QCA plasti.³

Prenos informacij se v okviru celic zgodi na takšen način, da se v področju, ki je v stanju »preklop« spremeni stanje glede na celice, ki so v sosednjem območju in v stanju »zadržanje«. Informacija se tako prenese iz območja v »zadržanje« stanju v območje v stanju »preklop«. Ideja USE sheme je ta, da se celice sosednjih števil vedno pozicionirane ena ob drugo, celice označene z nesosednjimi števili pa so bolj oddaljene oziroma poševno orientirane. Stanja se tako prenašajo iz celic v celice v takšnem zaporedju kakor so naraščujoče oštevilčene. Shema USE omogoča uporabo ravnih poti, kokor tudi manjših povratnih poti, ukrivljenih poti ali zank poljubnih dolžin. Takšna vezja je mogoče poljubno razširjati saj dobimo enakovredno strukturo, če posamezne enote združujemo.

Signal distribution grid

Prečkanje žic v isti ravnini je bil največji izziv v QCA avtomatih že od začetka njihovega razvoja. Tougaw⁴ je predstavil mrežo distribucije signalov, ki omogoča več paralelnih prečkanj samo z bližnjimi celičnimi interakcijami, s konstantnim časom za distribucijo signalov ne glede na število vhodov ali izhodov in regularno oblikovanih regij urinega signala kar bi bilo lažje za izdelavo.

Mreža distribucije signalov omogoča distribucijo določenega števila vhodnih signalov, da naredijo arbitrarne povezave na določeno število izhodnih signalov. To se napravi samo z sosednjimi celičnimi interakcijami in samo z fiksnim številom urinih ciklov. Ne glede na število vhodov ali izhodov in z uporabo sosednjih urinih regij katere so dostopne v obsegu naprave. Za doseg cilja mreža distribucije signalov vključuje dvoslojne urine signale komplimentarne pare na vertikalnih kolonah za uspešno rešitev problema nenamerno zaklenjenih celic in horizontalni rovi dvojne širine za eliminiranje možnosti napak predstavljenih navzdol od namernih povezav.

REALIZACIJA SEŠTEVALNIKA

Za realizacijo smo si izbrali 8 bitni polni seštevalnik ter Tile-based razporeditev. Pri realizaciji Tile-based polnega seštevalnika smo si pomagali s primerom enobitnega polnega seštevalnika, kateri je predstavljen v članku¹, za poganjanje simulacije pa smo uporabili privzete nastavitve Coherence Vector simulatorja v QCAdesignerju. Po realizaciji enobitnega polnega seštevalnika, tako kot je podan v članku in po prvem zagonu simulacije smo ugotovili, da ta seštevalnik ne deluje pravilno. Izkazalo se je, da se problem pojavi pri križanju žic (baseline element v članku), ko se v nekaterih primerih spremeni polarizacija celic sinala iz vhoda A v delu pod križiščem z žico po kateri potuje signal iz vhoda B. Prav tako se na izhodu MV-like vrat pojavi pri v nekaterih kombinacijah vhodov napačna vrednost. Ta problem smo rešili tako, da smo križanje žic realizirali na večih nivojih, MV-like vrata pa smo zamenjali z zaporedno vezavo negatorja in navadnih MV vrat. Ugotovili smo tudi, da sta v članku napačno označena izhoda (Sum in Cout), katera morata biti med seboj zamenjana. Ko je enobitni seštevalnik deloval je bilo precej enostavno realizirati dvobitnega. Potrebno je bilo narediti dva enobitna seštevalnika, ter ju povezati med seboj, med njiju pa vriniti še eno četrtno urine priode, prav tako je bilo potrebno zakasniti vhode seštevalnika drugega bita, ter izhod seštevalnika prvega bita. Dvobitni seštevalnik lahko realiziramo z enim polovičnim seštevalnikom (za prvi bit) in enim polnim seštevalnikom (za drugi bit). Polovični seštevalnik smo realizirali tako, da smo vhodu Cin določili fiksno vrednost -1 (logično ničlo).

8-bitni polni seštevalnik

Za realizacijo smo si izbrali 8 bitni polni seštevalnik (slika 4). 8 bitni seštevalnik je sestavljen iz 8 polnih seštevalnikov (slika 5), pri katerih je prvi lahko polovičen, saj ne upošteva preliva. Posamezna logična enota polnega seštevalnika je sestavljena iz dveh polovičnih seštevalnikov ter OR vrat in ima 3 vhode in 2 izhoda.

Vhode v polni seštevalnik sestavljajo vrednosti A ki je n-ta pozicija bita v prvem številu, B ki je n-ta pozicija bita v drugem številu, ter Cin ki se upošteva kot preliv iz rezultata izračuna n-1 mesta.

Izhod polnega seštevalnika je končni rezultat n-tega mesta (Sum) in Cout ki služi kot indikator, da je prišlo do preliva.

Tile-based polni seštevalnik

Pri realizaciji Tile-based polnega seštevalnika smo si pomagali s primerom enobitnega polnega seštevalnika, kateri je predstavljen v članku¹, za poganjanje simulacije pa smo uporabili privzete nastavitve Coherence Vector simulatorja v QCAdesignerju. Po realizaciji enobitnega polnega seštevalnika, tako kot je podan v članku in po prvem zagonu simulacije smo ugotovili, da ta

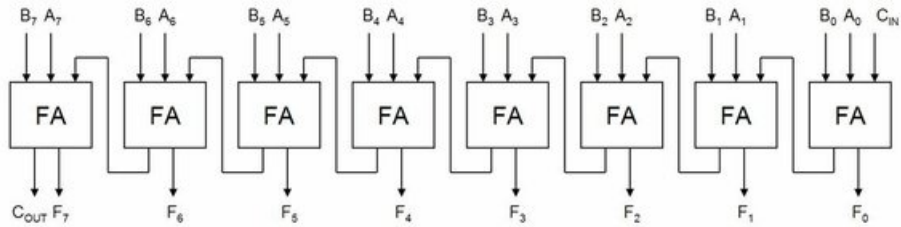


Figure 4. 8-bitni polni seštevalnik.

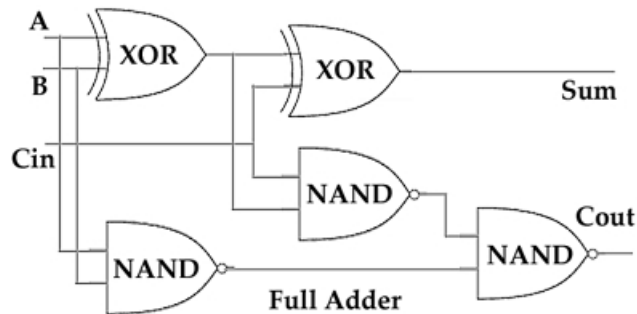


Figure 5. polni seštevalnik.

seštevalnik ne deluje pravilno. Izkazalo se je, da se problem pojavi pri križanju žic (baseline element v članku), ko se v nekaterih primerih spremeni polarizacija celic signala - iz vhoda A v delu pod križiščem z žico po kateri potuje signal iz vhoda B. Prav tako se na izhodu MV-like vrat pojavi v nekaterih kombinacijah vhodov napačna vrednost.

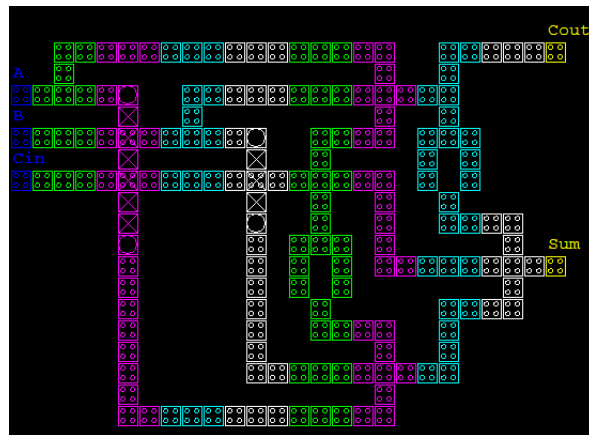


Figure 6. 1-bitni tile-based polni seštevalnik.

Ta problem smo rešili tako, da smo križanje žic realizirali na večih nivojih, MV-like vrata pa smo zamenjali z zaporedno vezavo negatorja in navadnih MV vrat (slika 6). Ugotovili smo tudi, da sta v članku napačno označena izhoda (Sum in Cout), katera morata biti med seboj zamenjana. Ko je enobitni seštevalnik deloval je bilo precej enostavno realizirati dvobitnega. Potrebno je bilo narediti dva enobitna seštevalnika, ter ju povezati med seboj, med njiju pa vrniti še eno četrtno urine priode, prav tako je bilo potrebno zakasnitih vhode seštevalnika drugega bita, ter izhod seštevalnika prvega bita.

Dvobitni seštevalnik (slika 7) lahko realiziramo z enim polovičnim seštevalnikom (za prvi bit) in enim polnim seštevalnikom (za drugi bit). Polovični seštevalnik smo realizirali tako, da smo vhodu Cin določili fiksno vrednost -1 (logično ničlo).

Delujočega 8 bitnega polnega seštevalnika s tile-based razvrstitvijo ure nam ni uspelo realizirati zaradi prevelike kompleksnosti in nemogočega razhroščevanja tako kompleksnega vezja v programu QCAdesigner. Slika 8 prikazuje kako naj bi približno zgedel tak seštevalnik.

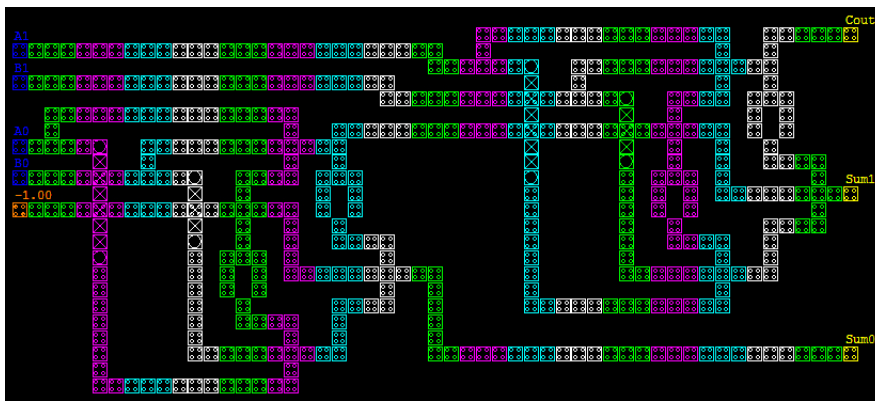


Figure 7. 2-bitni tile-based polni seštevalnik.

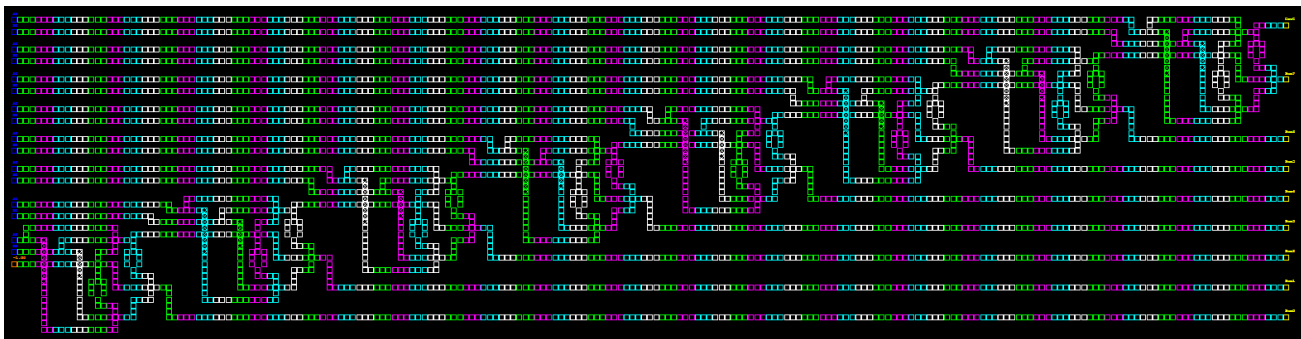


Figure 8. 8-bitni tile-based polni seštevalnik.

Gate-based polni seštevalnik

Za gate-based polni seštevalnik (slika 9) smo vzeli seštevalnik realiziran v⁵. Tega smo preuredili tako, da ima vhode na levi strani za kasnejše lažje povezovanje.

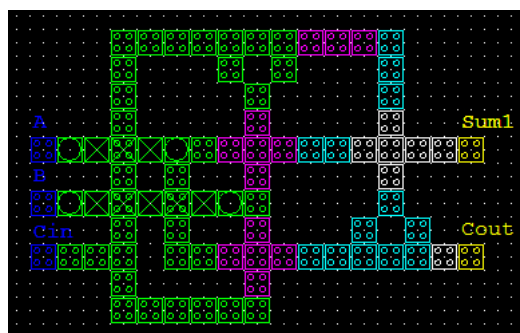


Figure 9. 1-bitni gate-based polni seštevalnik.

Pri dvobitnem seštevalniku je bilo potrebno dodati še nekaj dodatnih urinih priod (slika 10).

Pri tej verziji nam je uspelo realizirati največ 3-bitni polni seštevalnik (slika 11) iz istih razlogov kot pri tile-based polnemu seštevalniku.

Tudi za ta tip seštevalnika smo naredili skoraj primer kako naj bi približno izgledal polni 8-bitni gate-based seštevalnik (slika 11).

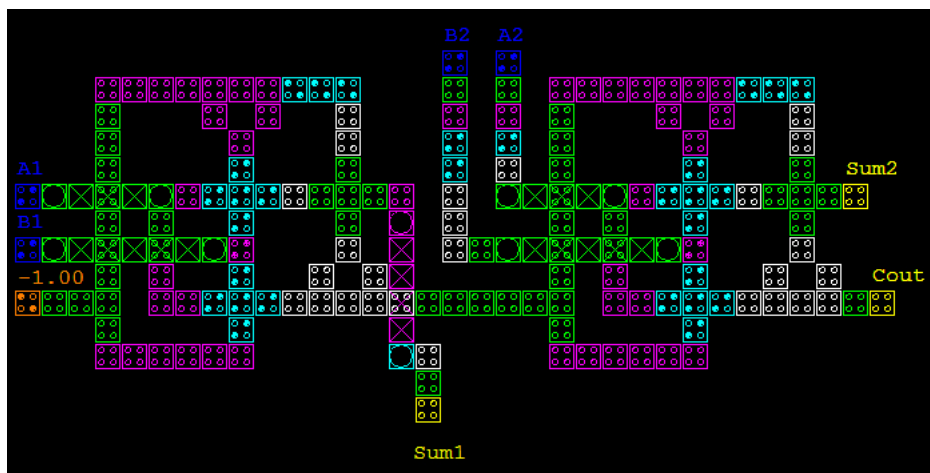


Figure 10. 2-bitni gate-based polni seštevalnik.

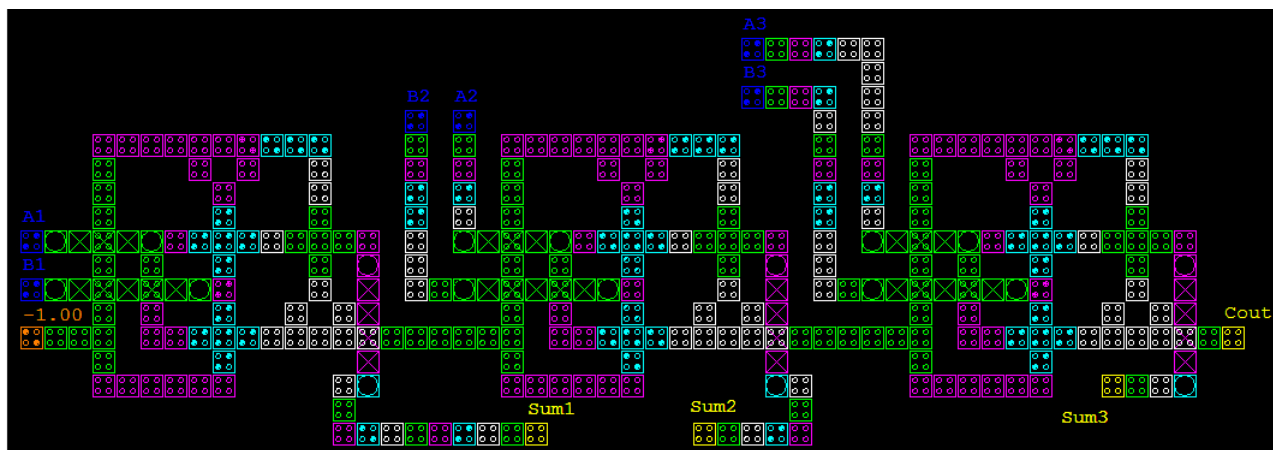


Figure 11. 3-bitni gate-based polni seštevalnik.

Primerjava

Ker nam je uspelo realizirati delujoč le 2-bitni seštevalnik v tile-based, lahko primerjamo samo tega. Pri tile-based razporeditvi smo porabili veliko več celic in površine - kar se lepo vidi če primerjamo sliki 8 in 12, saj je bilo potrebno urine periode razdeliti na pasove, vhode in izhode obeh bitov pa sinhronizirati med seboj.

S stališča porabljenih plasti sta obe izvedbi enakovredni - pri obeh smo uporabili 3 plasti. Pri gate-based razporeditvi smo porabili 2 in 1/4 urine periode, pri tile based pa 4 in 1/4 urinih period.

Za to, da bi bil tile based seštevalnik v vsaj eni metriki boljši kot gate-based (kot je navedeno v članku¹), bi rabili več znanja in časa, da bi ga lahko optimizirali, verjetno bi ga bilo potrebno popolnoma preurediti.

Zaključek

Največ težav smo imeli s samim orodjem QCADesigner. Pomagali smo si s članki in že realiziranimi seštevalniki, ki pa so v naših primerih vedno vračali drugačne rezultate. Ista implemetacija seštevalnika je vračala na različnih računalnikih različne rezultate, kljub uporabi različnih simulatorjev. Poskusili smo tudi z uporabo novejše različice QCADesigner-ja, ki je bila objavljena na spletni učilnici. Rezultati so bili podobni.

Težave smo imeli tudi s shranjevanjem realiziranih seštevalnikov. Največkrat shranjevanje ni bilo možno, ko pa je uspelo, pa smo ob ponovnem zagonu dobili eno veliko zeleno piko, namesto seštevalnika. Tako smo morali isti seštevalnik implementirati večkrat.

Dodatna težava je bila pomanjkanje znanja o samem simulatorju. Predvidevamo, da bi vsaj kakšen od več-bitnih seštevalnikov deloval pravilno ob pravih nastavitvah simulatorja.

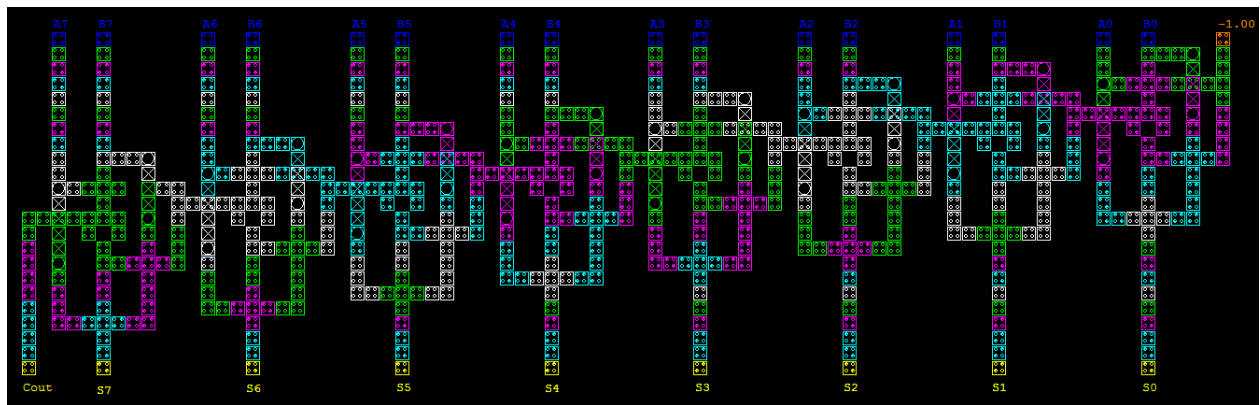


Figure 12. 8-bitni gate-based polni seštevalnik.

Izjava o prispevku avtorjev

M.P.: Poglavlje Tile-based razporeditev.

J.G.: Uvod in uporaba urnega signala, realizacija vseh tile-based polnih seštevalnikov

V.N.: Povzetek metodologije za standardni cell design za QCA, realizacija 4-bitnega single-layer seštevalnika po članku - nedeljujoče, zaključek

G.P.: Povzetek USE sheme za QCA, realizacija 3, 4 in 8-bitnega gate-based seštevalnika

R.G.: Povzetek mreže distribucije signalov

Literatura

1. Huang, J., Momenzadeh, L. S., Ottavi, M. & Lomabrdi, F. Tile-based qca design using majority-like logic primitives. *The journal of small papers* **1**, 513–517 (2005).
2. Campos, C. A. T., Marciano, A. L., Neto, O. P. V. & Torres, F. S. Use: A universal, scalable, and efficient clocking scheme for qca. *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS* **35** (2016).
3. Reis, D. A., Campos, C. A. T., Soares, T. R. B. S., Neto, O. P. V. & Torres, F. S. A methodology for standard cell design for qca. *The journal of small papers* **1**, 2114–2117 (2016).
4. Douglas, T., Justin, S. & Jeffrey, D. A signal distribution grid for quantum-dot cellular automata. *J Comput Electron* **20**, 446–454 (2016).
5. Orač, T. Realizacija aritmetično-logičnih primitivov s strukturami kvantnih celičnih avtomatov. *Diploma* **1** (2007).