

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

# Primerjava QCA struktur s CMOS vezji

Seminarska naloga pri predmetu  
Zanesljivost in vrednotenje računalniških sistemov

Blaž Lampreht, Luka Štepančič,  
Igor Vizec, Boštjan Žankar

# Kazalo

<b>1</b>	<b>Uvod</b>	<b>4</b>
1.1	Predstavitev problema . . . . .	4
1.2	Predstavitev QCA tehnologije . . . . .	5
1.3	Predstavitev CMOS tehnologije . . . . .	7
<b>2</b>	<b>Metode</b>	<b>8</b>
2.1	Orodja . . . . .	8
2.2	Poraba energije . . . . .	9
2.3	Zakasnitev vezij in frekvence delovanja . . . . .	9
2.4	Površina vezij . . . . .	10
<b>3</b>	<b>Seštevalnik</b>	<b>12</b>
3.1	Tipi seštevalnikov . . . . .	12
3.2	QCA implementacije . . . . .	14
3.2.1	RCA seštevalnik . . . . .	14
3.2.2	CLA seštevalnik . . . . .	14
3.2.3	CSA seštevalnik . . . . .	16
3.3	CMOS implementacija . . . . .	18
3.4	Primerjava . . . . .	19
<b>4</b>	<b>Množilnik</b>	<b>25</b>
4.1	Predstavitev serijsko-paralelnega množilnika . . . . .	25
4.2	CMOS izvedba . . . . .	26
4.3	QCA izvedba . . . . .	26
4.4	Primerjava zasedenega prostora . . . . .	26
4.5	Primerjava zakasnitev . . . . .	27
4.6	Primerjava porabe energije . . . . .	28
<b>5</b>	<b>Pomnilnik</b>	<b>29</b>
5.1	Paralelna arhitektura . . . . .	29
5.2	Serijska arhitektura . . . . .	30
5.3	Drevesna H-arhitektura . . . . .	30
5.4	Hibridna arhitektura . . . . .	30
5.5	Primerjave QCA pomnilnikov . . . . .	32
5.5.1	Dostopni čas . . . . .	32
5.5.2	Površina . . . . .	32

5.6	Ugotovitve . . . . .	37
5.6.1	Splošna primerjava . . . . .	37
5.6.2	Efektivna področja . . . . .	37
5.7	QCA vs. CMOS . . . . .	38
<b>6</b>	<b>Zaključek</b>	<b>39</b>

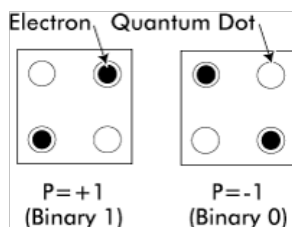
# 1 Uvod

## 1.1 Predstavitev problema

Človeški um že od začetkov prvih civilizacij teži k ustvarjanju novih izumov ter poskušanju razumevanja življenja in našega stvarstva. Seveda, pa si želimo, našo pot do razumevanja vsega čim bolj olajšati, zato se je že dolgo tega pojavila ideja o avtomatiziranem stroju, ki bi rutinska in kasneje kompleksnejša opravila izvrševal namesto človeka. Prvi začetki računalništva segajo v daljno zgodovino človeštva, saj po najdbah arheologov vemo, da so že stari Grki okoli 100 do 150 let pred našim štetjem uporabljali mehanični kalkulator, ki so ga arheologi poimenovali Antikythera. Ampak glavna prelomnica v avtomatizaciji računanja se je zgodila v 50 letih zaradi izuma tranzistorja. Prva integrirana vezja so se pojavila konec 50ih oz. začetek 60ih let prejšnjega stoletja konec 70ih pa prva CMOS vezja, s katerimi se je začela moderna doba računalništva. Sedaj pa se pred nami počasi odpira nova doba računalništva - računalništvo, ki temelji na takoimenovanih kvantnih celularnih avtomatih. V seminarski nalogi bomo predstavili strukture, ki so bile do danes že vsaj v teoriji realizirane in naj bi se v računalništvu uporabljale. Primerjava se sprva zdi razmeroma preprosta, vendar že takoj naletimo na problem. Strukture v QCA tehnologiji zaenkrat obstajajo le v teoriji ter simulacijah. Dejanskih *'in-vivo'* struktur zaenkrat še ni bilo realiziranih v in se jih posledično tudi ne da testirati. Ko pa se bo QCA tehnologija razvila do te mere, da bodo na voljo že realne strukture, pa bodo CMOS vezja še manjša in hitrejša od današnjih. Primerjava je tako mogoča le na ta način, da se odločimo za neko obdobje CMOS tehnologije ter jo primerjamo s potencialnimi zmožnostmi QCA tehnologije. Predstavili bomo rezultate primerjanj že ustaljenih CMOS vezij s QCA strukturami ter navedli prednosti in morebitne slabosti nove tehnologije.

## 1.2 Predstavitev QCA tehnologije

Do sedaj nam je s konstantnim manjšanjem tranzistorjev uspevalo izdelovati vedno hitrejša vezja. Sčasoma, pa bo majhnost tranzistorja dosegla svojo mejo, kar pomeni, da če želimo slediti zahtevam Moorovega zakona moramo poseči po drugačnih tehnologijah. Ena izmed takih tehnologij prihodnosti so kvantni celični avtomati. Temelj kvantne celične strukture je QCA celica, ki je zgrajena iz štirih polprevodniških kvantnih pik. V vsaki celici sta dva elektrona, ki prehajata med kvantnimi pikami. Elektrona lahko prehajata le med sosednjimi kvantnimi pikami ter ven iz celice ne moreta uiti. Vsaka celica deluje kot avtomat dveh stabilnih stanj. Da je celica v stabilnem stanju morata biti elektrona fiksirana v kvantnih pikah ki sta med seboj najbolj oddaljeni, torej po diagonali. Prva celica spodaj je fiksirana na logično 1, druga pa na logično 0.



Razdalja med dvema kvantnima pikama je 20 nm, velikost pike pa 10nm. Urin cikel v QCA strukturah je razdeljen na štiri faze:

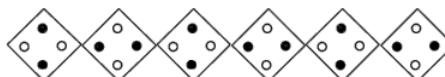
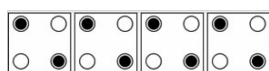
- **Faza preklopa (Switch):** Ob dvigovanju pregrad celice postanejo polarizirane.
- **Faza zadrževanja (Hold):** Pregrade so dvignjene in elektrona ne moreta prehajati ven iz svoje kvantne pike. Celice so v stabilnem stanju.
- **Faza sproščanja (Release):** Spuščanje pregrad. Elektrona lahko zopet začneta prehajati med pikami.

- **Faza sproščenosti (Relax):** Pregrade so spuščene in celice so v nevtralnem nepolariziranem stanju.

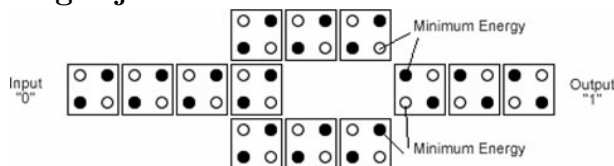
Ravno zaradi svoje majhnosti ter hitre ure so QCA strukture tako privlačne, saj lahko z manjšanjem vezij nadaljujemo tam, kjer bomo pri tranzistorjih končali.

Osnovne QCA strukture:

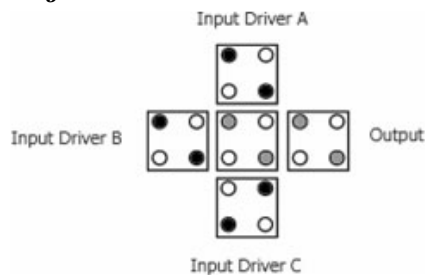
- **Žica 90° in 45°**



- **Negacija**



- **Majoritetna vrata AND in OR**



Poln nabor torej imamo. Iz polnega nabora izhajajo vse ostale QCA strukture, ki bodo v seminarski nalogi predstavljene.

Tabela 1: Trend razvoja CMOS tehnologije

Tehnološki proces	180 nm	130 nm	90 nm	65 nm	45 nm	32 nm	22 nm
Leto izdelave	1999	2001	2003	2005	2007	2009*	2011*
Efektivna dolžina vrat	130 nm	70 nm	50 nm	35 nm	25 nm	17 nm	12 nm
debelina 'gate' el. v atomih	10	8	5	5	5-10	5-10	5-10
št. vrat/ $mm^2$	160K	240K	480K	0.9M	1.5M	2.8M	4.5M
velikost SRAM celice ( $\mu m^2$ )	4.5	2.4	1.3	0.6	0.3	0.15	0.08

### 1.3 Predstavitev CMOS tehnologije

Complementary metal–oxide–semiconductor (CMOS) je tehnologija, ki se uporablja v praktično vseh integriranih vezjih. Osnovna logična vrata so vezje sestavljeno iz dveh komplementarnih MOSFET tranzistorjev (p in n tip kanala). Tehnologijo odlikujejo visoke hitrosti preklopov in relativno nizka poraba, saj moč troši le v primeru, ko pride do preklopa na tranzistorstem paru. Trenutno se v komercialno izdelujejo vezja v 90nm in 65nm tehnološkem procesu.

Skaliranje silicijevih tranzistorjev je bilo glavna vodilna sila, ki stoji za razvojem mikroelektronike v zadnjih 50 letih. Opazili smo, da z vsakim tehnološkim prehodom (izboljšavo fotolitografskih postopkov), se linearne dimenzije reducirajo približno za faktor 0.7 in površine za faktor 0.5. Tako je pri 90 nm tehnologiji gostota vezij približno 500000 logičnih vrat na  $mm^2$ , pri 65nm pa skoraj milijon!

Za primerjavo smo prikazali trend razvoja CMOS tehnologije od 180 nm (leta 1999) pa vse do 22 nm procesa (v proizvodnji bo leta 2011) v tabeli 1.

22 nanometerska tehnologija je že sedanost, saj je skupina proizvajalcev med katerimi so AMD, Freescale, IBM, STM v okviru skupnega projekta razvila in objavila prvo 22 nm klasično SRAM celico s šestimi tranzistorji. Širina celice je bila pičlih  $0.1 \mu m^2$ . Ključni mejnik v CMOS tehnologiji naj bi bila meja 16nm oz. 10nm. Po ocenah združenja ITRS naj bi 16 nm tehnologija prispela šele okoli leta 2018. Potrjeno je bilo, da se bo na tej meji zmanjševanje tranzistorja ustavilo zaradi kvantnega tuneliranja, ne glede na uporabljene materiale. Tako lahko sklepamo, da se bliža konec razvoja CMOS tehnologije.

## 2 Metode

Za pravilno primerjavo zmogljivosti potrebujemo ne samo ustrezno znanje in dokumentacijo ampak tudi ustrezno metodo, ki omogoča enakovredno primerjavo. Kajti nepravilna primerjava bi bila lahko zavajujoča, česar si seveda ne želimo. Naš cilj je karseda objektivno predstaviti razlike v performančnih sposobnosti obeh tehnologij. Tu se pri obeh tehnologijah znajdemo pred vprašanjem, kaj sploh želimo? Ali bo to platforma za visoko zmogljivo računanje, pomnjenje? Ali nas bolj zanimata poraba prostora in predvsem energetska učinkovitost? Zato je ključnega pomena postavitve metrik s katerimi lahko tehnologije primerjamo med seboj. Toda, poleg metrik, potrebujemo še meritve in izračune. Težave se pojavijo predvsem takrat, ko meritev ni na voljo ali so neprimerljive med seboj. Tedaj moramo prikazati druge (izpeljane) količine, ki so primerljive, kot so npr. skalabilnost in izkoriščenost prostora.

### 2.1 Orodja

Pri CMOS vezjih ni lahko se dokopati do konkretnih meritev in analiz vezij, ki trenutno nastopajo v najzmogljivejših mikroprocesorjih. Tudi modeli za nizkonivojske simulacije (na nivoju elektronike) so bodisi nedostopni bodisi prezahtevni ali premalo zreli in preizkušeni za praktično uporabo. Morda so še najprimernejši načini za izvajanje meritev in simulacij ravno višjenivojski modeli, ti so ponavadi vključeni v orodjih za optimizacijo in sintetizacijo logičnih vezij v polprevodniške strukture, ki jih ponujajo podjetja *Synopsys*, *Cadence*, *MentorGraphics*, ipd. Taka orodja niso samo pregrešno draga, ampak tudi dokaj zahtevna za uporabo, poleg tega je zelo težko ali celo nemogoče dobiti preizkusne različice. Naš namen ni bil preučiti delovanje le-teh, zato smo skušali dobiti orodja, ki nam čimenoostavneje sestavijo polprevodniško strukturo, to so taka orodja, katerim podamo datoteko v višjenivojskem HDL jeziku (običajno VHDL ali Verilog) in znajo zgenerirati vezja na najnižjem (polprevodniškem) nivoju. Nekatera orodja (polprofesionalna), ki smo jih uporabljali za sintetizacijo so *Alliance CAD system*, *Microwind*, za optimizacijo logičnih struktur *MGC Leonardo Spectrum*. Za izračune porabe energije in površine smo uporabljali okolje *Cadence InCyte*. Simulacije na nivoju električnih vezij smo opravljali s programom *Spice3*, vendar smo ta segment kmalu opustili, saj zahteva preveč časa in znanja.

Za izgradnjo CMOS vezij smo sintetizatorju podali kot opis strojne opreme



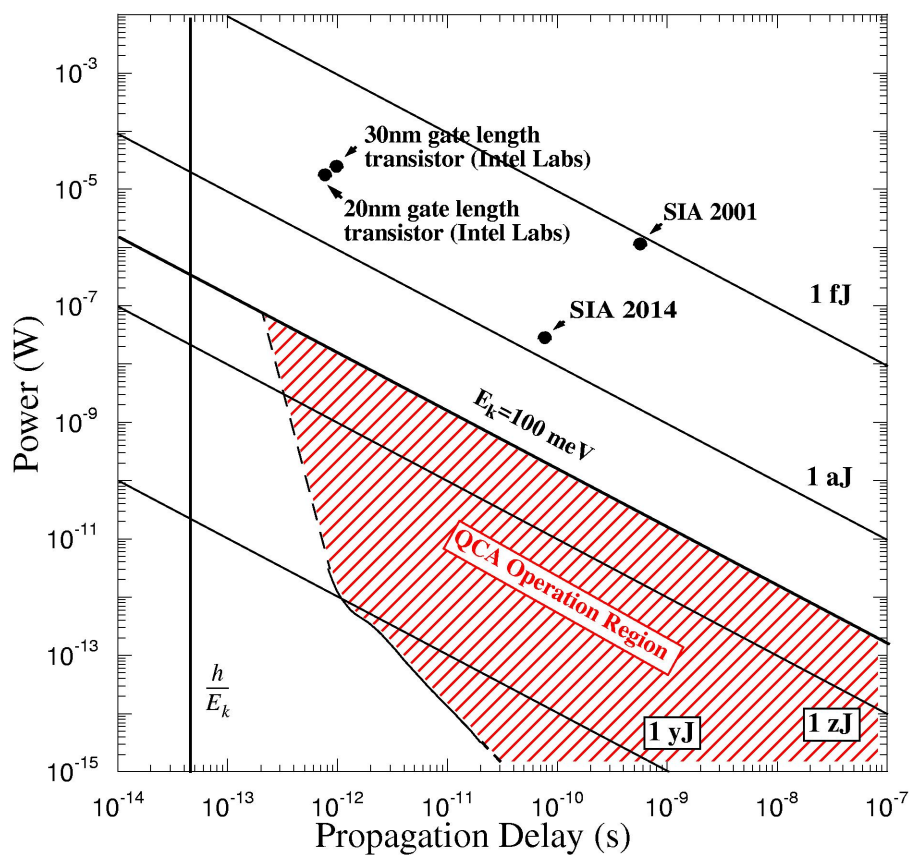
v VHDL jeziku, orodje za optimizacijo je izvršilo optimizacijo vezja in izpisalo ugotovitve analize. Podatke smo po optimizaciji dobili v poročilu, ti so vsebovali kritične poti (najdaljše zakasnitve) in velikost vezja (št. vrat, primitivov, povezav). Iz slednjih smo preračunali tudi površino in porabo energije z okoljem *InCyte*.

## 2.2 Poraba energije

Veliko je slišati o zelo dobrih lastnosti QCA tehnologije glede energetske učinkovitosti. Žal konkretni podatki zaenkrat ne kažejo takšno prednost v korist le-te, namreč potrebno je poudariti, da zaenkrat energetske učinkovite QCA strukture so brez sinhronizacije oz. adiabatnega preklopa neuporabne. Redko kdo se ubada s problemi sinhronizacije QCA struktur in iskanjem konkretne rešitve adiabatnega preklopa, strokovnjaki raje načrtujejo logične strukture in sestavljajo ‘QCA-lego’ kocke, problem sinhronizacije pa ostaja nerešen. Pri QCA strukturah smo uporabili model, ki predvideva da je disipacija moči enaka  $100 \text{ W/cm}^2$ , kar je precej groba in pesimistična ocena. Vendar boljšega modela za izračun disipacije žal ni na voljo. Zanesljiv vir [6] navaja, da se pri QCA vezjih disipacija moči povečuje s frekvenco adiabatnega preklapljanja. Podobno je pri CMOS tehnologiji, le da se tu disipacija meri v frekvenci preklopov na tranzistorjih in sami hitrosti preklopa, ki je seveda odvisna od napetosti. Tako ne moremo realno oceniti porabo energije, lahko pa upoštevamo najslabši primer, to je  $100 \text{ W/cm}^2$ , pri kateri lahko čipi še delujejo. Vir [6] navaja tudi, da adiabatno preklapljanje pri 1GHz še ne bi doseglo to mejo. Zato lahko sklepamo, da QCA vezja bi pri taki disipaciji lahko preklapljala vsaj 1GHz, verjetno tudi več. Bolj optimistični viri nakazujejo, da je poraba moči precej nižja (prikazano na grafu 1). Tako naj bi se poraba pri frekvenci 1 THz gibala v nano-wattnem območju. Vendar so ti viri nezanesljivi in nikjer ne poudarjajo, da gre za QCA strukture z adiabatnim preklpom na nivoju celice.

## 2.3 Zakasnitev vezij in frekvence delovanja

Najbolj pomembna količina za ovrednotenje CMOS vezij in QCA struktur je najbrž zakasnitev oz. hitrost delovanja. Žal je tudi najmanj primerljiva. Glavni problem je postaviti enotno metriko za pravilno primerjavo zakasnitev vezij. Seveda je problem toliko težji, če ne poznamo konkretnih (bodisi izmerjenih ali izračunanih) podatkov.



Slika 1: QCA: energetska poraba glede na hitrost preklapljanja

## 2.4 Površina vezij

Površino vezja je precej lažje dobiti, kot pa zakasnitve ali porabo. Za CMOS tehnologijo lahko uporabimo orodja, ki sintetizirajo vezja iz HDL jezikov, ta so najbolj primerna za pridobitev podatkov o površini vezij. Lahko pa se veliko podatkov dobi tudi iz literature. Za QCA vezja si lahko pri oceni površine pomagamo z orodjem *QCADesigner*, vendar veliko virov iz katerih črpamo vezja za primerjavo imajo natančno podano površino svojih vezij. Dilema se pojavi pri površini same celice. Skoraj vsa literatura navaja širino celice 60 nm (20 nm razdalje med kv. pikama znotraj celice), vendar je potrebno omeniti, da so v viru [6] navedli potrebno velikost celice 220 nm za normalno delovanje adiabatnega preklopa. Ker pa komercialno uveljavljenih rešitev tako ali tako ni, moramo pač vzeti tisti model celice, ki se najpogosteje

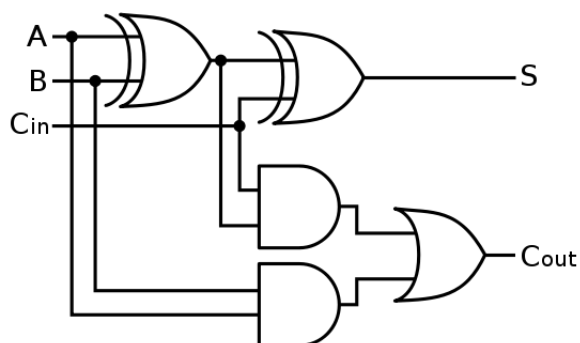
pojavlja. Tako so se naše meritve opirale na privzeto Lentovo velikost celice.

### 3 Seštevalnik

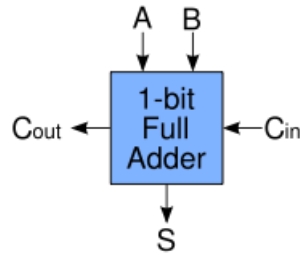
Je eno od osnovnih vezij, ki se uporablja v računalništvu. Implementacija se uporablja v preračunavanju naslovov (povečevanja programskega števca), kot del aritmetično logične enote. Z uporabo dvojiškega in eniškega komplementa števil, pa se da seštevalnike uporabiti tudi kot odštevalnike. Zaradi že tako težke primerjave dveh na oko podobnih tehnologij, se bomo omejili na seštevalnike. Za potrebe primerjave smo CMOS strukture generirali in optimizirali z orodji *AllianceCAD* in *LeonardoSpectrum* iz izvorne kode v HDL jeziku, QCA seštevalnike pa smo povzeli iz članka [2].

#### 3.1 Tipi seštevalnikov

Osnovni enobitni seštevalnik implementira formuli  $S = A \oplus B$  za vsoto, ter  $C_{out} = A \wedge B$  za izhodni prenos. To vezje pa ne upošteva možnega vhodnega prenosa. Zato formulo raje razširimo in realiziramo polni seštevalnik. Razlika med obema je da polni seštevalnik upošteva še dodatni vhodni prenos.

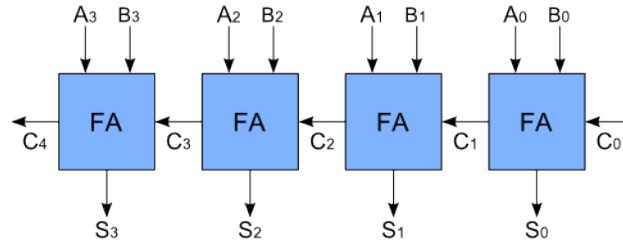


Slika 2: Diagram polnega seštevalnika



Slika 3: Blok shema 1-bitnega polnega seštevalnika

S kaskadno vezavo (eng. *RCA – Ripple Carry Adder*) več polnih seštevalnikov med seboj, pridemo do  $n$  bitnega seštevanja. Upoštevati je potrebno le, da se za prvi vhodni prenos postavi vrednost bita 0 na 1, ali pa da namesto polnega seštevalnika uporabimo navadni seštevalnik, ki ne upošteva vhodnega prenosa. Vsi nadaljni seštevalniki pa imajo na vhodni prenos vezan predhodni izhodni prenos. Slika prikazuje primer 4-bitnega kaskadnega seštevalnika.



Slika 4: Blok shema 4-bitnega polnega seštevalnika

Iz sheme 4-bitnega polnega seštevalnika je vidno, da kritično pot predstavlja izračun izhodnega prenosa. Saj potrebujemo za izračun iz  $C_0$  do  $C_4$  tri izračune za prenos (pri 32-bitnem seštevalniku pa 31 izračunov). Za zmanjšanje tega časa, pa se je razvilo seštevalnike s predikcijskim izračunom prenosa (eng. *CLA – Carry Lookahead adder*).

Ti generirajo dva signala  $P$  (propagate – predpostavi prenos na naslednjem mestu) in  $G$  (generate – prenos je). Signal  $P$  je v bistvu vsota navadnega seštevalnika,  $G$  pa prenos, za vsako bitno mesto. Signala nam povesta kako se bo prenos obnašal pri seštevanju dveh bitov (izračunamo ga v naprej).

Ko imamo za vsa bitna mesta izračunana P in G signala, smo izračunali v enem koraku (se izvede paralelno) vse vhodne prenose za bitna seštevanja na vseh mestih. Realizacija samega seštevalnika se pri več bitih razbije v sklope po več bitov skupaj in ne na bitnem nivoju, kar zmanjša redundanco vezja (skupaj se vzame 4,8,16 bitov)

Za še hitrejši izračun pa so bili razviti pogojni seštevalniki (eng. CSA – Conditional Sum Adder). Tu seštevanje n-bitne besede razbijemo na dve polovici, spodnjo in zgornjo. Spodnja polovica izračuna vsoto n/2 bitov. Zgornja pa je podvojena, saj izračunava dve vrednosti. Prva upošteva prenos 0 iz spodnje polovice, druga pa upošteva prenos vrednosti 1. Tako smo n-bitno besedo izračunali v polovičnem času. To razbitje se lahko nadaljuje na četrtine, osmine,... vse do bitnega nivoja. To izračunavanje pomeni veliko večjo strukturo od osnovne, saj se zgornja polovica izračuna s predikcijo, kar pomeni dvojno strukturo.

## 3.2 QCA implementacije

### 3.2.1 RCA seštevalnik

Je med tremi implementacijami najbolj enostaven, majhen a počasen. Avtorji članka so zaradi lažje primerjave z ostalima seštevalnikoma razširili implementacijo na cevovodno strukturo. To so naredili s sinhronizacijo vhoda in izhoda, kar je doprineslo dodatne QCA celice zaradi povezovalnih linij. Slika 5 prikazuje 4-bitno realizacijo. S podvajanjem pridemo do realizacije z 8-bitni, 16-bitni, ipd. Realizacija polnega seštevalnika v QCA, poteka z uporabo majoritetnih vrat. Tako pridemo do izraza:

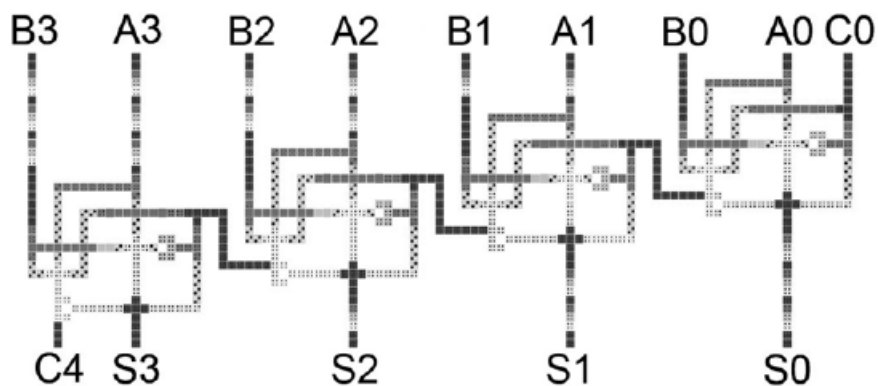
$$S_i = M(\overline{M}(A_i, B_i, C_i), M(A_i, B_i, \overline{C_i})) \quad (1)$$

$$C_{i+1} = M(A_i, B_i, C_i) \quad (2)$$

Slika 5 prikazuje 4-bitni RCA seštevalnik z večnivojskim križanjem linij. Polni seštevalnik potrebuje en urin cikel za izračun vsote in prenosa. Zaradi sinhronizacijskih linij je za izračun n-bitne vsote potrebnih n urin ciklov.

### 3.2.2 CLA seštevalnik

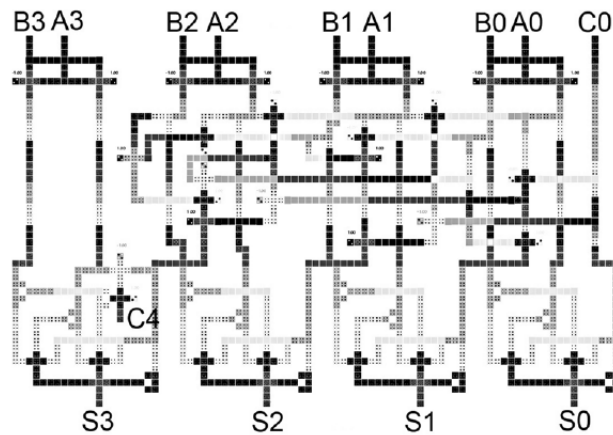
Z računanjem prenosa v naprej pridobimo večjo hitros pri še vedno sorazmerno normalni kompleksnosti vezja. Dizajn, ki so ga realizirali avtorji upošteva 4-bitne rezine za preračunavanje P in G signalov, oziroma velikost



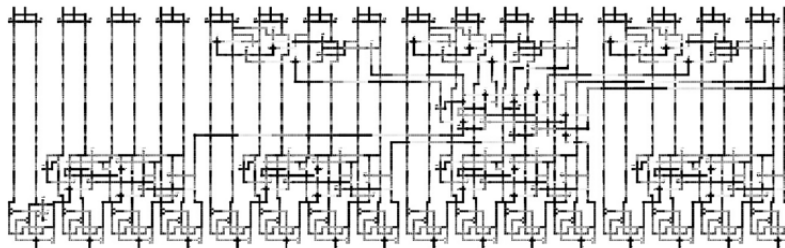
Slika 5: QCA vezje 4-bitnega RCA seštevalnika

PG bloka. Tako za vsake nadaljne 4 bite potrebujemo nov blok (kar nam dodatno doprinese k velikosti vezja). Ta realizacija se izogiba povratnega prenosa signala, ki ga uporabljajo CMOS vezja. Zaradi adiabatsnosti QCA je vezje cevodno. Bloki generirajo vse prenose paralelno za vse bloke. Ti se prenašajo v naslednje višje nivoje PG blokov. Izhodi iz CLA blokov in PG blokov so potrebni za končni izračun vsote na vsaki bitni poziciji. Zaradi cevovodnega izvajanja so vsi biti na voljo v isti urini periodi. Z upoštevanjem P, G ter vhodnih prenosov se izračuna končna suma z le tremi majoritenimi vrati za bit.

$$S_i = M(M(\overline{P}_i, G_i, C_i), M(P_i, \overline{G}_i, C_i), \overline{C}_i) \quad (3)$$



Slika 6: QCA vezje 4-bitnega CLA seštevalnika



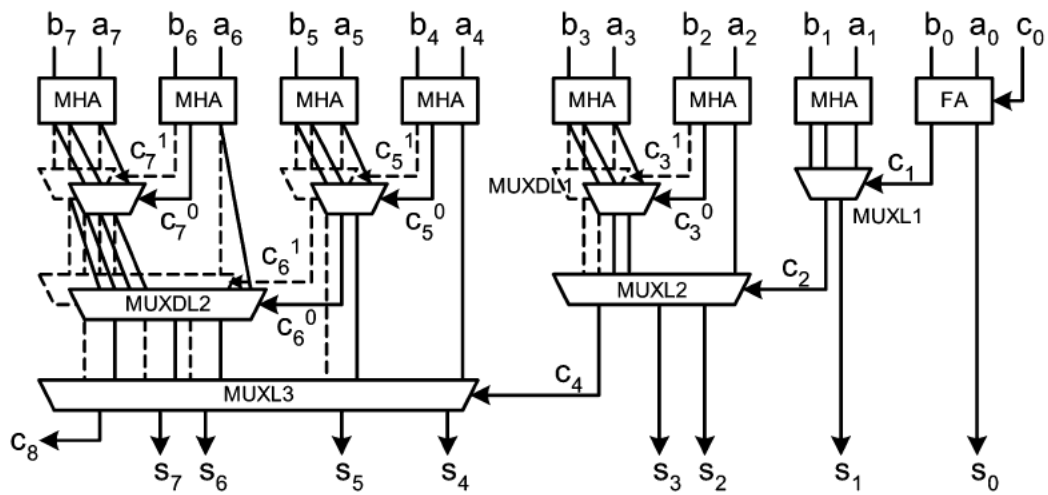
Slika 7: QCA vezje 16-bitnega CLA seštevalnika

### 3.2.3 CSA seštevalnik

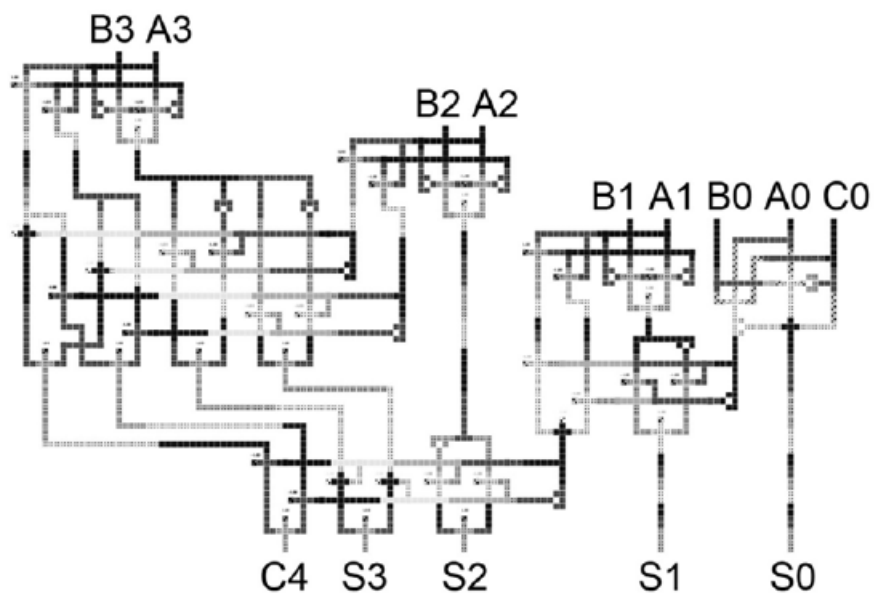
Tako v QCA, kot v CMOS tehnologiji je bil razvit pogojni seštevalnik za najhitrejše računanje vsote. Je pa najbolj kompleksno vezje od vseh, saj je veliko redundantnega računanja zaradi predikcije/pohitritve.

Realizacija je veliko bolj kompleksna, saj vsebuje elemente polnih seštevalnikov (FA), modificiranih seštevalnikov (MHA), multiplekserjev (MUX), in dupliciranih multiplekserjev (MUXD). Blok diagram 8-bitnega pogojnega seštevalnika je prikazan na sliki 8. Pod njim pa sta na slikah 9 in 10 prikazani še primerjavi 4 in 16 bitnega pogojnega seštevalnika v QCA realizaciji.

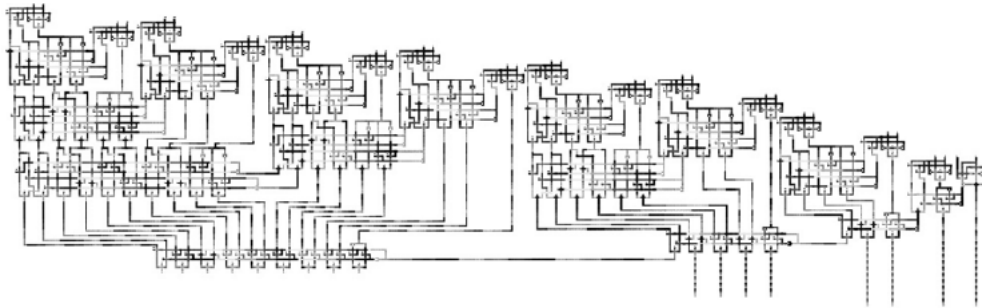




Slika 8: Blok shema 8-bitnega CSA seštevalnika



Slika 9: QCA vezje 4-bitnega CSA seštevalnika

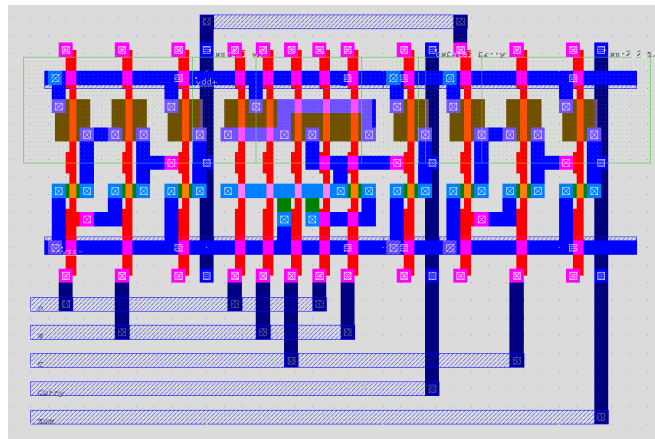


Slika 10: QCA vezje 16-bitnega CSA seštevalnika

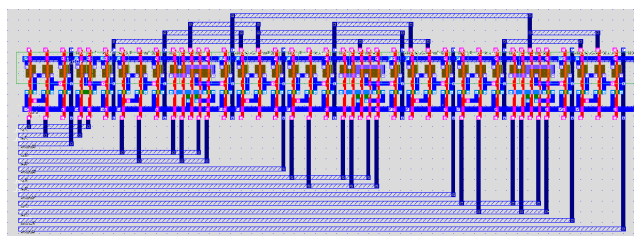
### 3.3 CMOS implementacija

QCA strukture so plod izdelave in testiranja avtorjev [?]chlo2006. Da bi lahko primerjali na nivoju seštevalnikov bi potrebovali izmerjene rezultate za CMOS realizacije. Le te smo pridobili s programskimi orodji.

Realizacijo smo pogojili glede na optimiziran CMOS polni seštevalnik v različnih tehnologijah, da bi se prostorsko in hitrostno čimbolj približali teoretičnim zmožnostim QCA tehnologije. Spodnja slika predstavlja sliko konkretnega vezja za 1 oz. 4 bitni polni seštevalnik.



Slika 11: CMOS polprevodniška slika 1-bitnega seštevalnika



Slika 12: CMOS polprevodniška slika 4-bitnega seštevalnika

### 3.4 Primerjava

Sama primerjava seštevalnika na nivoju tehnologije izdelave je precej težavna, saj sta si tehnologiji podobni bolj na 'oko'. Teoretična zasnova, ki jo obe tehnologiji implementirata je resda enaka in tudi ideje implementacije so podobne (strukture za operacije, povezave,...), toda v tehnološki izvedbi se precej razlikujeta. Prvi pogled primerjave je iz vidika površine, ki jo vezja zavzemajo.

Tabela 2: Površina vezja v  $\mu m^2$

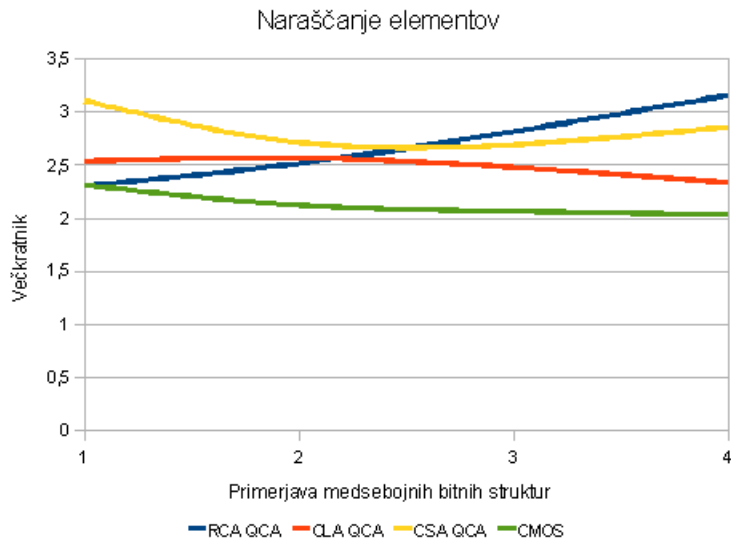
vezje		QCA			CMOS			
		RCA	CLA	CSA	Število vrat	130nm	90nm	65nm
4bit	[celic]	651	1575	1999	54			
	$[\mu m^2]$	1,20	1,90	4,44		339	225	100
8bit	[celic]	1499	3988	6216	125			
	$[\mu m^2]$	3,57	5,53	15,46		741	408	256
16bit	[celic]	3771	10217	16866	265			
	$[\mu m^2]$	11,78	15,51	48,46		1601	961	529
32bit	[celic]	10619	25308	45354	546			
	$[\mu m^2]$	42,23	42,88	158,38		3249	1954	968
64bit	[celic]	33531	59030	129611	1107			
	$[\mu m^2]$	159,22	105,18	551,65		6577	4007	2025

QCA celice so fiksne velikosti in k površini doprinese le število uporabljenih celic - aktivna površina. Če gledamo površino kot mrežo, ki ni nujno

popolnoma zapolnjena s celicami, pa je površina še večja. CMOS tehnologija pa strukturo celic zamenja z strukturo polprevodnika in kovin, ki opravljajo nalogo povezovalnih poti.

Primerjava samega števila celic in števila vrat med seboj je nesmiselna, saj je QCA celica veliko manjši element kot pa vrata CMOS tehnologije. Lahko pa primerjamo trende naraščanja na bitno strukturo, ki jo ena od tehnologij implementira.

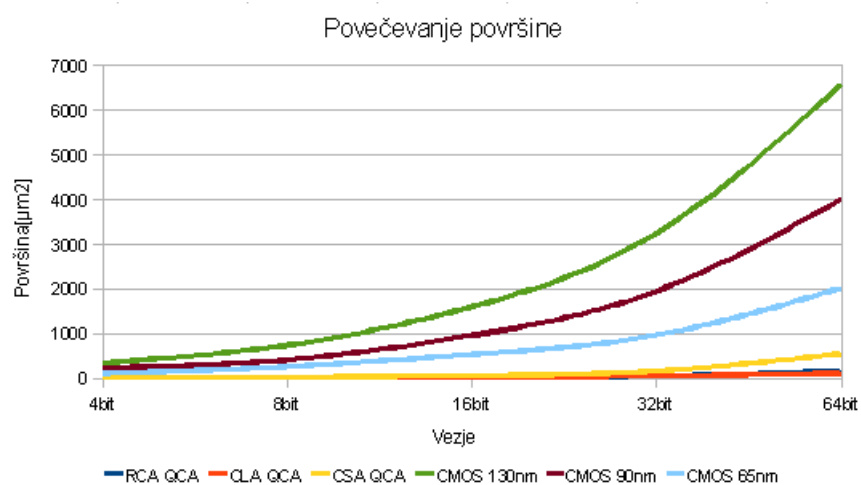
Naredili smo primerjavo sosednjih struktur, glede na število seštevanih bitov, po številu uporabljenih elementov. Tako da, na x osi točka 2 predstavlja kolikokrat je 8 bitna struktura manjša od 16 bitne, po številu uporabljenih elementov. Iz tega grafa je razvidno, da RCA seštevanik linearno narašča, kar je pričakovano, saj se z večanjem samo poveča struktura za dodano kopirano strukturo. Neglede na realizacijo seštevalnika, pa je CMOS edini, ki je vztrajno padal.



Slika 13: Primerjava elementov medseboj glede na naraščanje elementov.

V tabeli 2 je zapisana velikost površine, ki jo zasedajo strukture. S programom za izračun površine smo lahko testirali strukture v 60 nanometrski tehnologiji, saj manjših ni poznal, oziroma nam v preizkusni različici niso bile na voljo.

Zanimivo je, da so najmanjše (4 bitne) QCA strukture od največjih (64 bitne) manjše za faktor 100, le pri CLA je to okoli 50. Pri CMOS pa je ta faktor dosti manjši in je reda 20. Iz tega se vidi, da je CMOS vsaj glede velikosti struktur precej varčnejši s prostornostjo kot QCA. Tu ne smemo zanemariti, da je CMOS že stara tehnologija in so postopki izdelave že zelo optimizirani, kar verjetno pomaga k čimboljši izkoriščenosti prostora. QCA deluje na medsebojne vplive celic, kar pomeni, da struktur ne moremo vedno prostorsko optimizirati, saj bi to pripeljalo do motenj med celicami.



Slika 14: Primerjav površin

Naslednji vidik, ki smo ga primerjali je bila poraba vezij.

Tabela 3: Ocena porabe moči v  $mW$  oz. 'leakage-a' v  $\mu W$ 

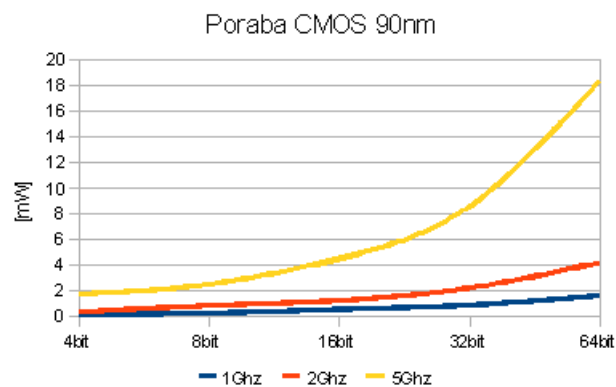
Freq/leakage	90nm [ $mW$ ]				65nm [ $mW$ ]			
	1Ghz	2Ghz	5Ghz	leakage	1Ghz	2Ghz	5Ghz	leakage
4bit	0,14	0,29	1,7	1,1 $\mu W$	0,12	0,22	0,56	0,7 $\mu W$
8bit	0,20	0,80	2,48	2,3 $\mu W$	0,16	0,32	1,5	1,6 $\mu W$
16bit	0,51	1,2	4,5	4,7 $\mu W$	0,26	0,82	3,2	3,5 $\mu W$
32bit	0,84	2,2	8,6	9,6 $\mu W$	0,61	1,5	5,8	7,0 $\mu W$
64bit	1,6	4,2	18,4	19,5 $\mu W$	1,2	3,0	12,4	14,5 $\mu W$

Tabela 4: Ocena porabe moči v  $\mu W$  za QCA

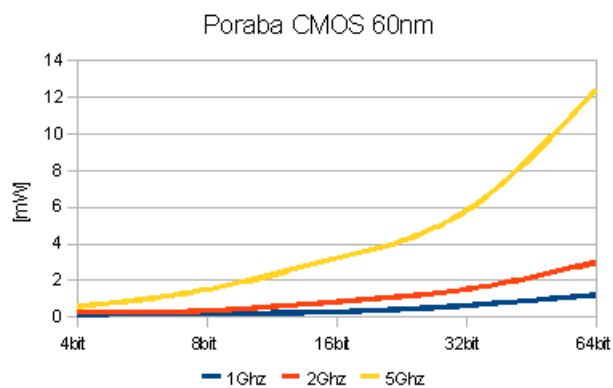
vezje	QCA		
	RCA	CLA	CSA
4bit	1,20	1,90	4,44
8bit	3,57	5,53	15,46
16bit	11,78	15,51	48,46
32bit	42,23	42,88	158,38
64bit	159,22	105,18	551,65

Za CMOS vezja so izbrane tri različne frekvence, saj je poraba odvisna od hitrosti preklopa števila vrat in ne od površine. Poleg tega je v tabeli zapisana še vrednost porabe v stanju mirovanja(leakage). Primerjava nam ni uspela najbolje, saj imata obe tehnologiji porabo odvisno od velikosti vezja. Pri QCA je poraba odvisna tudi od območij, ki jih kontroliramo za adiabatne preklope, pri CMOS tehnologiji pa je poraba odvisna tudi od željene hitrosti vezja, saj je za hitrejša preklope potrebno več energije. Ker simulator ni bil sposoben simulirati tehnologije CMOS na velikostnem razredu 45nm in manj, primerjava vezij z vidika porabe ni zelo uspešna. Pričakovali smo, da bo poraba CMOS vezij večja od QCA, saj QCA za medsebojno vplivanje ne porabi veliko energije. Zanimivo pa je videti, da največja QCA struktura porabi toliko energije, kot porabijo najmanjše strukture 60nm CMOS tehnologije.

Zadnja tabela predstavlja zakasnitve skozi vezja. Za QCA strukture v ciklih, za CMOS pa je zapisana zakasnitev v ns za kritično pot skozi vezje v

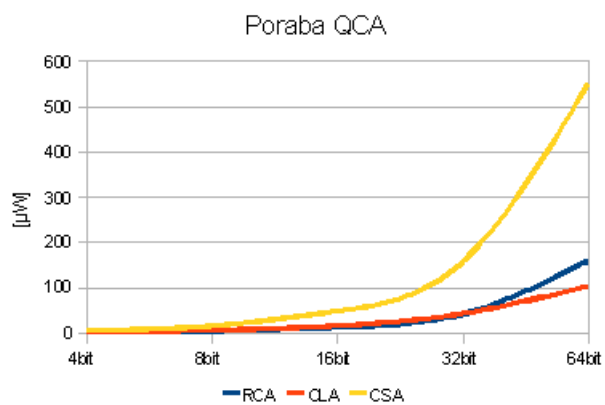


Slika 15: Poraba CMOS vezij v 90nm tehnologiji v mili W



Slika 16: Poraba CMOS vezij v 60nm tehnologiji v mili W

0.5 mikronski tehnologiji, ki je že globoko zastarela. Zato primerjave na tem delu ne moremo opraviti, ker sta metrika čisto drugačni. Tudi avtorji članka so se tej primerjavi odpovedali, saj ni točnega podatka na kateri frekvenco bi QCA strukture še stabilno delovale.



Slika 17: Poraba QCA vezij v mikro W

Tabela 5: Zakasnitve QCA vezij

vezje	QCA			CMOS
	RCA	CLA	CSA	
enota	cikel	cikel	cikel	ns
4bit	4,25	3,5	3,75	2,04
8bit	8,25	6,5	7,75	3,76
16bit	16,25	10,25	14	7,21
32bit	32,25	19	25	14,11
64bit	64,25	31,5	45	27,92



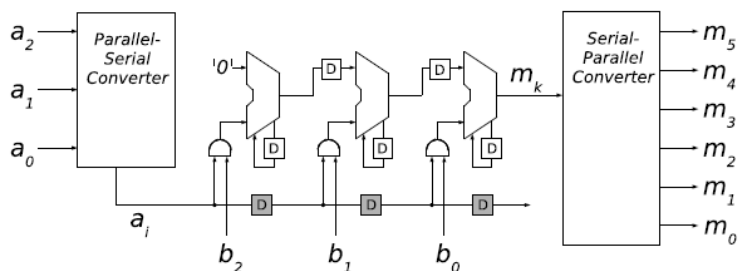
## 4 Množilnik

V tem poglavju bomo naredili primerjavo med serijsko-paralelnim množilnikom realiziranim kot CMOS vezje ter množilnikom kot QCA strukturo. Predstavili bomo samo izvedbo množilnika ter primerjali kakšen prostor zaseda njegova realizacija, zakasnitev ter porabo energije.

### 4.1 Predstavitev serijsko-paralelnega množilnika

Najprej opišimo, kaj serijsko-paralelni množilnik sploh je. Serijsko-paralelni množilnik je aritmetično vezje, katerega delovanje najbolj spomnija množenju na 'peš'. Biti njegovih operandov se delijo na serijske in paralelne. Z  $a_i$  označimo bite serijskega operanda ter z  $b_j$  bite paralelnega. Biti  $m_i$  pa so rezultat množenja.

Na sliki 18 je prikazana logična shema 3 bitnega serijsko-paralelnega množilnika.

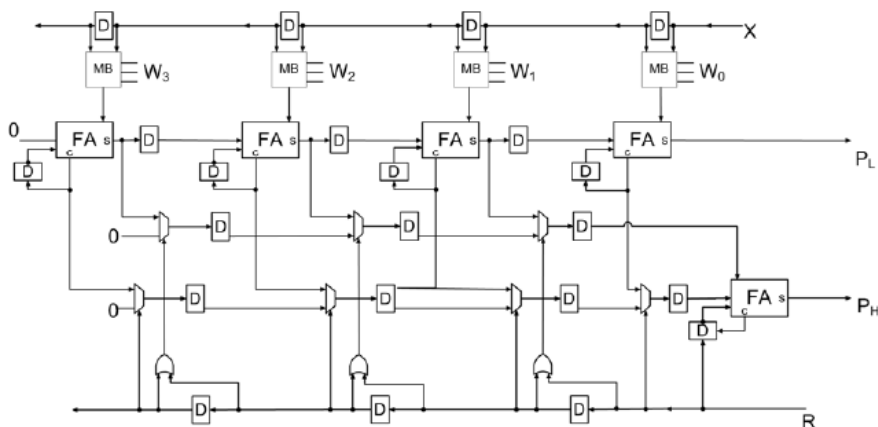


Slika 18: Serijsko paralelni množilnik

Vhod serijskih bitov je sicer paralelen, vendar vsebuje paralelno-serijski pretvornik kar omogoči, da biti  $a_i$  v množilnik vstopajo serijsko. Na izhodu iz množilnika pa je serijsko-paralelni pretvornik, tako da je izhod iz množilnika paralelen. Vmes so 3 množilne celice. Če množilne celice pobliže pogledamo, opazimo da delujejo tako, da v polni seštevalnik ki je v vsaki celici, vstopa rezultat AND operacije med  $a_i$  ter  $b_j$ , rezultat prejšnje množilne celice ter prenos prejšnje operacije te množilne celice. Podobno kot če bi računali na papir, se rezultat vsakega cikla množenja  $i$  prišteje naslednjemu.

## 4.2 CMOS izvedba

Na sliki je prikazan 4 bitni serijsko paralelni množilnik v CMOS tehnologiji, kjer je  $X$  serijski vhodni faktor v množilnik, biti  $W_{0-3}$  pa so biti paralelnega faktorja. Izhoda iz množilnika sta  $P_L$  ter  $P_H$ , kjer je  $P_L$  del produkta z najmanjšo težo (*angl.* least significant part - LSP) ter  $P_H$  del z največjo težo (*angl.* most significant part - MSP).



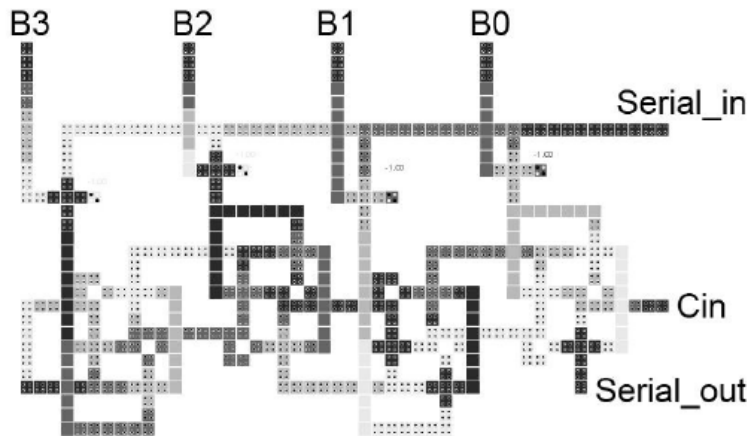
Slika 19: Serijsko paralelni množilnik v CMOS

## 4.3 QCA izvedba

Na sliki je realizacija 4 bitnega serijsko paralelnega množilnika v QCA, ki je bila sestavljena ter testirana s QCADesignerjem. Velikost posamezne celice je 20nm med dvema kvantnima pikama. 4 bitni množilnik na sliki 20 vsebuje 507 celic. Na sliki vidimo vhode  $b_j$ , pri označbi 'Serial\_in' pa v množilnik serijsko vstopajo biti  $a_i$ . Vhoda  $a_i$  in  $b_j$  prideta na vhod AND majoritetnih vrat, katerih izhod vstopa v polni seštevalnik posamezne množilne celice, kot je to predstavljeno tudi v logični shemi - slika 18. Celotno delovanje obeh (CMOS in QCA) množilnikov je torej precej podobno.

## 4.4 Primerjava zasedenega prostora

Velikosti celotnih QCA struktur ter primerjave z ekvivalenti CMOS tehnologije so podani v tabeli 6. Stolpci za CMOS tehnologijo so poimenovani 'CMOS65-130' kar označuje 65 do 130 nanometrsko tehnologijo.



Slika 20: Serijsko paralelni množilnik v QCA

Tabela 6: Poraba prostora

Št. bitov	QCA [ $\mu\text{m}^2$ ]	CMOS130 [ $\mu\text{m}^2$ ]	CMOS90 [ $\mu\text{m}^2$ ]	CMOS65 [ $\mu\text{m}^2$ ]
4	$1.04 \times 0.61$	$38.50 \times 38.50$	$29.31 \times 29.31$	$22.20 \times 22.20$
8	$1.93 \times 0.61$	$73.00 \times 73.00$	$54.86 \times 54.86$	$41.80 \times 41.80$
16	$3.67 \times 0.61$	$143.31 \times 143.31$	$107.76 \times 107.76$	$81.80 \times 81.80$
32	$7.24 \times 0.67$	$293.49 \times 293.49$	$220.50 \times 220.50$	$169.54 \times 169.54$

## 4.5 Primerjava zakasnitev

Za posamezne velikosti vhodnih faktorjev v množilnik, so v tabeli 7 predstavljene zakasnitve za QCA vezja kot tudi za CMOS. Stolpca ‘% povečanja’ predstavljata faktor relativnega povečanja zakasnitve glede na zakasnitev prejšnjega tj. red velikosti manjšega vezja.

Tabela 7: Primerjava zakasnitev

Št. bitov	QCA [cikli]	% povečanja	CMOS05 $\mu$ [ns]	% povečanja
4	8	-	5.31	-
8	16	100	10.42	96.23
16	32	100	20.32	95
32	64	100	39.37	93.75

Iz tabele je jasno, da se zakasnitev pri QCA vezjih povečuje sorazmerno z velikostjo oziroma kompleksnostjo vezja, saj za N-bitno množenje porabi 2N ciklov. Pri CMOS vezjih je povečanje zakasnitve malenkost manjše. Kar nakazuje, da je pri kompleksnejših vezjih mogoče uvajati logične in strukturne optimizacije, ki jih pri manjših vezjih ne moremo - te optimizacije so plod orodja LeonardoSpectrum. Takega orodja za optimizacijo QCA vezij trenutno nimamo, zato v tem primeru ocena relativne zakasnitve delno neupravičeno favorizira CMOS tehnologijo.

## 4.6 Primerjava porabe energije

Tabela 8: Poraba energije

Št.bitov/Ura	QCA[ $\mu$ W]	CMOS130[mW]			CMOS90[mW]			CMOS65[mW]		
		1GHz	2GHz	5GHz	1GHz	2GHz	5GHz	1GHz	2GHz	5GHz
4	0.63	1.29	3.59	14.04	0.50	1.209	4.50	0.24	0.80	2.80
8	1.17	3.72	9.97	47.67	1.35	3.5	15.65	1.04	2.56	9.72
16	2.23	14.31	39.27	177.22	5.08	13.32	57.14	3.79	9.73	38.50
32	4.85	61.13	166.74	757.55	21.36	55.77	240.62	16.67	41.27	166.33

V tabeli 8 je narejena primerjava porabe energije med CMOS ter QCA množilnimi vezji. Za QCA smo uporabili ocenjeno disipacijo  $100 W/cm^2$ . Glede na to, da je poraba energije pri CMOS vezjih odvisna tudi od frekvence ure, je pri CMOS vezjih upoštevana ocenjena poraba pri frekvencah 1GHz, 2GHz ter 5GHz. Dejanska poraba QCA vezij trenutno še ni znana, saj kvantni celični avtomati zaenkrat temeljijo le na teoretični podlagi.

## 5 Pomnilnik

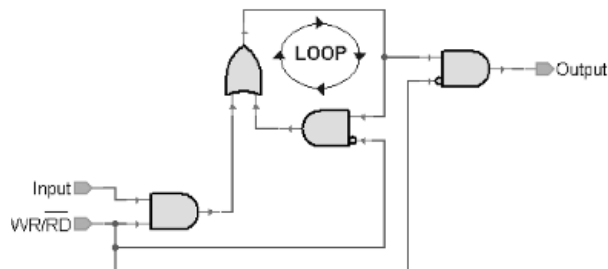
QCA ima tako zelo drugačno zasnovo od CMOS tehnologije, da je enostavna preslikava CMOS logike in vezja v QCA tehnologijo zelo neučinkovita. Zato so se pojavile potrebe po učinkovitem načinu hranjenja informacij v QCA tehnologiji. Da lahko dosežemo veliko gostoto vezja in visoke zmogljivosti moramo izkoristiti QCA sistem "procesiranja v žici". Pri temu sistemu hranimo informacijo v potujočem signalu med QCA celicami.

Celice lahko razporedimo v polje spominskih razdelkov. Predlagani so bili naslednji večji načini razporeditve razdelkov:

- Paralelna arhitektura - razporeditev je podobna kot pri CMOS RAM in vsak razdelek vsebuje zanko z enim bitom informacije
- Serijska arhitektura - zasnovan na pomikalnem registru - biti informacij so dostopni zaporedno
- Drevesna H arhitektura - spominsko polje sestoji iz majhnih spiral. Vsaka spirala vsebuje eno besedo in so razporejeni v rekurzivno drevesno strukturo
- Hibridna arhitektura - paralelno branje in serijsko pisanje

### 5.1 Paralelna arhitektura

Paralelna arhitektura je najbolj podobna tradicionalni CMOS spominski arhitekturi. Arhitektura osnovne spominske celice je prikazan na sliki 5.1. Podatkovni bit je shranjen v zanki. Zanka mora biti implementirana v vseh fazah adiabatnega preklopa kar omogoča 'potovanje' shranjenega podatkovnega bita. Več takih osnovnih celic lahko povežemo v večjo organizirano enoto spomina. Prednost, ki jo ta arhitektura prinaša je hkratna dostopnost vseh celic za branje/pisanje.



Slika 21: Osnovna celica paralelne arhitekture

## 5.2 Serijska arhitektura

Tudi serijska arhitektura temelji na zankah, toda s to razliko, da so tu zanke »raztegnjene« in lahko shranijo več kot samo 1 bit informacije. Pri tem moramo poskrbeti še za kontrolno logiko, da so biti sinhronizirani in da jih znamo naslavlјati. Ta arhitektura nam omogoča precejšen prihranek prostora v primerjavi s paralelno arhitekturo, saj hranimo v eni zanki več bitov. Po drugi strani pa imamo večjo zakasnitev pri bralno/pisalnih operacijah. Ker lahko dostopamo samo do najbolj zunanjega bita, zakasnitev narašča s številom shranjenih bitov eni zanki.

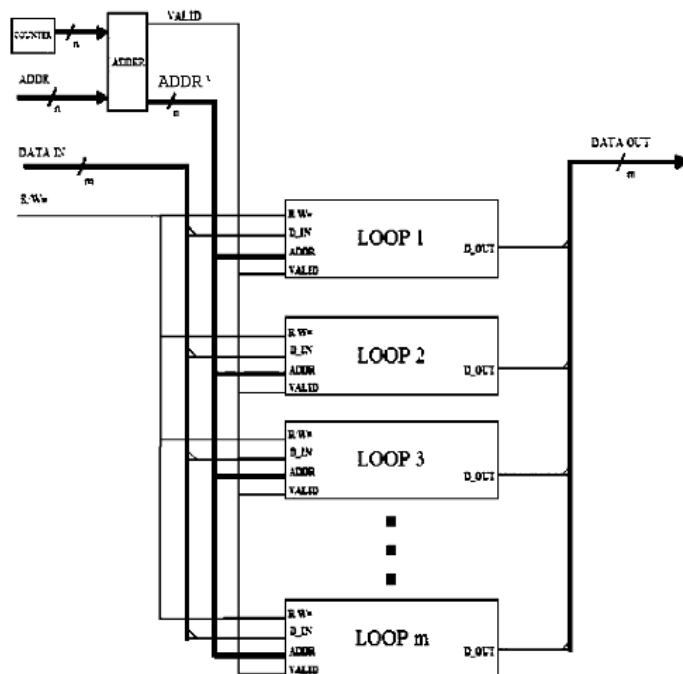
## 5.3 Drevesna H-arhitektura

Velika prednost pri implementaciji drevesne H arhitekture je logika za dekodiranje naslovov. Toda ravno ta logika je lahko problematična pri QCA pomnilniku z veliko gostoto. Uporablja rekurzivno H strukturo in je načrtovana tako, da ima poti enakih dolžin in urejena območja urinih faz. Ta struktura ima velike zakasnitve v primerjavi s serijsko arhitekturo. Prav tako ta arhitektura vključuje tudi naslovne tehnike, ki bazirajo na izmeničnih paketih podatkov in naslovov. Zaradi tega je potreben drugačen pristop pri načrtovanju takega tipa pomnilnika.

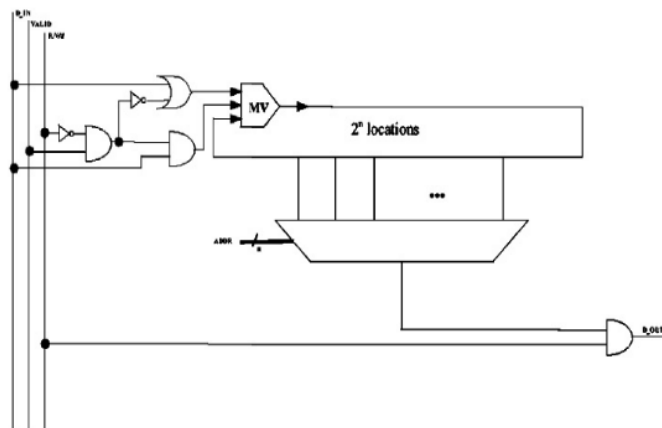
## 5.4 Hibridna arhitektura

To arhitekturo lahko obravnavamo kot evolucijo serijske arhitekture. Uporablja serijsko pisanje in paralelno branje podatkov. Ta lastnost omogoča kombiniranje kratkih zakasnitev paralelne arhitekture in velike gostote celic

serijske arhitekture. Blokovni diagram te arhitekture je prikazan na sliki 5.4. V tej strukturi je  $m$  zank po  $2n = N$  bitov, ki so urejeni tako, da tvorijo  $m$ -bitno besedo na  $N$  lokacijah. Dostopamo lahko sočasno do celotne  $m$ -bitne besede. Logična struktura znotraj posamezne zanke je predstavljena na sliki 23.



Slika 22: Blokovni diagram hibridnega QCA pomnilnika



Slika 23: Diagram implementacija zanke hibridnega pomnilnika

## 5.5 Primerjave QCA pomnilnikov

### 5.5.1 Dostopni čas

Najhitrejšje dostopne čase za branje in pisanje ima paralelni pomnilnik. Temu sledi hibridni pomnilnik, ki je pri branju enakovreden paralelnemu in pri pisanju serijskemu pomnilniku z enako dolžino zanke. Pri hibridnem pomnilniku je zakasnitev pri branju konstantna, zakasnitev pri pisanju pa linearno narašča s dolžino zanke. Najdaljše dostopne čase za branje in pisanje imata serijski in hibridni pomnilnik.

### 5.5.2 Površina

Uporabno površino računamo po enačbi:

$$A = (d \cdot n_x)(d \cdot n_y) = A_u + A_{un}$$

pri čemer so:

$d$  – dolžina stranice QCA celice

$n_x, n_y$  – število celic po dimenziji  $x$  in  $y$

$A_u, A_{un}$  – delež uporabljenih in neuporabljenih celic



Kot osnovno primerjavo bomo vzeli 1 x 4 in 4 x 4 strukturo pomnilnika. Pri 1 x 4 strukturi lahko primerjamo paralelno in hibridno arhitekturo, pri 4 x 4 strukturi pa lahko primerjamo še drevesno H strukturo.

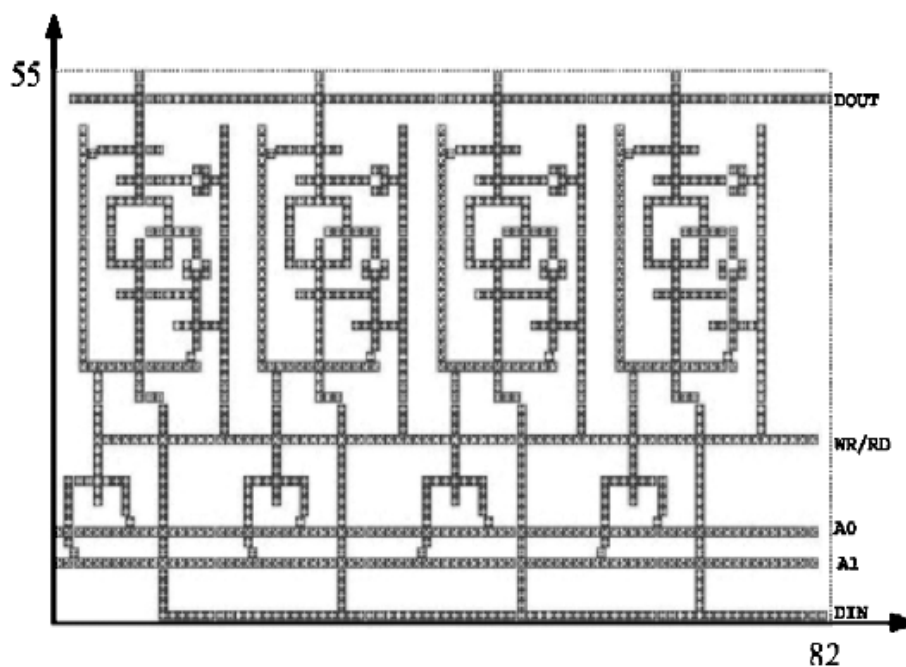
### Strukture: 1 x 4 biti

- **Paralelna struktura**

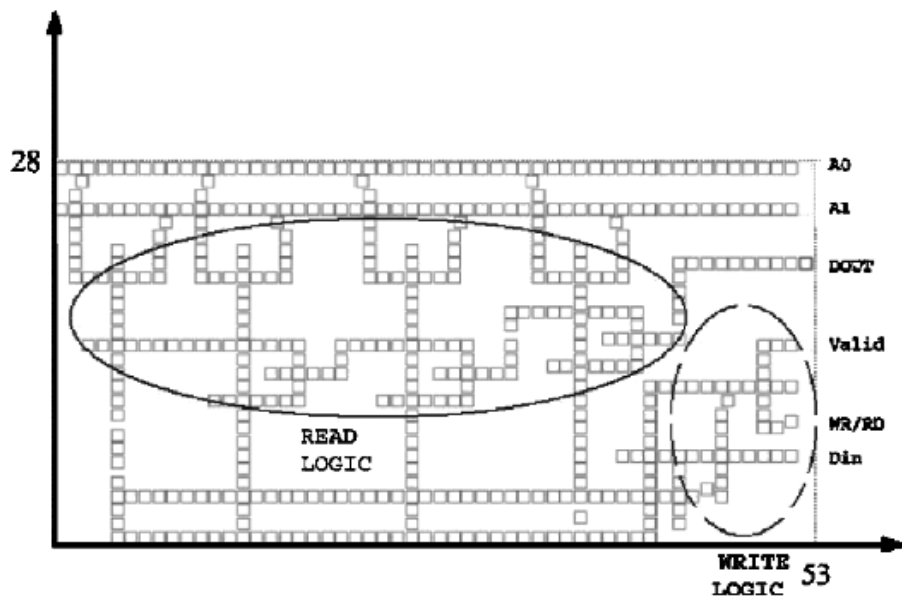
Dimenzija efektivne površine je 82 in 55 QCA celic. Efektivna površina je  $A = 0.455 \mu m^2$ ,  $d = 10 nm$ . Velik del zavzema naslovna logika. Slika 24.

- **Hibridna struktura**

Kot je razvidno iz slike 25 je efektivna površina dimenzije 53 in 28 QCA celic. Torej zavzema hibridna struktura samo 32% efektivne površine paralelne strukture. Pomanjšanje je možno na račun naslovne logike, ki je deljena za vse 4 bite in realizacije na manjšem področju.



Slika 24: 1 x 4 paralelni pomnilnik



Slika 25: 1 x 4 hibridni pomnilnik

**Struktura: 4 x 4 biti**

- **Paralelna struktura**

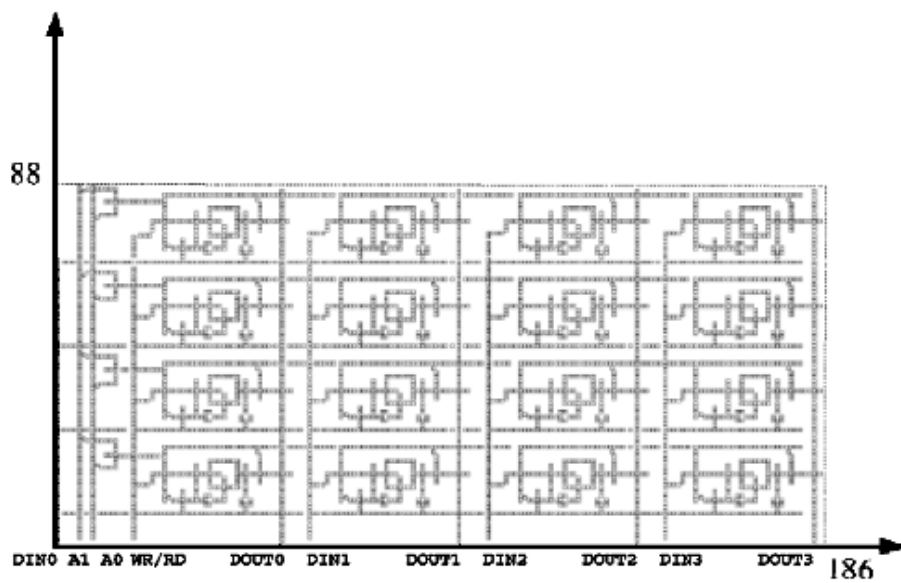
Kot je razvidno iz slike 26 je efektivna površina 88 in 186 celic,  $A = 1.63 \mu m^2$ ,  $d = 10 \text{ nm}$ .

- **Hibridna struktura**

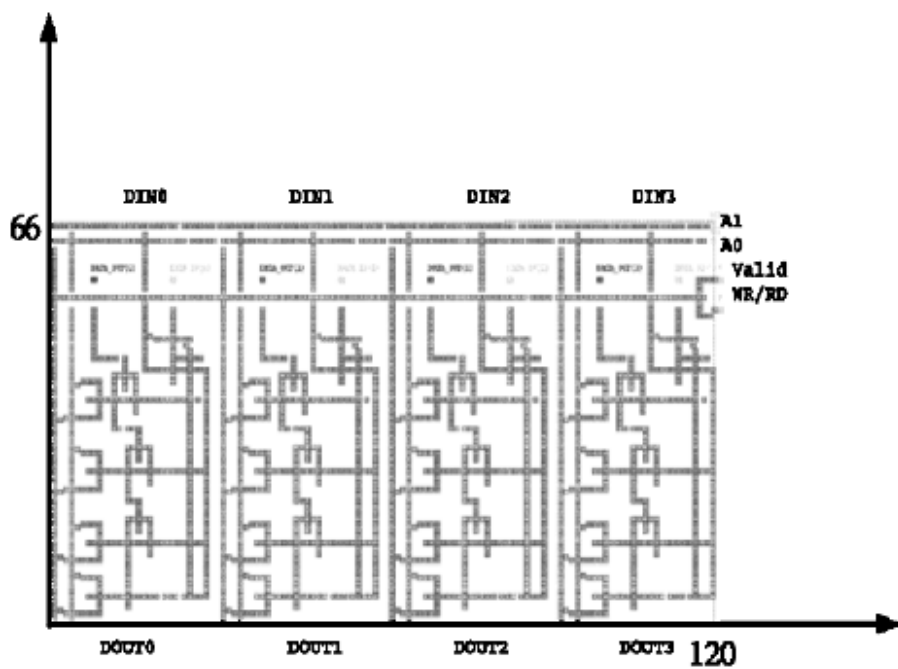
Na sliki 27 je hibridna struktura 4 x 4 pomnilnika. Efektivna površina je 66 in 120 celic,  $A = 0.792 \mu m^2$ ,  $d = 10 \text{ nm}$ . Površina je glede na 1 x 4 hibridni pomnilnik več kot 4-krat večja, ker moramo pripeljati kontrolne signale do vsake zanke.

- **Drevesna H struktura**

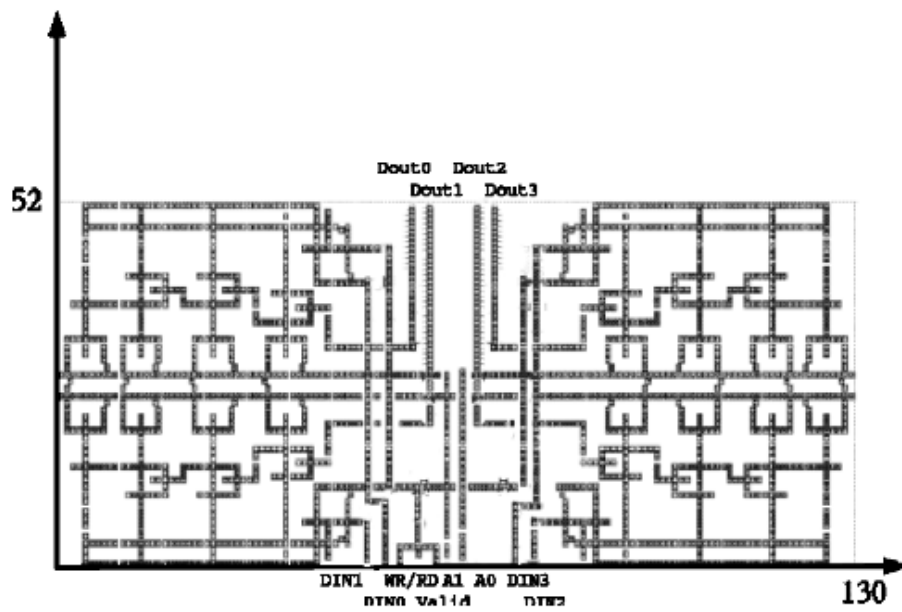
Na sliki 28 je drevesna H struktura. Tu je možno deljenje kontrolnih signalov in s tem zmanjšanje površine. Efektivna površina je 130 in 52 celic,  $A = 0.676 \mu m^2$ ,  $d = 10 \text{ nm}$ .



Slika 26: 4 x 4 paralelni pomnilnik



Slika 27: 4 x 4 hibridni pomnilnik



Slika 28: 4 x 4 drevesni H-pomnilnik

## 5.6 Ugotovitve

### 5.6.1 Splošna primerjava

	<b>Prednosti</b>	<b>Slabosti</b>
Serijski pomnilnik	Gostota na površino, 160x120 celic za 16 bitov	Zakasnitve pri branju/pisanju
Paralelni pomnilnik	Zakasnitev pri branju/pisanju	Gostota na površino
Drevesni H pomnilnik	Gostota na površino	Zakasnitev pri branju/pisanju Nastandardni dostopni način
Hibridni pomnilnik	Zakasnitev pri branju Gostota na površino	Zakasnitev pri pisanju

### 5.6.2 Efektivna področja

	$A_{hyb}(\mu m^2)$	$A_{par}(\mu m^2)$	$A_{ser}(\mu m^2)$	R
1 x 4	148	455	74	32%
4 x 4	1.36	1.63	0.296	80%
4 x 4 - H strukt.	0.676	1.63	0.296	41%

V zgornji tabeli so prikazana efektivna področja hibridnih, paralelnih in serijskih QCA pomnilnikov. Poleg tega je še izračunano razmerje  $R = A_{hyb}/A_{ser}$  ( $A_x$  – efektivno področje) hibridnega pomnilnika. Večji kot je hibridni pomnilnik, slabše razmerje učinkovitosti ima na račun kontrolne logike.

	$A(\mu m^2)$	$A_u(\mu m^2)$	$A_{un}(\mu m^2)$	$A_u(\%)$	$A_{un}(\%)$
1 x 4 SER	0,07	0,02	0,05	28	72
1 x 4 PAR	0,45	0,12	0,33	27	73
1 x 4 HYB	0,15	0,05	0,1	32,5	67,5
4 x 4 SER	0,3	0,08	0,21	28	72
4 x 4 PAR	1,63	0,39	1,24	24	76
4 x 4 HYB	0,79	0,23	0,56	29,5	70,5
4 x 4 H drev,	0,68	0,18	0,49	27,1	72,9

V zgornji tabeli so prikazana razmerja med  $A_u$  (delež uporabljenih celic) in  $A_{un}$  (delež neuporabljenih celic) pri posameznih strukturah. Kot lahko razberemo je delež uporabljenih celic na delavnem področju med 24

## 5.7 QCA vs. CMOS

Najbolj optimizirana verzija QCA drevesne H strukture trenutno obsega 324 *celic/bit* oz. kapaciteto 4.26 *GBit/cm<sup>2</sup>*. To je že skoraj primerljivo s CMOS tehnologijo. V prihodnje se pričakujejo nadaljne optimizacije, s pomajšanjem na molekularni nivo, vse do samo 18.75 *celic/bit* kar bi pomenilo kapacitete do 281.58 *GBit/cm<sup>2</sup>*, kar je mnogo več kot omogoča CMOS tehnologija.

Slabost QCA realizacij je tudi slaba izkoriščenost prostotora. Glavni razlog za to je mnogo večji medsebojni vpliv QCA celic kot električnih elementov pri CMOS tehnologiji. Delež uporabljenih celic glede na celotno površino se giblje pri QCA okrog 30% kar je mnogo manj kot pri CMOS tehnologiji.

Glede na to, da bo QCA pomnilnik realiziran iz vsaj nekaj 100 celic, se bodo predvidene zakasnitve skozi pomnilnik gibale od 0.1 ns do 10 ns, kar je dosti bolje a še vedno primerljivo s CMOS tehnologijo.

Skupna velikost	2 GB
Gostota modula	256 Mb
Konfiguracija	32M x 64
Pasovna širina modula	3.7 GB/s
Ura pomnilnika	4.3 ns (232.5 Mhz)
Podatkovni tok	466 MT/s
Zakasnitve (tCL-tRCD-tRP)	3.0 – 4 – 4
Čas osveževanja	7.8125 s
Minimum Row Precharge Time (tRP)	18 ns
Minimum Row Active to Row Active delay (tRRD)	10 ns
Minimum RAS to CAS delay (tRCD)	18 ns
Minimum Active to Precharge Time (tRAS)	40 ns
Minimum Active to Active/Refresh Time (tRC)	60 ns
Minimum Refresh to Active/Refresh Command Period (tRFC)	70 ns
Power dissipation	10.4 W

## 6 Zaključek

Za učinkovitejše rezultate, kar se CMOS tehnologije tiče, bi bilo potrebno dobiti orodja, ki znajo izdelati vezje iz višjenivojskega (HDL) jezika tudi na nivoju 90, 65 in 45 nanometerske tehnologije. Žal so taka orodja nam nedostopna. Poleg teh bi potrebovali ustrezno znanje za izvajanje zahtevnih simulacij na nivoju polprevodnikov z najnovejšimi modeli oz. abstrakcijami tranzistorjev (npr. BSIM4), kar je zunaj domene inženirja računalništva.

Orodje s katerim smo ocenjevali površino vezij (*InCyte*), je precej grobo in je seveda namenjeno ocenjevanju večjih vezij, ki vključujejo tudi pomnilnike in krmilnike raznih naprav, zato bi lahko bili podatki na nivoju manjših vezij delno zavajajoči. Zato smo nekatera (manjša) vezja preverili tudi z orodjem Microwind, in ugotovili, da so rezultati primerljivi, niso pa eksaktni. Žal nismo imeli na voljo 45nm knjižnice, kajti to je bila le preizkusna različica. Podobno je bilo iz orodjem za analizo in optimizacijo vezij. Knjižnic za delo v 'sub-100nm' tehnologijah preprosto ni moč najti. Zato so bile zakasnitve merjene na podlagi edino vgrajene ASIC knjižnice: 0.5 mikronske vzorčne knjižnice. Ti podatki so glede na trenutno tehnologijo neuporabni.

Najbolj kočljiva tema so torej primerjave zakasnitev saj za te nimamo oprijemljivih podatkov. Pri QCA je zakasnitev vezana na maksimalno hitrost adiabatnega preklopa. Pri najbolj verjetni molekularni QCA realizaciji je preklonni čas odvisen od časa, ki je potreben, da elektron prepotuje skozi molekulo. Predvidene hitrosti so od 100 GHz do 10 THz, vendar določeni skeptiki utemeljeno ocenjujejo, da bodo (vsaj na začetku) hitrosti večjih vezij bistveno nižje, celo v MHz. Vse bo odvisno od implementacije adiabatnega preklopa in realnih sktruktur, ki bodo delale na sobni temperaturi, zato so dobljeni rezultati zgolj informativni in kažejo kvečjemu primerljivost skalabilnosti posameznih vezij.

Povsem druga zgodba je poraba energije. Tudi tukaj smo ugotovili, da bo, kot kaže, vse odvisno od implementacije adiabatnega preklopa. V trenutni, prej omenjeni (teoretični) rešitvi bi disipacija lahko omejila frekvenco na največ nekaj GHz, kar je zelo primerljivo s CMOS tehnologijo. V naših izračunih smo uporabljali disipacijo  $100 \text{ W/cm}^2$  - to je zelo pesimistična ocena, kljub temu se poraba energije v QCA strukturah še vedno giblje v območju  $\mu\text{W}$ . Povsem drugače je pri CMOS tehnologiji. Faktor porabe je tu že v štartu 1000-krat večji in se s skaliranjem samo še povečuje.

Kvantni celični avtomati so zaenkrat še stvar prihodnosti in so bili tekom seminarske naloge tudi tako obravnavani. Vseeno pa je konec klasičnih

CMOS vezij neizbežen, saj s konstantnim manjšanjem CMOS tehnologije počasi prehajamo v področja kvantne fizike, kjer se pojavljajo novi problemi (npr. kvantno tuneliranje). Kvantni celični avtomati pa so le ena izmed možnih rešitev, obstajajo tudi druge tehnologije, ki si obetajo prevzeti mesto CMOS. Trenutno ne moremo napovedati kaj bo naslednik CMOS tehnologije. Verjetno bodo različne tehnologije ustrezale različnim problemom.

Po testiranjih ter primerjanjih le-teh s CMOS tehnologijo smo prišli do spoznanja, da če bi nam uspelo realizirati QCA strukture, bi se vsekakor odrezale enako ali bolje od trenutne CMOS tehnologije, vsaj pri porabi energije ter prostora. Vsekakor pa je težko napovedati, kakšna bo poraba energije ter prostora pri bodočih CMOS tehnologijah, še težje pa je primerjati bodoče tehnologije med seboj, zato je ta izdelek bolj informativne narave kot kaj drugega.



## Literatura

- [1] H. Cho and E. Swartzlander. Modular design of conditional sum adders using quantum-dot cellular automata.
- [2] H. Cho and E. Swartzlander. Adder designs and analyses for quantum-dot cellular automata. *IEEE Transactions on Nanotechnology*, 6(3):374–383, 2007.
- [3] Jr. Earl E. Swartzlander. Serial parallel multiplier design in quantum-dot cellular automata. *Computer Arithmetic*, 2007.
- [4] Sarah Elizabeth Frost and Randal Thomas Rausch. Memory in motion: A study of storage structures. 2002.
- [5] Ottavi, Vankamamidi, Lombardi, and Pontarelli. Novel memory designs for qca implementation. *Nanotechnology, 2005. 5th IEEE Conference on*, pages 545–548, 2005.
- [6] M. Ottavi, S. Frost-Murphy, E. DeBenedictis, M. Frank, and P. Kogge. Design and characterization of a clock distribution circuit for qca. <https://cfwebprod.sandia.gov/cfdocs/CCIM/docs/clocking.pdf>.
- [7] Etienne Sicard and Syed Mahfuzul Aziz. Introducing 65 nm technology in microwind3. 2005. [www.microwind.org](http://www.microwind.org).
- [8] I. Sideris, K. Anagnostopoulos, P. Kalivas, and K. Pekmestzi. Novel systolic schemes for serial-parallel multiplication. *Eurasip*, 2005.
- [9] W. Wang, K. Walus, and G. Jullien. Quantum-dot cellular automata adders. *IEEE Transactions on Nanotechnology*, pages 461–464.