



**УНИВЕРЗИТЕТ  
У НИШУ**

**Универзитетска  
Библиотека  
"Никола Тесла"**

**UNIVERSITY  
OF NIŠ**

**University  
Library  
"Nikola Tesla"**



**ДИГИТАЛНИ  
РЕПОЗИТОРИЈУМ  
УНИВЕРЗИТЕТА  
У НИШУ**

**Библиотека  
Дисертације**

**DIGITAL  
REPOSITORY  
OF THE UNIVERSITY  
OF NIŠ**

**Ph.D. Theses**

---

Univerzitet u Nišu  
Elektronski fakultet

Vladimir Ćirić

PROJEKTOVANJE PARCIJALNO  
VISOKOPOUZDANIH PROCESORSKIH  
POLJA

Doktorska disertacija

Niš, 2010.



# Projektovanje parcijalno visokopouzdatih procesorskih polja

Vladimir Ćirić

– Doktorska disertacija –

Niš, 2010.



Zahvaljujem se svom mentoru, prof. dr Ivanu Milentijeviću, na trudu i vremenu, i na svojoj pomoći koju mi je pružio, smernicama, revizijama, komentarima i mnogobrojnim diskusijama u toku izrade ovog rada. Takođe se zahvaljujem prof. dr Aleksandru Cvetkoviću na pomoći pri formulaciji i reviziji matematičkih izraza, definicija i teorema u kojima su izloženi rezultati ovog rada. Zahvaljujem se prof. dr Emini Milovanović i prof. dr Miletu Stojčevu na savetima u toku izrade rada. Zahvalan sam kolegama iz laboratorije L3 na svojoj stručnoj i tehničkoj podršci koju su mi pružili.

Naravno, neizmerno sam zahvalan svojoj supruzi Milici na podršci i razumevanju, kao i svojim roditeljima i sestri.



# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Digitalna obrada signala</b>	<b>7</b>
2.1	DSP algoritmi . . . . .	8
2.1.1	Konvolucija . . . . .	8
2.1.2	Korelacija . . . . .	11
2.1.3	Digitalno filtriranje . . . . .	13
2.1.4	Diskretna kosinusna transformacija . . . . .	14
2.2	Reprezentacija DSP algoritama . . . . .	17
2.2.1	Blok dijagrami . . . . .	18
2.2.2	Graf toka signala . . . . .	19
2.2.3	Graf toka podataka . . . . .	20
2.2.4	Graf zavisnosti . . . . .	22
2.3	Sistolička polja . . . . .	24
2.3.1	Procesorska polja . . . . .	24
2.3.2	Klase i topologije sistoličkih polja . . . . .	26
2.3.3	Semi-sistolička polja . . . . .	29
2.3.4	Primeri mapiranja DSP algoritama . . . . .	32
2.4	Implementacija prototipa DSP sistema . . . . .	44
2.4.1	Arhitektura FPGA kola . . . . .	44
2.4.2	Spartan familija FPGA kola . . . . .	45
2.4.3	Rezultati implementacije . . . . .	49
<b>3</b>	<b>Pouzdana sistemi i tolerancija na greške</b>	<b>55</b>
3.1	Nanotehnologija . . . . .	56
3.1.1	Ograničenja litografije . . . . .	56
3.1.2	VLSI nanoelektronika . . . . .	58
3.1.3	Otpornost na defekte u nanotehnologiji . . . . .	60
3.2	Defekt, greška, otkaz . . . . .	62
3.3	Tehnike za povećanje pouzdanosti sistema . . . . .	64
3.3.1	Hardverska redundansa . . . . .	65



3.3.2	Informaciona, vremenska i softverska redundansa . . . . .	71
3.4	Analiza pouzdanosti sistema . . . . .	72
3.4.1	Minimalna jedinica zamene . . . . .	72
3.4.2	Pouzdanost sistema sa N jedinica zamene . . . . .	73
3.5	Tolerancija grešaka . . . . .	76
3.5.1	Neprecizna izračunavanja . . . . .	76
3.5.2	Sistemi tolerantni na greške . . . . .	78
3.5.3	Primer sistema tolerantnog na greške . . . . .	79
<b>4</b>	<b>Parcijalno visokopouzdati sistemi</b>	<b>81</b>
4.1	Metrika otkaza sistema . . . . .	82
4.2	Definicija parcijalno visokopouzdatih sistema . . . . .	85
4.3	Tranzitivna zatvaranja . . . . .	89
4.3.1	Definicija tranzitivnog zatvaranja . . . . .	90
4.3.2	Konstrukcija tranzitivnog zatvaranja digrafa: Warshallov algoritam . . . . .	91
4.3.3	Tranzitivno zatvaranje <i>bit-plane</i> FIR filtra . . . . .	95
4.4	Primer projektovanja parcijalno visokopouzdanog bit-plane FIR filtra . . . . .	101
4.5	Implementacija parcijalno visokopouzdanog bit-plane FIR filtra	108
<b>5</b>	<b>Prinos parcijalno visokopouzdatih sistema</b>	<b>113</b>
5.1	Prinos ET i PDT sistema . . . . .	114
5.2	Poređenje prinosa ET i PDT sistema . . . . .	118
5.2.1	Uticao broj podistema na prinos PDT sistema . . . . .	121
5.3	Netolerantna oblast ortogonalnog polja . . . . .	123
<b>6</b>	<b>Primeri primene parcijalno visokopouzdatih sistema</b>	<b>129</b>
6.1	Parcijalno visokopouzdana 1D polja . . . . .	129
6.1.1	Jednosmerno PDT polje za množenje matrice i vektora	130
6.1.2	Dvosmerno PDT polje za množenje matrice i vektora .	138
6.2	Projektovanje parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva . . . . .	142
6.2.1	Primer primene Hamingove metrike . . . . .	143
6.2.2	Min-plus algebra . . . . .	146
6.2.3	Primer primene Euklidske metrike . . . . .	149
6.2.4	Analiza prinosa parcijalno visokopouzdanog heksago- nalnog polja za množenje brojeva . . . . .	158
6.2.5	Rezultati implementacije parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva . . . . .	161
6.3	Projektovanje parcijalno visokopouzdanog polja sa PDT ćelijama	162

<b>7 Zaključak</b>	<b>167</b>
<b>Literatura</b>	<b>182</b>
<b>Dodatak A</b>	<b>183</b>



# Lista slika

2.1	Vizuelizacija konvolucije . . . . .	10
2.2	Linearni FIR filter sedmog reda . . . . .	15
2.3	Primer DCT transformacije . . . . .	16
2.4	Primer inverzne DCT transformacije . . . . .	16
2.5	Blok dijagram FIR filtra trećeg reda . . . . .	19
2.6	Blok dijagram FIR filtra trećeg reda sa emisionom linijom za uvođenje ulaznih podataka . . . . .	19
2.7	SFG FIR filtra trećeg reda . . . . .	21
2.8	Primer grafa toka podataka . . . . .	21
2.9	DFG reprezentacija filtra trećeg reda . . . . .	22
2.10	DG reprezentacija filtra trećeg reda . . . . .	23
2.11	Klasifikacija paralelnih računara . . . . .	24
2.12	SIMD i MIMD arhitektura . . . . .	25
2.13	Osnovna konfiguracija sistoličkih polja . . . . .	26
2.14	Linearna sistolička polja . . . . .	28
2.15	Pokrivanje površine . . . . .	30
2.16	Arhimedova pokrivanja . . . . .	31
2.17	Semi-sistoličko $4^4$ polje . . . . .	31
2.18	Linearno jednosmerno polje za množenje matrice i vektora . . . . .	33
2.19	Linearno dvosmerno polje za množenje matrice i vektora . . . . .	33
2.20	Tok podataka kroz 1-D dvosmerno polje . . . . .	34
2.21	Algoritam za množenje brojeva . . . . .	35
2.22	Sistoličko $3^6$ polje za množenje brojeva . . . . .	36
2.23	Wavefront $6^3$ polje za množenje brojeva . . . . .	38
2.24	Blok dijagram bit-plane arhitekture . . . . .	39
2.25	Funkcionalni blok dijagram BP arhitekture . . . . .	40
2.26	Graf toka podataka BP arhitekture . . . . .	41
2.27	Tok podataka kroz BP FIR filter . . . . .	43
2.28	Uopštena struktura FPGA čipa . . . . .	45
2.29	Blok dijagram osnovnih gradivnih blokova Spartan-II čipova . . . . .	46
2.30	Blok šema SL Spartan-II čipa . . . . .	48

2.31	Implementacija jednosmernog polja za množenje matrice i vektora na SpartanII XCS200 čipu . . . . .	51
3.1	Struktura MOSFET tranzistora . . . . .	57
3.2	Struktura grafitne nanocevi . . . . .	59
3.3	Struktura memorije u nanotehnologiji . . . . .	59
3.4	Langmur-Blodžetova tehnika za postavljanje paralelnih nanocevi na čip . . . . .	60
3.5	Koraci pri CAD projektovanju FPGA kola . . . . .	62
3.6	Uzročno - posledične veze defekta, greške i otkaza sistema . . .	63
3.7	Karakteristike defekta . . . . .	64
3.8	Model pasivne hardverske redundanse sa tri istovetna modula i jednim glasačem . . . . .	66
3.9	Model za eliminaciju jedinstvene tačke otkaza korišćenjem tri nezavisna glasača . . . . .	67
3.10	Uporedni prikaz pouzdanosti različitih metoda projektovanja .	68
3.11	Model aktivne hardverske redundanse sa jednim rezervnim modulom . . . . .	69
3.12	Osnovni princip vremenske redundanse . . . . .	71
3.13	Pozicija ET i FT sistema u modelu defekt-greška-otkaz . . . .	78
3.14	Rezultati subjektivne ocene kvaliteta digitalne telefonske sekretarice sa defektnom memorijom . . . . .	80
4.1	Primer grafa . . . . .	85
4.2	Graf sa pridruženim izlaznim čvorovima . . . . .	86
4.3	Primer skupa značajnih čvorova . . . . .	87
4.4	Primer funkcije uređenja skupa . . . . .	87
4.5	Digraf $G$ i njegovo tranzitivno zatvaranje $G^*$ . . . . .	91
4.6	Konstrukcija grafa po Warshallovom algoritmu . . . . .	91
4.7	Warshallov algoritam . . . . .	92
4.8	Primer matrice reprezentacije grafa . . . . .	92
4.9	Floyd-Warshallov algoritam . . . . .	93
4.10	Warshallov algoritam za određivanje tranzitivnog zatvaranja na osnovu matrice susedstva . . . . .	94
4.11	Primer tranzitivnog zatvaranja . . . . .	95
4.12	Graf $G$ toka podataka <i>bit-plane</i> FIR filtra . . . . .	97
4.13	Modifikovani graf toka podataka <i>bit-plane</i> FIR filtra . . . . .	98
4.14	Graf povezanosti kolona <i>bit-plane</i> FIR filtra . . . . .	99
4.15	Graf povezanosti vrsta <i>bit-plane</i> FIR filtra . . . . .	99
4.16	Osnovno polje <i>bit-plane</i> FIR filtra za projektovanje PDT <i>bit-plane</i> polja . . . . .	102

4.17	Dijagram toka projektovanja PDT sistema . . . . .	103
4.18	Primer modifikovanog grafa propagacije greške <i>bit-plane</i> FIR filtra . . . . .	104
4.19	Primer tranzitivnog zatvaranje <i>bit-plane</i> FIR filtra . . . . .	105
4.20	Primer mapiranja elemenata tranzitivnog zatvaranja na elemente mape uticaja greške . . . . .	106
4.21	Graf parcijalno visokopouzdanog arhitekture <i>bit-plane</i> FIR filtra	108
4.22	Primer parcijalno visokopouzdanog <i>bit-plane</i> FIR filtra . . . . .	108
4.23	Relativni odnos resursa potrebnih za implementaciju arhitektura u funkciji od stepena parcijalne visoke pouzdanosti $\alpha$ . . . . .	111
5.1	Uporedni prikaz prinosa ET i PDT polja za slučaj SC3 tehnike	117
5.2	Uporedni prikaz prinosa ET i PDT polja za slučaj TMR tehnike	118
5.3	Grafik zavisnosti verovatnoće za koju se prinosi ET i PDT tehnike sa SC3 ćelijama seku . . . . .	121
5.4	Grafik zavisnosti verovatnoće za koju se prinosi ET i PDT tehnike sa TMR ćelijama seku . . . . .	122
5.5	Tranzitivno zatvaranje i određivanje kardinalnosti skupa $\aleph(\mathbf{P}_{DT}(1))$ <i>bit-plane</i> FIR filtra . . . . .	125
5.6	Grafik funkcije $\Gamma(\alpha)$ <i>bit-plane</i> FIR filtra . . . . .	127
5.7	Funkcija $P(\alpha, T)$ <i>bit-plane</i> FIR filtra za konstantno $T$ . . . . .	128
6.1	Grafovi mogućih propagacija grešaka kroz jednosmerno 1D polje za množenje matrice i vektora . . . . .	131
6.2	Arhitektura parcijalno visokopouzdanog jednosmernog 1D polja za množenje matrice i vektora . . . . .	134
6.3	Uporedni prikaz $\Gamma(\alpha)$ funkcija 1D jednosmernog polja i 2D polja sa $4^4$ topologijom . . . . .	135
6.4	Promena verovatnoće pojave defekta za koju se prinosi ET i PDT tehnike seku kod 1D jednosmernog polja . . . . .	136
6.5	Grafik uštede resursa potrebnih za implementaciju 1D jednosmernog PDT polja za množenje matrice i vektora . . . . .	138
6.6	Graf propagacije grešaka kroz dvosmerno 1D polje za množenje matrice i vektora . . . . .	139
6.7	Ćelija heksagonalnog polja za množenje brojeva . . . . .	142
6.8	Graf propagacije grešaka tipa "A" kroz 2D $3^6$ polje za množenje brojeva . . . . .	143
6.9	Graf propagacije grešaka tipa "B" kroz 2D $3^6$ polje za množenje brojeva za Hamingovu metriku . . . . .	145
6.10	Težinski uticaj grešaka kod heksagonalnog 2D polja za množenje brojeva . . . . .	150

---

6.11	Težinski graf propagacije grešaka kroz heksagonalno 2D polje za množenje brojeva . . . . .	150
6.12	DFG parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva . . . . .	158
6.13	Grafički prikaz funkcije $\Gamma(\alpha)$ parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva . . . . .	160
6.14	Verovatnoća pojave defekta za koju ET i PDT tehnike imaju isti prinos u funkciji od $\alpha$ kod PDT heksagonalnog polja za množenje brojeva . . . . .	160
6.15	Grafik uštede resursa potrebnih za implementaciju parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva . . .	162
6.16	Ćelija jednosmernog 1D polja za množenje matrice i vektora . . .	163
6.17	Parcijalno visokopouzdanano jednosmerno 1D polje za množenje matrice i vektora sa parcijalno visokopouzdanim ćelijama . . .	165
6.18	Grafički prikaz rezultata implementacije PDT jednosmernog 1D polja za množenje matrice i vektora sa PDT ćelijama . . .	166

# Lista tabela

2.1	Primeri tipičnih DSP algoritama i njihove primene . . . . .	9
2.2	Kapacitet raspoloživog blok RAM-a za Spartan-II . . . . .	49
2.3	Rezultati FPGA implementacije jednosmernog polja za množenje matrice i vektora . . . . .	50
2.4	Rezultati FPGA implementacije dvosmernog polja za množenje matrice i vektora . . . . .	51
2.5	Rezultati FPGA implementacije heksagonalnog polja za množenje brojeva . . . . .	52
2.6	Rezultati implementacije semi-sistoličkog BP FIR filtra . . . . .	53
4.1	Dimenzije implementiranih <i>bit-plane</i> polja . . . . .	109
4.2	Rezultati implementacije <i>bit-plane</i> FIR filtra . . . . .	110
5.1	Cene upotrebljivog čipa za različite verovatnoće otkaza $p$ . . . . .	117
6.1	Rezultati implementacije jednosmernog 1D PDT polja za množenje matrice i vektora sa SC3 tehnikom . . . . .	137
6.2	Rezultati implementacije jednosmernog 1D PDT polja za množenje matrice i vektora sa TMR tehnikom . . . . .	137
6.3	Rezultati implementacije dvosmernog 1D PDT polja za množenje matrice i vektora sa SC3 tehnikom . . . . .	142
6.4	Rezultati implementacije dvosmernog 1D PDT polja za množenje matrice i vektora sa TMR tehnikom . . . . .	142
6.5	Rezultati implementacije parcijalno visokopouzdanog heksagonalnog polja za množenje matrice i vektora sa TMR tehnikom	161
6.6	Rezultati implementacije parcijalno visokopouzdanog heksagonalnog polja za množenje matrice i vektora sa SC3 tehnikom	161
6.7	Rezultati FPGA implementacije parcijalno visokopouzdanog jednosmernog 1D polja za množenje matrice i vektora sa parcijalno visokopouzdanim ćelijama . . . . .	165





# Lista skraćenica

DSP –	Digitalna obrada signala (eng. <i>Digital Signal Processing</i> )
ILP –	Paralelizam na nivou instrukcija (eng. <i>Instruction Level Parallelism</i> )
LLP –	Paralelizam na nivou petlji (eng. <i>Loop Level Parallelism</i> )
FT –	Otpornost na otkaze (eng. <i>Fault Tolerance</i> )
DT –	Tolerantnost na defekte (eng. <i>Defect Tolerance</i> )
ET –	Tolerantnost na greške (eng. <i>Error Tolerance</i> )
VLSI –	Tehnologija proizvodnje integrisanih kola sa veoma velikim stepenom integracije (eng. <i>Very Large Scale Integration</i> )
PDT –	Parcijalna opornost na defekte (eng. <i>Partial Defect Tolerant</i> )
BP –	Bitske ravni (eng. <i>Bit-Plane</i> )
VHDL –	Jezik za opis hardvera (eng. <i>Very high speed circuit Hardware Definition Language</i> )
DSP –	Digitalna obrada signala (eng. <i>Digital Signal Processing</i> )
SNR –	Odnos signal – šum (eng. <i>Signal to Noise Ratio</i> )
FIR –	Sistem sa konačnim impulsnim odzivom (eng. <i>Finite Impulse Response</i> )
IIR –	Sistem sa beskonačnim impulsnim odzivom (eng. <i>Infinite Impulse Response</i> )
HDL –	Jezik za opis hardvera (eng. <i>Hardware Description Language</i> )
SFG –	Graf toka signala (eng. <i>Signal Flow Graph</i> )
DFG –	Graf toka podataka (eng. <i>Data Flow Graph</i> )

---

DG –	Graf zavisnosti (eng. <i>Dependence Graph</i> )
SIMD –	Sistem sa jedinstvenom instrukcijom nad višestrukim podacima (eng. <i>Single Instruction Multiple Data - SIMD</i> )
MIMD –	Sistem sa višestrukim instrukcijama nad višestrukim podacima (eng. <i>Multiple Instruction Multiple Data - MIMD</i> )
SRSP –	Strogo regularno sistoličko polje
MAC –	Procesna jedinica koja izvršava operaciju množenja, a rezultat dodaje na prethodni međurezultat (eng. <i>Multiply-ACcumulate</i> )
ASIC –	Integrirana kola specifične namene (eng. <i>Application Specific Integrated Circuits</i> )
PGA –	Programabilno logičko kolo (eng. <i>Programmable Gate Array</i> )
FPGA –	Programabilno polje (eng. <i>Field-Programmable Gate Array</i> )
CLB –	Konfiguracioni logički blok (eng. <i>Configurable Logic Block</i> )
PSM –	Programabilna matrica veza (eng. <i>Programmable Switch Matrix</i> )
LC –	Logička ćelija
LUT –	<i>Look-Up</i> tabela
SoC –	Sistem na čipu (eng. <i>System On Chip</i> )
TMR –	Trostruka modularna redundansa (eng. <i>Triple Modular Redundancy</i> )
NMR –	N-Modularna redundansa (eng. <i>N-Modular Redundancy</i> )
SC –	rezervna komponenta (eng. <i>Spare Component</i> )
SC3 –	hardverska redundansa SC sa dve rezervne kopije (tri modula ukupno)
ME –	Blok za određivanje pomeraja na slici (eng. <i>Motion Estimator</i> ) MPEG-2, MPEG-4, i H.26x video kodeka
SNR –	Odnos signal-šum (eng. <i>Signal-to-Noise Ratio</i> )
DCT –	Diskretna kosinusna transformacija (eng. <i>Discrete Cosine Transform</i> )
FET –	Tranzistor sa efektom polja (eng. <i>Field Effect Transistor</i> )

- 
- CNT – Grafitne nanocevi (eng. *Carbon NanoTube*)  
SNW – Silikonski nanoprovodnici (eng. *Silicon NanoWires*)  
CAD – Računarski podržano projektovanje (eng. *Computer-Aided Design*)



# Glava 1

## Uvod

Od samog početka razvoja računarskih arhitektura značajan deo tržišta zauzimaju arhitekture za digitalnu obradu signala (eng. *Digital Signal Processing - DSP*). DSP sisteme od ostalih sistema izdvajaju dve važne karakteristike. To su zahtev za obradom u realnom vremenu, kao i obrada vođena podacima. Naime, kod DSP algoritama program se izvršava nebrojano mnogo puta, tj. ponavlja se u beskonačnoj petlji [1]. Brzinu obrade moguće je povećati eksploatacijom paralelizma na nivou instrukcija algoritma (eng. *Instruction Level Parallelism - ILP*), ili na nivou petlji (eng. *Loop Level Parallelism - LLP*). Primenom neke od tehnika projektovanja, DSP algoritam je moguće preslikati na polje procesnih elemenata i korišćenjem visokog stepena paralelizacije odgovoriti zahtevu za obradom u realnom vremenu [1, 2, 3].

Uz trend povećanja brzine obrade, i u poslednje vreme nezaobilaznog smanjenja potrošnje, praktično od nastanka integrisanog kola postoji i trend smanjenja potrebne površine silicijuma za implementaciju digitalnih sistema. Smanjenje površine na kojoj se implementira arhitektura postiže se unapređenjem tehnologije, odnosno smanjenjem veličine individualnih komponenti koje čine sistem [4]. Posledica većeg stepena integracije je mogućnost implementacije složenijih računarskih arhitektura i DSP algoritama na jednom čipu. U novije vreme mere veličine pojedinih komponenti izražavaju se u broju atoma, odnosno molekula koje zauzimaju [5]. Najznačajniji problemi u procesu fabrikacije ovakvih čipova su varijacije u proizvodnji koje dovode do razlika u istoj seriji proizvoda, stepen defekata koji se javlja, škart, kao i osetljivost samog procesa na interne i eksterne smetnje. Ovakav trend će po svemu sudeći smanjiti efektivnu ekonomsku isplativost budućih tehnologija. Trend razvoja tehnologije nameće zaključak da će proizvodnja integrisanih kola uskoro dostići nivo od više desetina milijardi tranzistora po čipu, među kojima je par hiljada neispravnih [5, 6, 7]. Proizvodnja čipa sa 100% is-

pravnim komponentama i vezama postaće veoma skupa. Uz trend razvoja integrisanih kola, koji je sve bliži fizičkim granicama koje je teoretski trenutnom tehnologijom moguće postići, projektanti će morati da ovu činjenicu uvrste u projektne zahteve i na arhitekturnom nivou [5].

Ekonomska isplativost kompleksnih računarskih arhitektura u tehnologiji sa veoma velikim stepenom integracije dovodi klasične tehnike projektovanja u pitanje, tako da je neophodno istražiti alternativne metode [7]. Uz povećanje stepena integracije raste i verovatnoća pojave defekata. Defekt se može definisati kao fizička neispravnost, nepotpunost, ili oštećenje koje se javlja unutar neke hardverske ili softverske komponente. Greška je eventualna manifestacija defekta, ili nekog štetnog spoljnog uticaja. Konačno, ako greška, prouzrokovana defektom, rezultira time da sistem neku svoju funkciju izvršava pogrešno, kaže se da je došlo do otkaza sistema. Suštinski, otkaz je neizvršavanje neke akcije na način na koji bi trebalo, ili kako se to očekuje [8, 9].

Koncept izgradnje sistema od delova koji mogu biti inicijalno neispravni, privremeno neispravni, ili koji mogu postati neispravni nakon određenog vremena upotrebe nije nov. Ovaj problem razmatran je još 40-tih godina prošlog veka u radovima Von Neumanna, Gödela i Kleina [10]. Veliki broj metoda projektovanja arhitektura otpornih na otkaze razvijen je počev od 1960-tih godina. Otpornost arhitekture na otkaze (eng. *Fault Tolerance - FT*) predstavlja osobinu sistema da spreči pojavu greške, a samim tim i mogućnost otkaza sistema usled pojave defekta [8, 9, 11]. Pod ispravnim radom podrazumeva se da je izlaz sistema identičan izlazu kada su sve komponente sistema ispravne. Ove osobine uglavnom se postižu uvođenjem dodatnih, redundantnih komponenti. Sistemi koji se koriste u hazardnim uslovima, kao na primer u nuklearnim reaktorima, avionima i svemirskim stanicama su, pre svega zbog uticaja zračenja, u velikoj meri podložni greškama. Održavanje i popravka takvih sistema je vrlo skupa i zahteva dosta vremena. Zbog toga, pored performansi, otpornost na otkaze, koja predstavlja mogućnost sistema da korektno funkcioniše uprkos pojavi defekata, postaje vrlo važno pitanje [12].

U mnogim DSP algoritmima određene greške ne dovode do otkaza sistema, odnosno moguće je tolerisati greške na izlazu sistema, pod uslovom da se nalaze u određenim granicama [7, 13, 14]. Takve sisteme nazivamo sistemima tolerantnim na greške (eng. *Error Tolerant - ET*). Kod ovih sistema, za razliku od FT sistema, reč "otporni" u nazivu zamenjena je rečju "tolerantni". Ovo je i glavna odlika novog pravca projektovanja arhitektura kod kojih je moguće tolerisati određene tipove grešaka [13]. Drugim rečima, ET je koncept kod koga je projektant svestan mogućnosti postojanja defekta koji uzrokuje pojavu greške, zna njenu poziciju i uticaj, i ne menja svojstva

arhitekture kako bi otklonio eventualni nastanak ovakve greške, jer je primena sistema takva da greška neće bitno uticati na izlaze sistema. Multimedijalni sistemi su primer sistema tolerantnih na greške [13, 15]. Kod multimedijalnih sistema uobičajeno je da projektanti koriste mogućnosti čoveka da rekonstruiše informaciju na osnovu signala slabijeg kvaliteta. Svi algoritmi za kodiranje video signala koji uvode i neki nivo kompresije manje ili više uspešno eksploatišu ovu činjenicu [16, 17, 18]. Na ovaj način vrši se ušteda u propusnom opsegu potrebnom za prenos informacije, ili smanjuje obim hardvera i time postiže, pored ostalog, niža cena proizvodnje.

Interesantno pitanje je: ukoliko neki uređaj za procesiranje signala ima manji hardverski defekt, da li će i dalje biti dovoljno dobar za krajnjeg korisnika? Ukoliko je i u prisustvu defekta dovoljno dobar za krajnjeg korisnika, ovakav uređaj pre može biti prodat, umesto da bude odbačen kao škart [13]. Ublažavanje zahteva za 100% ispravnim uređajima i vezama značajno može smanjiti cenu proizvodnje i testiranja [19].

U cilju povećanja ekonomske isplativosti fabrikacije arhitekture u VLSI (eng. *Very Large Scale Integration*) tehnologiji, ili u nano-tehnologiji, gde su problemi još izraženiji, autor u ovom radu predlaže primenu FT metoda samo nad delom ET sistema, koji je definisan kao najznačajniji za konkretnu primenu, na osnovu puteva prostiranja greške [14, 20]. Ovakav kompromis dveju tehnika uvodi novi stepen slobode kod projektovanja sistema i rezultuje arhitekturom za koju se kaže da je parcijalno oporna na defekte (eng. *Partial Defect Tolerance - PDT*) [20].

Kako bi se definisale greške nastale usled defekata i formalno postavile osnovne tehnike za analizu propagacije grešaka, u radu će biti definisana metrika otkaza sistema. Otkaz sistema će biti precizno definisan u odnosu na Hamingovu i Euklidsku metriku, a parcijalna visokopouzdanost definisana kao osobina arhitekture za izabranu metriku [20]. Parcijalna visokopouzdanost biće definisana formalnim matematičkim jezikom, a uz definiciju parcijalne visoke pouzdanosti biće predložene i sistematske tehnike za markiranje dela arhitekture od najvećeg značaja za datu primenu [14]. Tehnika za projektovanje parcijalno visokopouzdanih polja biće ilustrovana na primeru semi-sistoličke arhitekture BP FIR filtra [14, 21, 22]. Matematički aparat i postupak za analizu arhitekture biće izložen i dat u formi algoritma. Na primeru relativno složene topologije BP filtra biće ilustrovan postupak za projektovanje parcijalno visokopouzdanih sistema.

Očigledno je da PDT dizajn uvek donosi uštedu resursa u poređenu sa arhitekturom visokopouzdanom u celosti [14]. Međutim, ključno pitanje na koje će u ovom radu biti dat odgovor je da li je, i ako jeste kada je PDT dizajn isplativiji u odnosu na ET dizajn? Formalnim matematičkim jezikom biće definisan prinos ET i PDT sistema [11, 20]. Analiza isplativosti PDT



tehnike projektovanja u odnosu na ET tehniku će biti izvedena i detaljno prikazana. Analiza će biti izvedena za polja u opštem obliku, bez obzira na topologiju i izabranu metriku otkaza. U radu će biti dat opšti oblik obrasca za izračunavanje parametara proizvodnog procesa za koje PDT daje veći prinos u odnosu na ET tehniku. Korišćenjem predloženog matematičkog aparata za analizu PDT sistema biće pokazano da za verovatnoće pojave defekata tipične za nano-tehnologiju predloženi PDT pristup porjektovanju arhitektura donosi veću ekonomsku dobit u odnosu na ET pristup.

Matematički aparat za analizu prinosa predložene tehnike biće ilustrovan na primeru ortogonalnog semi-sistoličkog polja BP FIR filtra [20]. Za topologiju BP FIR filtra će biti određena verovatnoća pojave defekta počev od koje PDT tehnika daje veći prinos u odnosu na ET tehniku. Verovatnoća će biti data u formi funkcije kod koje su parametri dimenzije polja.

Kako bi se odgovorilo na pitanje kako topologija polja utiče na isplativost primene PDT metoda, u ovom radu detaljno će biti analizirane različite topologije sistema i pravci prostiranja grešaka kroz njih. U cilju povećanja prinosa fabrikacije, biće razmotrena veličina i pozicija dela arhitekture gde je potrebno primeniti metode visoke pouzdanosti, za zadatu verovatnoću greške procesa proizvodnje i nivo greške koji se u primeni algoritma može tolerisati. Da bi se pojednostavile analitičke operacije nad grafovima u matricnoj reprezentaciji, analiza za Euklidsku metriku biće izvedena u tropskoj algebri. Biće dat set lema i teorema sa dokazima u tropskoj algebri koje opisuju oblike težinskih matrica najkraćih puteva, težinskih mapa uticaja i mapa uticaja grafova. Korišćenjem rezultata dobijenih tropskom algebrom nad grafovima propagacije biće dat postupak projektovanja heksagonalnog parcijalno visokopouzdanog polja za množenje brojeva.

Teoretski rezultati dobijeni projektovanjem parcijalno visokopouzdanih polja različitih topologija biće eksperimentalno provereni. Uz svaki eksperimentalno dobijeni rezultat biće data i teoretski određena vrednost. U cilju implementacije prototipa na FPGA čipu, sva predložena polja će biti opisana u VHDL-u, a koncept parcijalne visoke pouzdanosti će biti ugrađen u odgovarajuće VHDL funkcije.

Naučni doprinos ovog rada obuhvata: uvođenje metrike otkaza, formalnu matematičku definiciju PDT tehnike, definisanje metoda za markiranje najznačajnijeg dela arhitekture u odnosu na Hamingovu i Euklidsku metriku, uvođenje tropske algebre u postupak projektovanja visokopouzdanih sistema, ilustraciju metoda na primeru projektovanja PDT semi-sistoličkog BP FIR filtra, izvođenje opšteg oblika parametara procesa proizvodnje za koje je prinos od primene predložene tehnike pozitivan, ilustracija i analiza metoda na različitim topologijama, proveru metoda u slučaju kompleksnije primene na dva nivoa, primenu PDT metoda u VHDL opisima sa ciljem automatizacije

generisanja PDT polja.

Ovaj rad sadrži sedam glava.

U drugoj glavi biće dat pregled sistema za digitalnu obradu signala. Biće prikazani karakteristični DSP algoritmi i uobičajeni način reprezentacije DSP algoritama. Biće dat kratak pregled operacije konvolucije i korelacije, digitalnog filtriranja i diskretne kosinusne transformacije. Biće opisan način reprezentacije DSP algoritama blok dijagramima, grafom toka signala, grafom toka podataka i grafom zavisnosti. Karakteristične klase i topologije sistoličkih polja i semi-sistoličkih polja će biti opisane. Detalno će biti prikazana implementacija DSP algoritama na sistoličkim i semi-sistoličkim poljima, koja će biti korišćena u narednim glavama kao osnova za projektovanje PDT polja. Biće prikazano 1D jednosmerno i dvosmerno polje za množenje matrice i vektora, ortogonalno semi-sistoličko *bit-plane* polje za digitalno FIR filtriranje i heksagonalno sistoličko polje za množenje brojeva. U cilju ilustracije rezultata implementacije polja na FPGA čipu ukratko će biti predstavljena arhitektura Xilinx Spartan2 FPGA čipa, na kome će biti izvršena implementacija svih polja datih u ovom radu. Rezultati FPGA implementacije osnovnih polja biće dati na kraju druge glave.

Treća glava posvećena je tehnikama za projektovanje visokopouzdanih sistema. Na početku glave biće data uobičajena terminologija koja je u upotrebi kod potencijalno defektnih sistema. Od tehnika za povećanje pouzdanosti sistema detaljno će biti opisana hardverska redundansa, koja je korišćena za projektovanje PDT sistema, a informaciona, vremenska i softverska redundansa biće pomenute radi celovitosti izlaganja. Za hardverske metode će biti data analiza pouzdanosti sistema. Biće definisana minimalna jedinica zamene, i u odnosu na minimalnu jedinicu zamene biće izvedena verovatnoća otkaza celog sistema u zavisnosti od verovatnoće pojave defekta na jednom podsistemu. Poslednji deo glave biće posvećen konceptu tolerancije sistema na greške i nepreciznim izračunavanjima, kao uvodnom delu u projektovanje parcijalno visokopouzdanih sistema. Neprecizna izračunavanja će biti ilustrovana na primeru telefonske sekretarice sa defektnom memorijom.

U četvrtoj, petoj i šestoj glavi biće izloženi originalni rezultati ovog rada. Četvrto poglavlje biće posvećeno formalno-matematičkoj definiciji parcijalne visoke pouzdanosti. Biće uvedena metrika otkaza i formalno definisana kao Hamingova i Euklidska metrika otkaza. Definicija PDT sistema biće data kroz set definicija i lema, ilustrovanih primerima. U okviru lema biće opisane i dokazane osobine neophodne za projektovanje PDT sistema. Tranzitivno zatvaranje, kao operacija neophodna za određivanje uticaja čvorova grafa, detaljno će biti prikazana i ilustrovana na primeru tranzitivnog zatvaranja relativno složene topologije BP FIR filtra. Postupak određivanja tranzitivnog zatvaranja BP arhitekture detaljno će biti prikazan. Tehnika projek-

tovanja parcijalno visokopouzdanih arhitektura biće uobličena u formu algoritma i prikazana dijagramom toka. Na kraju trećeg poglavlja detaljno će biti prikazan postupak projektovanja parcijalno visokopouzdanog BP FIR filtra.

U petoj glavi biće dat odgovor na pitanje o isplativosti predložene tehnike projektovanja. Na primeru će biti pokazano da primena tehnika za povećanje pouzdanosti, u cilju smanjenja uticaja defekta nastalih u procesu proizvodnje, može dovesti do ekonomskih gubitaka. Prvi deo glave biće posvećen definiciji i određivanju prinosa ET i PDT tehnike, dok će u centralnom delu glave biti dat postupak određivanja parametara procesa fabrikacije čipova za koje je prinos fabrikacije sistema u predloženoj tehnici bolji u odnosu na postojeće tehnike projektovanja. Biće formiran opšti oblik obrasca koji daje ovu zavisnost za bilo koju topologiju i obim sistema. Biće pokazano da je za obim sistema kakav je tipičan za nano-tehnologiju, zavisno od topologije sistema, predloženi metod može dati određeni doprinos. Na primeru topologije BP FIR filtra biće ilustrovani rezultati vezani za prinos u fabrikaciji parcijalno visokopouzdanog arhitekture.

Uticaj topologije na prinos fabrikacije parcijalno visokopouzdanog polja biće razmatran je u šestoj glavi. Prvi deo glave biće posvećen jednodimenzionalnim poljima. Za jednosmerno i dvosmerno 1D polje za množenje matrice i vektora biće data analiza uticaja i propagacije različitih tipova grešaka kroz polje. Projektovanje parcijalno visokopouzdanog 1D polja biće izvedeno za izabrane klase defekata koji se mogu javiti u polju. Procedura projektovanja biće primenjena nad jednosmernom i dvosmernom 1D polju. Projektovana PDT polja biće predstavljena DFG-om. Analiza prinosa će biti izvedena. Rezultati FPGA implementacije polja će biti dati. U drugom delu ove glave biće predstavljeno parcijalno visokopouzdanog heksagonalno polje za množenje brojeva. Polje će biti projektovano u odnosu na Hamingovu i Euklidsku metriku otkaza. Biće data analiza propagacije grešaka nastalih od različitih tipova defekata. PDT polje projektovano u odnosu na Hamingovu metriku će biti analizirano sa aspekta dva tipa defekata. Da bi se pojednostavile analitičke operacije nad grafovima u matričnoj reprezentaciji, analiza za Euklidsku metriku biće izvedena u tropskoj algebri. Biće dat pregled osnovnih definicija tropske algebre i izveden set lema koje povezuju tropsku algebru sa algoritmima za obilazak grafova. U tropskoj algebri će biti izvedeni opšti oblici mapa težinskih uticaja greške date topologije. Heksagonalno PDT polje će biti predstavljeno DFG-om. Analiza uticaja tolerantnosti aplikacije na visokopouzdanog deo PDT arhitekture biće detaljno prikazana. Projektovano polje će biti opisano u VHDL-u i implementirano na FPGA čipu. Rezultati implementacije će biti dati tabelarno i grafički.

## Glava 2

# Digitalna obrada signala

Sistemi za digitalnu obradu signala imaju mnoge prednosti u odnosu na analogne sisteme. Digitalni sistemi su robusniji u odnosu na analogne u pogledu dozvoljenih temperaturnih varijacija, kao i u odnosu na ostale izvore šumova. Preciznost digitalnih reprezentacija moguće je kontrolisati brojem cifara u reprezentaciji digitalnog signala. Osim toga, DSP tehnike je moguće primeniti na taj način da se šum i greške nastale u sistemu eliminišu. Nasuprot ovome, kod analognih sistema šumovi i greške nastale u sistemu se pojačavaju zajedno sa signalom. Prednost digitalnih signala u odnosu na analogne je u tome da je digitalne signale moguće pamtiti, prenositi i obrađivati praktično bez pojave grešaka. Analogna obrada je neizbežna kod sistema koji zahtevaju ekstremno malo zauzeće resursa na čipu i ekstremno nisku potrošnju koja se ulaže u obradu signala. Mnogi kompleksni sistemi realizuju se u vidu digitalnih sistema sa visokom preciznošću i pouzdanošću, i visokim odnosom signala i šuma (eng. *Signal to Noise Ratio - SNR*) [1].

DSP sistemi mogu se realizovati korišćenjem programabilnih procesora ili specijalno projektovanim VLSI kolima. Uvek je cilj maksimizirati performanse uz očuvanje troškova na što nižem nivou. U kontekstu projektovanja digitalnih kola, performanse se mere količinom iskorišćenih hardverskih resursa (površina ili oblast na čipu); brzinom izvršavanja algoritma, koja zavisi od propusnosti samog sistema (eng. *throughput*); količinom utrošenje energije za izvršenje određenog izračunavanja. Kod DSP sistema koristi se i dodatna mera za ocenu kvaliteta sistema. Mera kvaliteta sistema koja se uvodi je preciznost sistema, tj. stepen kvantizacije signala. Ova mera se uvodi i kod digitalnog filtriranja, jer su u nekim slučajevima digitalnih filteri sa velikom greškom zaokruživanja neupotrebljivi [1, 23, 22].

Dve važne karakteristike izdvajaju DSP sisteme od ostalih sistema. To su zahtev za obradom u realnom vremenu, kao i obrada vođena podacima. Ukoliko je propusnost sistema manja od brzine uzorkovanja signala, tada

nove vrednosti moraju biti memorisane (baferovane), što u krajnjoj instanci zahteva bafer neograničene veličine. Druga osobina implicira da je međuoperacije moguće izvršiti tek kada su podaci dostupni, tako da su sistemi globalno sinhronizovani tokom podataka, a lokalno sistemskim taktom.

U ovom poglavlju dat je pregled tipičnih DSP algoritama. Izloženi su ciljevi i izazovi u projektovanju DSP sistema, kao i uobičajena notacija i reprezentacija DSP algoritama. U poglavlju 2.1 prikazano je nekoliko tipičnih DSP algoritama. Poglavlje 2.2 posvećeno je reprezentaciji DSP algoritama, dok je u poglavlju 2.3 data klasifikacija procesorskih polja. U cilju ilustracije implementacije DSP algoritama na procesorskim poljima prikazana su procesorska polja koja su korišćena u narednim poglavljima ove disertacije. Poglavlje 2.4 bavi se implementacijom procesorskih polja na FPGA arhitekturi. FPGA implementacija polja korišćena je u toku izrade ovog rada za praktičnu proveru teoretskih rezultata.

## 2.1 DSP algoritmi

Neki DSP algoritmi i njihova tipična primena prikazani su u tabeli 2.1 [1, 24]. U ovom poglavlju prikazano je par ključnih DSP izračunavanja u koja se ubrajaju konvolucija, korelacija, digitalno filtriranje i diskretna kosinusna transformacija.

### 2.1.1 Konvolucija

Matematički posmatrano, konvolucija je binarna operacija, koja funkcije  $f(t)$  i  $g(t)$  preslikava u funkciju  $(f * g)(t)$ . Rezultujuća funkcija posmatra se kao modifikovana verzija originalnih funkcija i koristi se u analizi funkcija. Konvolucija je originalno bila poznata pod nazivom *faltung*, ili prevedeno sa nemačkog, savijanje. Operaciju konvolucije prvi je predstavio nemački matematičar Gustav Doetsch [25].

**Definicija 2.1 (Konvolucija)** *Konvolucija funkcija  $f$  i  $g$  označava se sa  $(f * g)$  i definisana je kao*

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) \cdot g(t - \tau) d\tau. \quad (2.1)$$

Vizuelizacija operacije konvolucije prikazana je na slici 2.1. Koraci konvolucije, dati na slici 2.1, su: (1) predstavljanje funkcija u smislu nove promenljive  $\tau$ ; (2) transponovanje funkcije  $g(t) \rightarrow g(-t)$ ; (3) dodavanje vremenskog pomeraja  $\tau$  koji će se kretati po  $\tau$  osi; (4) početi od  $-\infty$  i pomerati

$\tau$  do  $+\infty$ , na mestima gde se funkcije preklapaju odrediti integral proizvoda u toj tački. Konvolucija funkcija  $f * g$  (slika 2.1 a i b) prikazana na slici 2.1(f).

Ukoliko je funkcija  $f(t)$  jedinični impuls  $h(t)$ , tada je rezultat ovog procesa sama funkcija  $g(t)$  koja se u tom slučaju naziva jedinični impulsni odziv sistema.

**Definicija 2.2 (Diskretna konvolucija)** *Konvolucija dveju diskretnih sekvenci  $h(n)$  i  $x(n)$  definiše se na sledeći način:*

$$y(n) = x(n) * h(n) = \sum_{i=-\infty}^{\infty} x(i) \cdot h(n - i). \quad (2.2)$$

Rezultat funkcije (2.2) u vremenskom trenutku  $n$ ,  $y(n)$  predstavlja proizvod odgovarajućih elemenata niza  $x(i)$  i  $h(-i + n)$ , sumiran na intervalu  $-\infty < i < \infty$ .

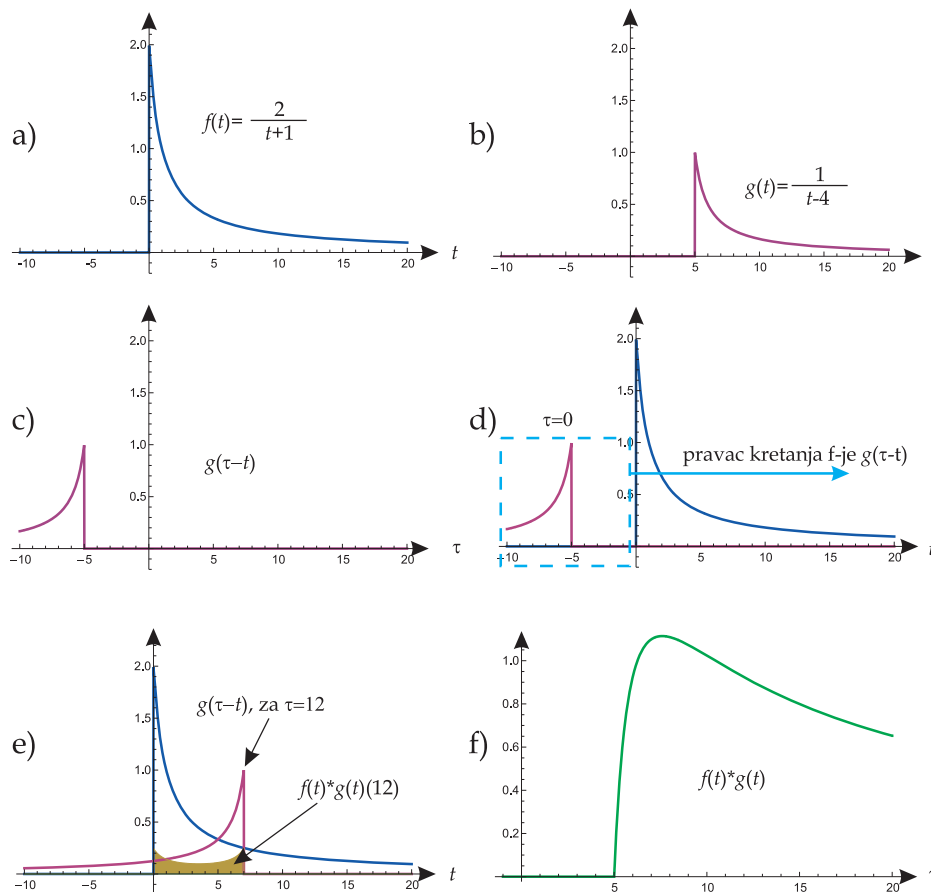
Konvolucija se koristi u analizi linearnih, vremenski invarijantnih sistema, koji su u potpunosti definisani svojim jediničnim impulsnim odzivom  $h(n)$ . Jedinični odziv dobija se kada se na ulaz sistema dovede Dirakov impuls  $\delta(t)$  [26]. Znajući jedinični impulsni odziv sistema, izlazna sekvenca

DSP algoritam	Primena
Kodiranje i dekodiranje govora	Mobilna telefonija, sistemi za komunikaciju, multimedija, zaštita i kriptografija
Kriptovanje i dekrptovanje govora	Mobilni telefoni, personalni uređaji za komunikaciju, digitalni bežični telefoni, sigurne komunikacije
Prepoznavanje glasa	Napredni korisnički interfejsi, multimedijalne radne stanice, roboti i automatizovane aplikacije, mobilni telefoni, komunikacioni sistemi, bežični telefoni
Sinteza govora	Multimedijalni računari, napredni korisnički interfejsi, robotika
Algoritmi za pripremu signala za prenos	Mobilni telefoni, komunikacioni sistemi, bežični sistemi, emitovanje digitalnog radio i TV signala, računarske komunikacije, bežični prenos podataka, navigacija, modemi, sigurne komunikacije
Uklanjanje šumova i smetnji	Profesionalna audio obrada
Obrada zvuka	Audio aplikacije, profesionalna audio obrada
Obrada slika	Digitalne kamere, obrada digitalnih video signala, multimedija, video aplikacije
Zračenje	Navigacija, radar, sonar, obaveštajni signali
Uklanjanje eha	Sistem zvučnik i mikrofona, modemi, telefonski aparati

Tabela 2.1: Primeri tipičnih DSP algoritama i njihove primene

linernih, vremenski invarijantnih sistema može se odrediti kao konvolucija ulazne sekvence  $x(n)$  i jediničnog impulsnog odziva  $h(n)$ .

Kada jedinični impulzni odziv sistema sadrži konačan broj elemenata različitih od nule sistem se naziva sistem sa konačnim odzivom (eng. *Finite Impulse Response - FIR*). U suprotnom, sistem se naziva sistem sa beskonačnim odzivom (eng. *Infinite Impulse Response - IIR*). Na primer, sistem za određivanje prosečne vrednosti koristi se za analizu u mnogim oblastima tehnike. Ovaj sistem određuje prosečnu vrednosti dela sekvence podataka, gde je deo čiji se prosek određuje određen parametrima, a svakom podatku



Slika 2.1: Vizuelizacija konvolucije: a) funkcija  $f(t)$ ; b) funkcija  $g(t)$ ; c) predstavljanje funkcije  $g$  u zavisnosti od promenljive  $\tau$ ; d) funkcije  $f$  i  $g$  za pomeraj  $\tau = 0$ , integral proizvoda je 0; e) dalje pomeranje funkcije  $g$ , prikazano za  $\tau = 12$ , presek je osenčan i predstavlja vrednost konvolucije u tački  $\tau = 12$ ; f) konvolucija funkcija  $f$  i  $g$

je pridružena težina. Impulsni odziv ovog sistema je

$$h(n) = \frac{1}{M_1 + M_2 + 1} \sum_{i=-M_1}^{M_2} \delta(n - i)$$

i predstavlja FIR sistem. Primer IIR sistema je akumulatorski sistem sa jediničnim impulsnim odzivom

$$h(n) = \sum_{i=-\infty}^n \delta(i)$$

Ovo je tzv. step funkcija koja je jednaka 1 za  $n \geq 0$  i jednaka 0 za  $n < 0$ .

Za sistem se kaže da je uzročni (eng. *causal*), odnosno da pamti istoriju, ukoliko se za izračunavanje vrednosti  $y(n_0)$  koriste ulazni elementi  $x(i)$ ,  $i \leq n_0$ . Jedinični impulsni odziv uzročnog linearnog vremenski invarijantnog sistema je  $h(n) = 0$ , za  $n < 0$  [1].

### 2.1.2 Korelacija

Korelacija je matematička operacija koja nalazi veliku primenu u digitalnim komunikacijama i drugim sistemima za obradu signala. U digitalnoj obradi signala korelacija (eng. *cross-correlation*) je mera sličnosti dva talasna oblika predstavljena funkcijama, gde je jedan talasni oblik vremenski pomeren. Ovo je takođe poznato i pod nazivom pokretni proizvod (eng. *sliding dot product*), ili unutrašnji proizvod (eng. *inner-product*). Korelacija se često koristi kako bi se u signalu dužeg trajanja odredila mesta pojavljivanja određenog oblika, zadatog kraćom funkcijom. Kao takva, korelacija nalazi primenu i u sistemima za prepoznavanje uzoraka, elektronskoj tomografiji, kripto-analizi, i sl. [27].

**Definicija 2.3 (Korelacija)** *Neka su  $f$  i  $g$  neprekidne funkcije na intervalu  $(-\infty, +\infty)$ , tada je korelacija, u oznaci  $(f \star g)$ , definisana kao*

$$(f \star g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f^*(\tau)g(t + \tau)d\tau, \quad (2.3)$$

gde je  $f^*$  konjugovana funkcija kompleksne funkcije  $f$ .

Korelacija je po prirodi slična konvoluciji, s tim da se u slučaju konvolucije signal prvo transponuje, pomera i množi sa drugim signalom, dok se kod korelacije vrši pomeranje i množenje, bez transponovanja. Na primer, razmotrimo dve funkcije  $f$  i  $g$  koje se razlikuju jedino po pomeraju na  $x$  osi. Korelacijom je moguće odrediti za koliko su funkcije pomerene jedna u odnosu na



drugu. Izraz (2.3) pomera funkciju  $g$  po  $x$  osi, računajući integral proizvoda funkcija za svaki pomeraj. Kada je poklapanje funkcija najveće, vrednost  $(f \star g)$  je najveća. Razlog za ovo je taj da je prilikom poklapanja pozitivnih oblasti površina proizvoda je najveća. Isto važi i za proizvod negativnih oblasti, koje takođe prilikom poklapanja povećavaju površinu (proizvod dva negativna broja) [27].

**Definicija 2.4 (Diskretna korelacija)** *Korelacija dveju diskretnih sekvenci  $a(n)$  i  $x(n)$  definisana je kao*

$$y(n) \stackrel{\text{def}}{=} \sum_{i=-\infty}^{\infty} a(i) \cdot x(n+i). \quad (2.4)$$

Operacija korelacije (2.4) može se predstaviti pomoću operacije konvolucije na sledeći način:

$$y(n) = \sum_{i=-\infty}^{\infty} a(-i)x(n-i) = a(-n) * x(n). \quad (2.5)$$

U slučaju kada su nizovi  $a(n)$  i  $x(n)$  konačne dužine  $N$ , tj. funkcije imaju vrednost različitu od nule za  $i = 0, 1, \dots, N-1$ , diskretna korelacija ima oblik

$$y(n) = \sum_{i=0}^{N-1} a(i)x(n+i) \quad (2.6)$$

za  $n = -N+1, -N+2, \dots, -1, 0, 1, \dots, N-2, N-1$ . Jednačina (2.6) takođe može biti napisana i u formi proizvoda matrice i vektora kao

$$\begin{bmatrix} y(-3) \\ y(-2) \\ y(-1) \\ y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & x(0) \\ 0 & 0 & x(0) & x(1) \\ 0 & x(0) & x(1) & x(2) \\ x(0) & x(1) & x(2) & x(3) \\ x(1) & x(2) & x(3) & 0 \\ x(2) & x(3) & 0 & 0 \\ x(3) & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a(0) \\ a(1) \\ a(2) \\ a(3) \end{bmatrix} \quad (2.7)$$

za  $N = 4$  [1].

Proširenjem jednodimenzionalne jednačine (2.4) na dve dimenzije dobija se oblik korelacije koji je čest u digitalnoj obradi slika. Tada jednačina (2.7) iz oblika proizvoda matrice i vektora prelazi u oblik proizvoda matrica [28].

### 2.1.3 Digitalno filtriranje

Digitalni filtri su važna klasa linearnih vremenski invarijantnih sistema koji modifikuju frekventne osobine ulaznog signala  $x(n)$ . Osobine digitalnog filtra u potpunosti su opisane njegovim jediničnim odzivom  $h(n)$ , ili njegovim frekventnim odzivom  $H(e^{j\omega})$ . Takođe, osobine digitalnog filtra moguće je opisati i prenosnom funkcijom  $H(z)$ , ili diferencnim jednačinama. Jedinični odziv i frekventni odziv oslikavaju ponašanje filtra u vremenskom i frekventnom domenu respektivno, dok diferencne jednačine eksplicitno prikazuju operacije koje su neophodne za implementaciju filtra [1].

**Definicija 2.5 (Digitalni filter)** *Linearni, vremenski invarijantni, uzročni filter definiše se na sledeći način:*

$$y(n) = - \sum_{i=1}^N a_i y(n-i) + \sum_{i=0}^{M-1} c_i x(n-i). \quad (2.8)$$

Za slučaj  $a_i = 0$  za  $1 \leq i \leq N$  jednačina (2.8) postaje

$$y(n) = \sum_{i=0}^{M-1} c_i x(n-i), \quad (2.9)$$

što predstavlja jednačinu filtra sa konačnim impulsnim odzivom (FIR filter), čiji je jediničnim impulsni odziv

$$h(i) = c_i$$

za  $0 \leq i \leq M-1$ , i  $h(i) = 0$  u suprotnom. Vrednosti  $c_i$  nazivamo koeficijentima FIR filtra, a  $M$  označava broj koeficijenata filtra. Za filter (2.9) se kaže da je filter reda  $M$  sa koeficijentima  $c_i, i = 0, 1, \dots, M-1$ .

Izračunavanje (2.9) je nerekurzivno. Ukoliko je bar jedna vrednost  $a_i \neq 0$  za  $1 \leq i \leq N$ , jednačina (2.8) predstavlja rekurzivno izračunavanje kod koga izračunavanje vrednosti  $y(n)$  zahteva prethodne vrednosti izlaznog niza  $y$ , i naziva se rekurzivni filter. Broj elemenata jediničnog odziva rekurzivnog filtra koji su različiti od nule je beskonačan, pa se kaže da je rekurzivni filter sistem sa beskonačnim impulsnim odzivom (IIR). Izbor FIR ili IIR filtra u DSP sistemima zavisi od primene DSP algoritma [1, 23].

Prilikom projektovanja filtra propusnika dela frekventnog opsega, poželjno je imati filter koji ima ujednačeno slabljenje u datom opsegu frekvenca koje propušta [29]. Linearnost prenosne funkcije filtra u frekventnom domenu odgovara ujednačenom kašnjenju i pojačanju u vremenskom domenu, čime se minimizira fazna distorzija izlaznog signala. S tim u vezi, poželjno je

projektovati filter sa idealno, ili bar približno linearnom fazom. FIR filtri sa linearnom fazom su posebno interesantni zbog simetrije svog jediničnog odziva. Zbog simetrije, moguće je izvršiti optimizaciju ovih filtra i prepoloviti broj potrebnih izračunavanja u implementaciji [1]. Na primer, neka jedinični odziv linearnog filtra reda  $M$  zadovoljava uslov simetrije

$$h(n) = h(M - n).$$

Tada, FIR filter sedmog reda sa linearnom fazom, čiji je jedinični impulsni odziv dat sa

$$h(0) = h(6) = c_0, h(1) = h(5) = c_1, h(2) = h(4) = c_2, h(3) = c_3,$$

može biti predstavljen jednačinom

$$\begin{aligned} y(n) &= c_0x(n) + c_1x(n-1) + c_2x(n-2) + c_3x(n-3) \\ &+ c_0x(n-6) + c_1x(n-5) + c_2x(n-4) = \\ &= c_0(x(n) + x(n-6)) + c_1(x(n-1) + x(n-5)) + \\ &+ c_2(x(n-2) + x(n-4)) + c_3x(n-3), \end{aligned}$$

i može se implementirati na način prikazan na slici 2.2(a), ili na način prikazan na slici 2.2(b) korišćenjem 4 množača, 6 sabirača i 6 registara. DSP algoritam prikazan na slici 2.2(b) predstavlja arhitekturu sa emisionom linijom, s obzirom na to da se ulazni podaci uvode istovremeno u sve množače [1].

### 2.1.4 Diskretna kosinusna transformacija

Diskretna kosinusna transformacija (eng. *Discrete Cosine Transform - DCT*) predstavlja transformaciju signala iz prostornog u frekventni domen. Ova transformacija često je u upotrebi kod sistema za kodiranje video signala i primenjuje se na matricnoj reprezentaciji dela slike, dimenzija  $N \times N$  [16]. Cilj ove transformacije kod kompresije video signala je uklanjanje visokih frekvencija, čime se ne gubi značajno na kvalitetu slike, jer je ljudsko oko manje osetljivo na više frekvencije [16, 17]. Rezultat transformacije je matrica  $Y$  dimenzija  $N \times N$  koja se naziva matrica koeficijenata.

**Definicija 2.6 (Diskretna kosinusna transformacija)** *Diskretna kosinusna transformacija je preslikavanje  $DCT : X_{N \times N} \rightarrow Y_{N \times N}$  definisano kao*

$$Y = A \cdot X \cdot A^T, \quad (2.10)$$

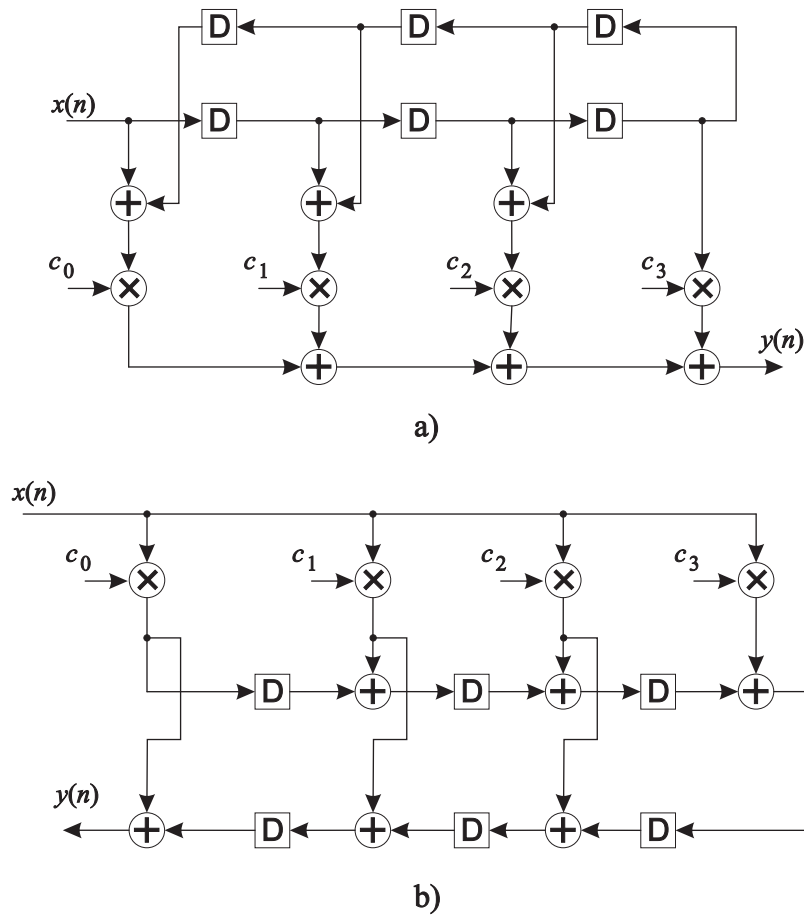
gde su elementi matrice  $A = [a_{i,j}]$  oblika

$$a_{i,j} = C_i \cos \frac{(2j+1) \cdot i \cdot \pi}{2N}, \quad C_i = \begin{cases} \sqrt{\frac{1}{N}}, & i = 0 \\ \sqrt{\frac{2}{N}}, & i > 0 \end{cases},$$

a matrica  $A^T$  transponovana matrica matrice  $A$ .

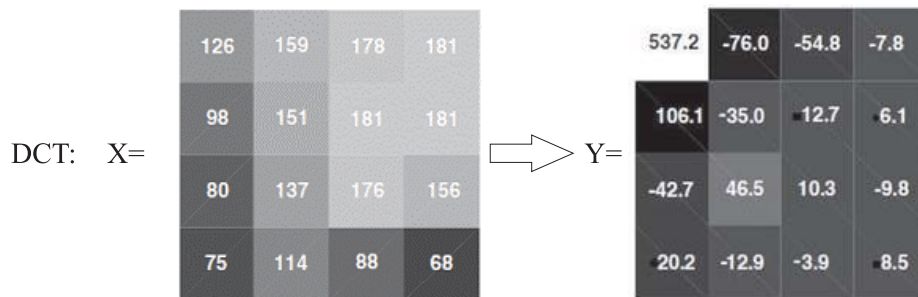
Koeficijent  $a_{0,0}$  predstavlja jednosmernu komponentu signala i naziva se DC (eng. *Direct Current*) koeficijent, a frekvencija se povećava sa povećanjem indeksa  $i$  i  $j$  [16]. Inverzna transformacija transformacije (2.10) je

$$X = A^T \cdot Y \cdot A.$$



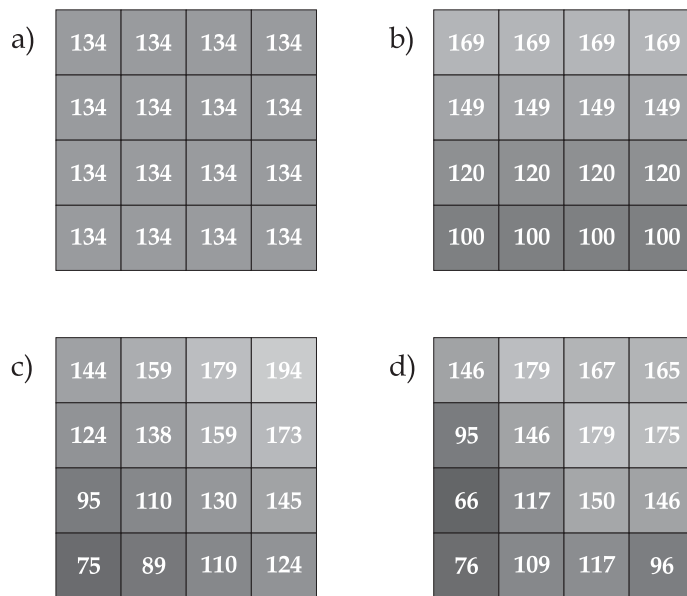
Slika 2.2: Blok dijagram linearnog FIR filtra sedmog reda: a) bez emisione linije, b) sa emisionom linijom

Na slici 2.4 prikazan je primer DCT nad matričnim oblikom dela fotografije dimenzije 4x4 i odgovarajući skup DCT koeficijenata.



Slika 2.3: Primer DCT transformacije, matrični oblik dela fotografije dimenzije 4x4 i odgovarajući skup DCT koeficijenata

Na slici 2.4 prikazan je rezultat inverzne DCT kada su iz matrice koeficijenata uzeti u obzir redom: samo DC koeficijent, 2 koeficijenta, 3 i 5 koeficijenata. Sa slike 2.4d se može uočiti velika sličnost rekonstruisanog bloka u odnosu na originalni blok sa slike 2.3, iako je u ovom slučaju inverzna transformacija izvršena nad samo 31% koeficijenata (5 od ukupno 16).



Slika 2.4: Primer inverzne DCT transformacije: a) samo nad DC koeficijentom, b) 2 koeficijenta, c) 3 i d) 5 koeficijenata

## 2.2 Reprezentacija DSP algoritama

DSP algoritmi se predstavljaju programima koji se neprekidno izvršavaju. Na primer filter trećeg reda opisan je neprekidnim programom

$$y(n) = c_0x(n) + c_1x(n-1) + c_2x(n-2), \quad 2 \leq n < \infty. \quad (2.11)$$

**Definicija 2.7 (Iteracija)** *Jedno izvršenje svih operacija algoritma naziva se iteracija. Jedan period iteracije je vreme koje je potrebno za izvršenje jedne iteracije algoritma. Brzina iteracije je recipročna vrednost periode iteracije.*

U toku svakog iterativnog perioda FIR filter trećeg reda (2.11) obrađuje jednu novu ulaznu reč, izvršava 3 množenja i 2 sabiranja, i generiše jednu izlaznu reč.

Karakteristika DSP sistema je takođe i brzina uzorkovanja signala (eng. *sampling rate*), koje se poistovećuje sa propusnošću sistema (eng. *throughput*). Propusnost sistema izražava se kao broj obrađenih uzoraka u sekundi.

**Definicija 2.8 (Kritični put)** *Kritični put je najduži put signala kroz kombinacionu mrežu između dva memorijska elementa, ili ulaza i izlaza.*

Vreme propagacije signala direktno je proporcionalno dužini putanje, tako da je brzina rada kola uslovljena kritičnim putem. Kritični put kod sinhronih sistema određuje minimalnu periodu takta kojom je moguće taktovati sistem.

**Definicija 2.9 (Latencija)** *Latencija, ili kašnjenje sistema definiše se kao razlika u vremenima kada je generisan izlaz i vremena kada je odgovarajući ulazni podatak uveden u sistem.*

Kod kombinacionih mreža latencija se obično izražava kao broj gejtova kroz koje signal prolazi, dok se kod sekvencijalnih kola latencija izražava u broju taktnih intervala. Generalno, brzina taktovanja kola kod sekvencijalnih DSP sistema nije jednaka brzini uzorkovanja ulaznog signala i brzini generisanja izlaznog rezultata.

DSP algoritmi se na visokom nivou opisuju matematičkim formulama, gde je bitnije definisati funkcionalnost sistema od interne strukture pojedinih operacija. Na arhitekturnom nivou neophodno je ove matematičke izraze konvertovati u neki jezik za opis ponašanja sistema (eng. *Hardware Description Language - HDL*), ili grafičku reprezentaciju.

Jezici za opis ponašanja sistema dele se na aplikativne, preskriptivne i opisne jezike [1, str. 32]. Kod aplikativnih jezika sistem se opisuje skupom jednačina čije promenljive i zavisnosti između promenljivih pokazuju tok DSP

algoritma. Aplikativni jezici često se koriste za opis DSP algoritama. Primer aplikativnog jezika je *Silage* jezik [30], koji je u upotrebi kao jezgro alata za sintezu *Chatedral* [31]. Preskriptivni jezici eksplicitno opisuju redosled naredbi dodele. Primer ovog tipa jezika su jezici visokog nivoa kao što Pascal, C, Fortran i sl. U skorije vreme, za opis DSP algoritama se koriste opisni jezici kojima se opisuje struktura sistema. Primer ovog tipa jezika su Verilog i VHDL [32]. Nešto više o jezicima za opis sistema i o implementaciji prototipa sistema na osnovu opisa biće reči u poglavlju 2.4.

Grafičke reprezentacije su korisne za istraživanje i analizu osobina DSP algoritma, koje imaju za cilj eksploataciju paralelizma između različitih podzadataka, i još važnije, za mapiranje DSP algoritma na hardver. Pomoću grafičkih reprezentacija arhitekture moguće je projektovati nezavisno od tehnologije.

U daljem tekstu opisane su grafičke reprezentacije DSP algoritama. Od grafičkih reprezentacija predstavljeni su blok dijagrami, grafovi toka signala (eng. *Signal Flow Graph - SFG*), grafovi toka podataka (eng. *Data Flow Graph - DFG*) i grafovi zavisnosti (eng. *Dependence Graph - DG*).

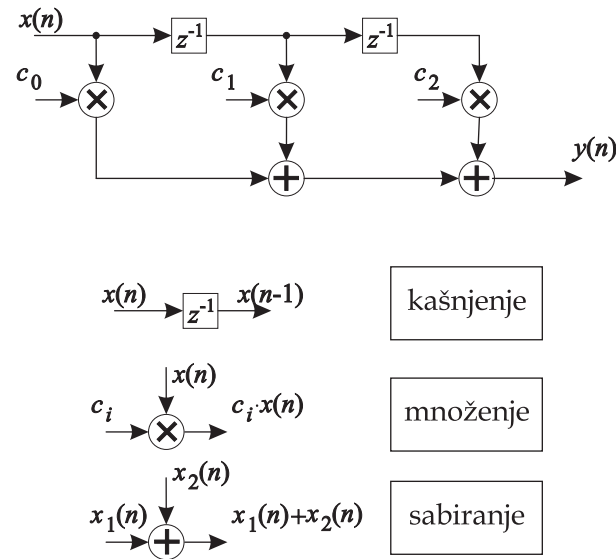
## 2.2.1 Blok dijagrami

Za grafičku reprezentaciju DSP sistema najčešće se koriste blok dijagrami. Blok dijagrame čine funkcionalni blokovi povezani usmerenim granama, koje predstavljaju smer toka podataka. Grane mogu sadržati elemente za kašnjenje. Blok dijagramom sistem je moguće opisati na različitim nivoima apstrakcije [1].

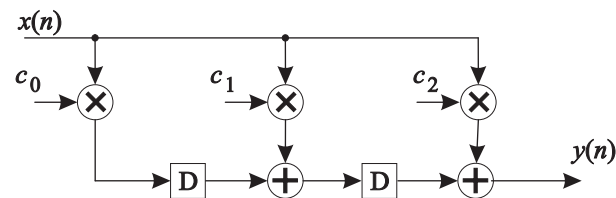
FIR filter sa 3 koeficijenta, dat jednačinom (2.11), moguće je opisati blok dijagramom kao na slici 2.5. Blok dijagram sa slike 2.5 sastoji se od dva tipa funkcionalnih blokova, sabirača i množaca. Kola za kašnjenje takođe se mogu smatrati funkcionalnim blokovima, jer se na čipu implementiraju registrima. Element za kašnjenje označen je sa  $z^{-1}$  (ili **D**, od engleske reči *Delay*), slika 2.5.

Sistem je moguće predstaviti različitim blok dijagramima, od kojih svaki predstavlja različitu implementaciju iste funkcionalnosti. Na primer, filter trećeg reda sa slike 2.5 može se predstaviti i blok dijagramom sa slike 2.6. Arhitektura FIR filtra trećeg reda sa slike 2.6 je arhitektura sa emisionom linijom, kod koje se ulazni podatak  $x(n)$  uvodi u sve množače istovremeno.

Blok dijagram je konkretni model koji predstavlja tačan opis implementacije funkcionalnosti sistema. Različite arhitekture koje realizuju istu funkcionalnost mogu biti projektovane i predstavljene zasebnim blok dijagramima.



Slika 2.5: Blok dijagram FIR filtra trećeg reda



Slika 2.6: Blok dijagram FIR filtra trećeg reda sa emisionom linijom za uvođenje ulaznih podataka

### 2.2.2 Graf toka signala

Graf toka signala (eng. *Signal Flow Graph - SFG*) je kolekcija čvorova i usmerenih grana koji se koriste za predstavljanje toka signala kroz arhitekturu [33]. Usmerena grana  $(j, k)$  predstavlja transformaciju signala od čvora  $j$  do čvora  $k$ . Sa ulaznim signalom u čvoru  $j$  i izlaznim signalom u čvoru  $k$ , grana  $(j, k)$  predstavlja linearnu transformaciju signala iz čvora  $j$  u signal u čvoru  $k$ . SFG se koriste za predstavljanje i analizu linearnih digitalnih mreža, a naročito su pogodni za reprezentaciju strukture digitalnih filtra. SFG se ne mogu koristiti za predstavljanje sistema kod kojih se brzine smplovanja signala razlikuju u različitim delovima sistema (eng. *multirate*). Kod digitalnih mreža, grane obično označavaju linearno pojačanje signala (određišni kraj grane je proizvod konstante i ulaznog signala), ili kašnjenja (određišni



kraj grane je zakašnjeni ulazni signal).

Sabirači i množaci predstavljaju se čvorovima sa višestrukim ulaznim granama i jednom izlaznom granom. Postoje dva specijalna tipa čvora. To su početni čvor i završni čvor. Početni čvor je čvor bez ulaznih grana i koristi se za označavanje ulaza signala u graf, dok je završni čvor bez izlaznih grana i označava izlaze iz sistema [1, 33].

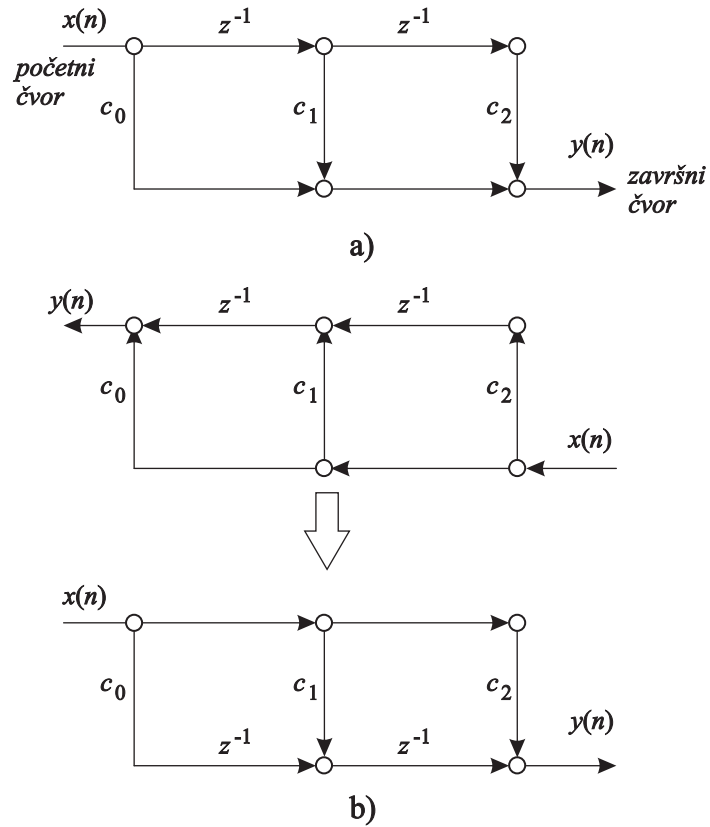
SFG direktne forme FIR filtra trećeg reda predstavljenog jednačinom (2.11) dat je na slici 2.7(a), gde grane bez težinske oznake predstavljaju jediničnu transformaciju (prenos signala bez izmena). Linearni SFG može se transformisati bez promene funkcije sistema. Transponovanje grafa toka signala je operacija primenljiva na sisteme sa jednim ulazom i jednim izlazom [1]. Operacija transponovanja vrši se tako što se smer svih grana okrene uz zamenu mesta početnom i završnom čvoru. Težine grana, kao i lokacije kašnjenja ostaju nepromenjene. Na slici 2.7(b) prikazan je rezultat transponovanja SFG-a grafa sa slike 2.7(a). Potrebno je napomenuti da transponovana forma SFG-a (slika 2.7) odgovara arhitekturi FIR filtra trećeg reda sa emisionom linijom, prikazanoj blok dijagramom na slici 2.6 [1].

### 2.2.3 Graf toka podataka

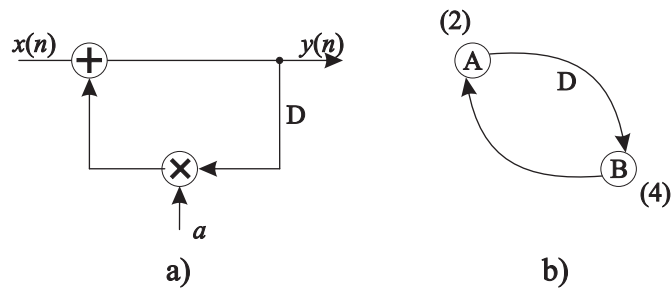
Kod DFG reprezentacije čvorovi predstavljaju izračunavanja, odnosno funkcije ili podzadatke, dok usmerene grane predstavljaju komunikaciju između čvorova, kojom su opisani tokovi podataka u algoritmu. Svaka grana ima pridružen nenegativan broj kašnjenja. Na primer, na slici 2.8(b) prikazan je graf toka podataka DSP algoritma sa slike 2.8(a), kojim se izračunava  $y(n) = ay(n-1) + x(n)$ , gde čvor  $A$  predstavlja sabiranje, a čvor  $B$  množenje. Grana od čvora  $A$  do čvora  $B$  (u oznaci  $A \rightarrow B$ ) sadrži jedan element za kašnjenje, dok grana od čvora  $B$  do čvora  $A$  ( $B \rightarrow A$ ) ne sadrži kašnjenja. Svakom čvoru se pridružuje broj vremenskih jedinica koliko traje obrada podataka u čvoru. Kako je prikazano na slici 2.8, izvršenje operacija čvora  $A$  traje 2 vremenske instance, dok je za izvršenje operacija čvora  $B$  potrebno 4 vremenske instance. Iz ovoga se može zaključiti da je iterativni period ovog algoritma 6 vremenskih instanci [1].

DFG oslikava izvršenje DSP algoritma vođeno podacima. Čvor kod DFG reprezentacije izvršava operaciju svaki put kada su novi podaci dostupni na ulaznim granama. Iz ovakve osobine čvorova dolazi se do zaključka da više čvorova istovremeno mogu izvršavati svoje operacije, što vodi konkurentnosti. Takođe, za DFG reprezentaciju se može reći da grane grafa pokazuju zavisnost po podacima između čvorova. Ukoliko grana ima nulto kašnjenje zavisnost između čvorova je *intra-iterativna*. Za čvorove koji su povezani granom sa nenultim kašnjenjem kaže se da su *inter-iterativno* zavisni. Obe

zavisnosti zajedno daju redosled izvršavanja operacije koje čvorovi implementiraju. Na primer, grana ( $A \rightarrow B$ ) na slici 2.8(b) ukazuje na to da  $k$ -ta



Slika 2.7: SFG FIR filtra trećeg reda: (a) direktna forma; (b) transponovani FIR filter

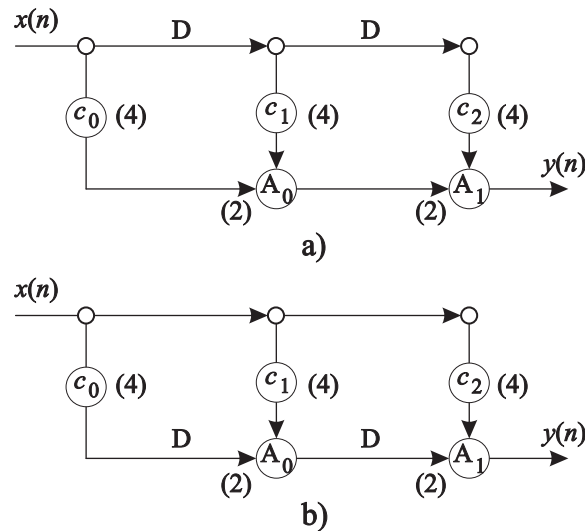


Slika 2.8: (a) DFG za izračunavanje  $y(n) = ay(n - 1) + x(n)$ ; (b) konvencionalna DFG reprezentacija.

iteracija čvora  $A$  mora biti završena pre početka  $(k + 1)$  iteracije čvora  $B$ . Grana  $(B \rightarrow A)$  ukazuje da operacija čvora  $B$  u  $k$ -toj iteraciji mora biti završena pre početka operacije  $A$  u istoj toj iteraciji.

DFG se kao i blok dijagrami mogu koristiti za predstavljanje kako linearnih sistema sa jedinstvenim vremenom uzorkovanja, tako i nelinearnih sistema sa različitim vremenom uzorkovanja signala u različitim delovima kola. Opis algoritma blok dijagramom, gde su funkcije za obradu signala predstavljene blokovima, bliži je hardverskoj implementaciji. DFG sa druge strane opisuje tok podataka između podsistema. Različite reprezentacije istog algoritma mogu se dobiti transformisanjem DFG-a [34, 35, 36, 37]. Na slici 2.9 prikazan je DFG FIR filtra trećeg reda opisanog jednačinom (2.11). DFG FIR filtra trećeg reda u direktnoj formi, predstavljen na osnovu blok dijagrama sa slike 2.5, dat je na slici 2.9(a). Graf toka podataka transponovane forme FIR filtra trećeg reda sa slike 2.6 dat je na slici 2.9(b).

DFG reprezentacija DSP algoritama je najčešće u upotrebi za sintezu na visokom nivou apstrakcije, kada je potrebno razmotriti konkurentnost zadataka, tj. nivo paralelizma aplikacije [1].



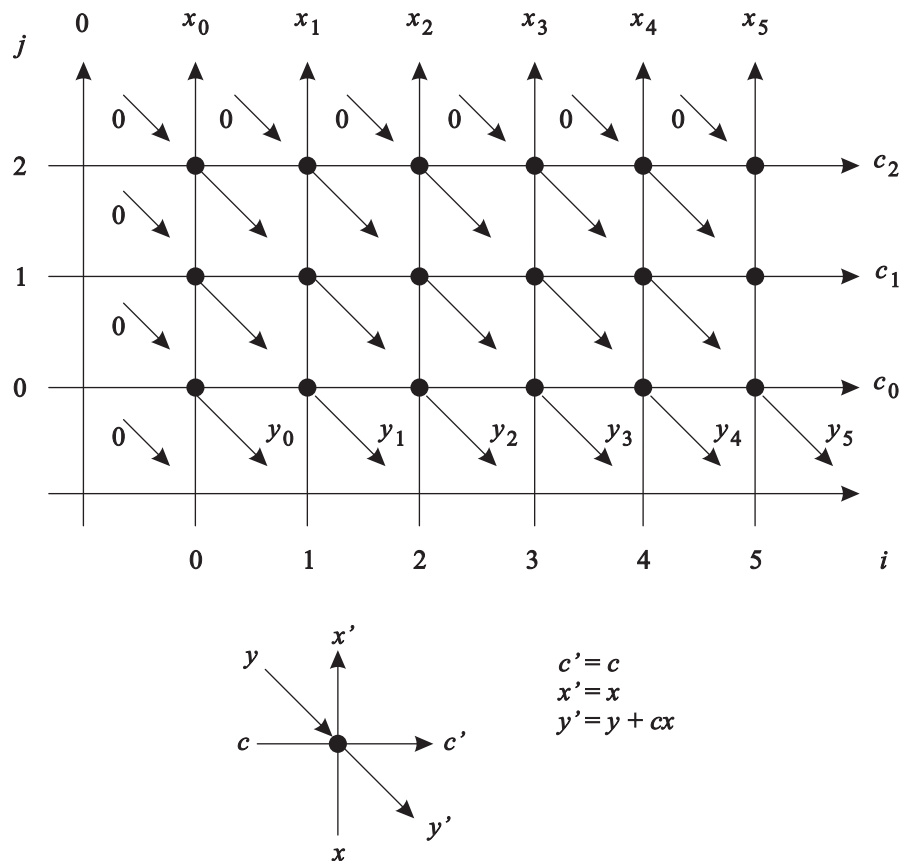
Slika 2.9: Filter trećeg reda: (a) direktna forma, (b) transponovana forma

## 2.2.4 Graf zavisnosti

Graf zavisnosti (DG) je usmereni graf koji prikazuje zavisnosti između podataka u algoritmu. Čvorovi DG-a ilustruju izračunavanja, a grane između

čvorova pokazuju zavisnosti među izračunavanjima. DG sadrži po jedan čvor za svako izračunavanje. Bitno je naglasiti da se kod DG-a jedan čvor izvršava (koristi) samo jednom [1]. DG reprezentacija je slična DFG reprezentaciji, jer eksplicitno pokazuje zavisnosti čvorova po podacima. Razlika je u tome da čvorovi DFG-a predstavljaju operacije u jednoj iteraciji, a njihove operacije se izvršavaju sa ponavljanjem, dok DG sadži sva izračunavanja algoritma. DFG sadrže memorijske elemente koji prenose međurezultate iz jedne iteracije u drugu. DG ne sadrže memorijske elemente. Dvodimenzionalni DG FIR filtra sa 3 koeficijenta, datog jednačinom (2.11), prikazan je na slici 2.10.

Grafovi zavisnosti su najčešće u upotrebi kod projektovanja sistoličkih polja, gde je pomoću ovih grafova moguće projektovati nove topologije eksploatacijom paralelizma između operacija [1].



Slika 2.10: DG reprezentacija filtra trećeg reda

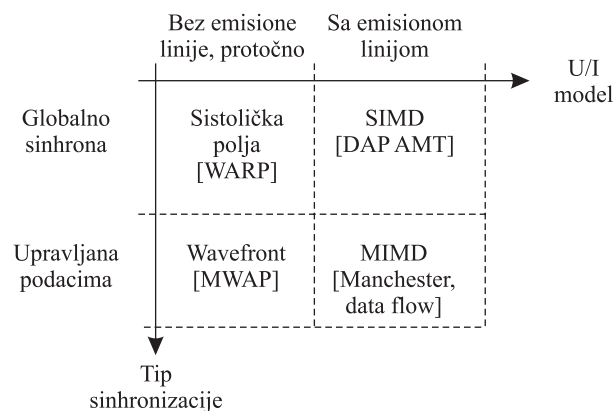
## 2.3 Sistolička polja

Moderne DSP aplikacije zahtevaju veliku propusnu moć arhitektura, koja je često praćena intenzivnim saobraćajem sa memorijom [38]. Sistemi sa sekvencijalnom obradom su često neadekvatni po pogledu brzine obrade za primenu u modernim DSP aplikacijama koje zahtevaju obradu signala u realnom vremenu. Alternativno arhitekturno rešenje koje može zadovoljiti zahteve obrade signala u realnom vremenu su višeprosorski sistemi i procesorska polja specijane namene.

Na početku ovog poglavlja dat je osvrt na klase procesorskih polja. Sistoličkim poljima, kao jednoj klasi procesorskih polja, posvećeno je više pažnje u smislu ilustracije topologija i principa mapiranja DSP algoritama na polja koja su korišćene dalje u ovom radu za proveru rezultata. Na kraju ovog poglavlja dati su primeri sistoličkih polja za množenje matrice i vektora, množenje dva broja i digitalno FIR filtriranje, koji realizuju DSP algoritme prikazane u poglavlju 2.1.

### 2.3.1 Procesorska polja

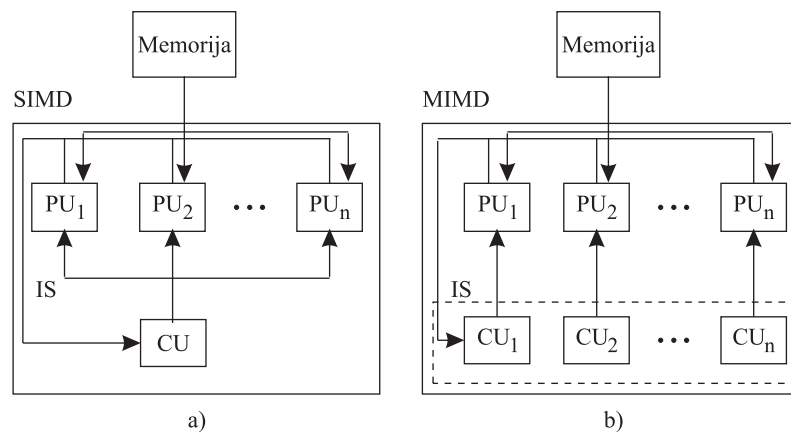
Moderne paralelne računare možemo na osnovu strukture svrstati u tri klase: vektorske procesore, višeprosorske sisteme i procesorska polja [3]. Prve dve klase spadaju u procesore opšte namene, dok je treća klasa fokusirana na aplikacijama specijalne namene. Podela paralelnih računara na osnovu tipa sinhronizacije delova sistema i korišćenog modela za unos podataka, odnosno izlaz rezultata (U/I), prikazana je na slici 2.11.



Slika 2.11: Klasifikacija paralelnih računara na osnovu tipa sinhronizacije delova sistema i korišćenog U/I modela

Sistolička i *wavefront* polja (slika 2.11) određena su konkurentnom obradom podataka kroz arhitekturu sistema. Kod sistoličkih polja sinhronizacija lokalnih instrukcija izvedena je na globalnom nivou. Za razliku od sistoličkih polja, kod *wavefront* polja obrada podataka je upravljana samim podacima (eng. *data driven*). Sistolička i *wavefront* polja nemaju globalnu magistralu, već je komunikacija lokalizovana na komunikaciju susednih procesnih elemenata. Sistolička i *wavefront* polja nazivaju se i VLSI procesorska polja. Glavna razlika ova dva tipa polja i SIMD i MIMD polja (slika 2.11) je u komunikaciji među procesnim elementima [3].

Sistemi sa emisionom linijom dele se na sisteme sa jedinstvenom instrukcijom nad višestrukim podacima (eng. *Single Instruction Multiple Data - SIMD*) i sisteme sa višestrukim instrukcijama nad višestrukim podacima (eng. *Multiple Instruction Multiple Data - MIMD*). SIMD sistemi su takvi sistemi kod kojih svi procesni elementi izvršavaju istu instrukciju, ali nad različitim podacima. Primer ovakvog sistema je MMX multimedijalni koprocesor kod Pentium računara. MIMD sistemi su sistemi kod kojih procesni elementi u istom trenutku mogu da izvršavaju različite instrukcije. Cena ove osobine je zasebna upravljačka jedinica za svaki procesni element (slika 2.12). Kod ova dva sistema koristi se pristup distribucije podataka i/ili upravljanje preko emisione linije (eng. *broadcast*). MIMD polja predstavljaju nadskup svih pomenutih mogućnosti obrade, kao što su n distribuirano upravljanje, izračunavanja upravljana podacima, podaci dobijeni preko emisione linije i mogućnost korišćenja deljive memorije.



Slika 2.12: Tipičan primer arhitekture procesorskih polja a) tipa SIMD b) tipa MIMD

Eksploatacija protočnosti jedna je od glavnih odlika sistoličkih polja, zbog

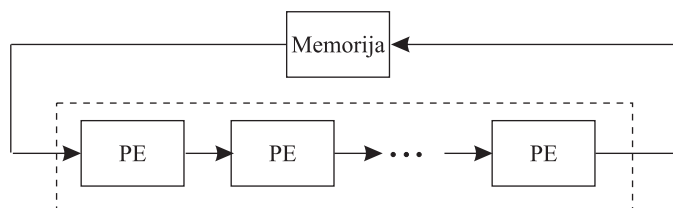
koje ovaj tip procesorskih polja nalazi veoma veliku primenu kod DSP sistema. Protočnošću se uspešno rešava problem *Von Neumannovog* ograničenja, koje je rezultat komunikacije između delova sistema i odnosi se na razliku u brzinama rada procesorskih jedinica i memorije [39]. Ova razlika uspešno se prevazilazi protočnošću, gde se jednom uvedeni podatak višestruko koristi.

### 2.3.2 Klase i topologije sistoličkih polja

Sistolička polja su vrsta procesorskih polja bez emisione linije (slika 2.11) [40]. Glavna odlika ove grupe procesorskih polja je lokalizacija komunikacije između susednih procesorskih jedinica. Razlika između sistoličkih polja i *wavefront* polja ogleda se u načinu međusobne sinhronizacije procesorskih jedinica. Koncept sistoličkih polja uveli su Kung i Leiserson 1978. godine [41].

**Definicija 2.10 (Sistolički sistem)** *Sistolički sistem je mreža procesora kod koje procesni elementi ritmički obavljaju izračunavanja i prosleđuju međurezultate narednom procesnom elementu.*

Analogija je učinjena sa sistolom. Sistolom u medicini nazivaju se kontrakcije srčanih pregrada koje ritmički pumpaju krv. Kod sistoličkih polja podaci se "pumpaju" i prosleđuju na regularan način. Kod tipične *Von Neumann* arhitekture u memoriju se podatak upisuje nakon obrade u bilo kojoj procesorskoj jedinici, što stvara usko grlo u obradi. Kod sistoličkih polja, kada se podatak jednom uvede iz memorije u sistem, podatak se efikasno koristi na taj način što se prenosi iz ćelije u ćeliju sve dok obrada traje. Ovaj proces je prikazan na slici 2.13. Bez obzira što vreme obraćanja memoriji može biti i za red veličine duže od vremena obrade u okviru jedne procesorske jedinice, podaci se višestruko obrađuju u polju pre pristupa memoriji, tako da je na ovaj način moguće otkloniti usko grlo koje predstavlja komunikacija sa memorijom.



Slika 2.13: Osnovna konfiguracija sistoličkih polja

Faktori koji su doveli do ekspanzije primene sistoličkih polja jesu jednostavan i regularan dizajn, konkurentnost i lokalizovana komunikacija, kao i mogućnost balansiranja obima izračunavanja sa obimom komunikacije sa memorijom [2].

### Topologije sistoličkih polja

Kriterijumi koji definišu odnos cena/performanse sistoličkog polja su efikasnost mapiranja algoritma, regularnost, skalabilnost, rekonfigurabilnost, efikasnost komunikacionih linija, tolerantnost na greške, nivo protičnosti i paralelizma koji se može postići, i broj tokova podataka kroz polje [42]. Može se s pravom reći da svi pobrojani kriterijumi direkto zavise od načina na koji se procesni elementi međusobno povezuju [43]. Način povezivanja procesnih elemenata naziva se topologijom polja. Veliki broj topologija za povezivanje procesnih elemenata je predložen, gde svaka topologija ima ciljni skup algoritama koji se efikasno mogu izvršavati na njoj [43, 44, 45, 46, 47]. Važno je napomenuti da se bilo koji DSP algoritam može izvršavati različitim topologijama polja, manje ili više efikasno, zavisno od samog DSP algoritma, kao i od topologije polja na kojoj se algoritam izvršava. Ovaj proces se naziva *emulacija*, i može biti vremenska i prostorna. Autori su u [43] pokazali efikasnost emulacije DSP algoritama na različitim topologijama polja.

Po broju potrebnih dimenzija prostora za opis topologije polja, sistolička polja dele se na

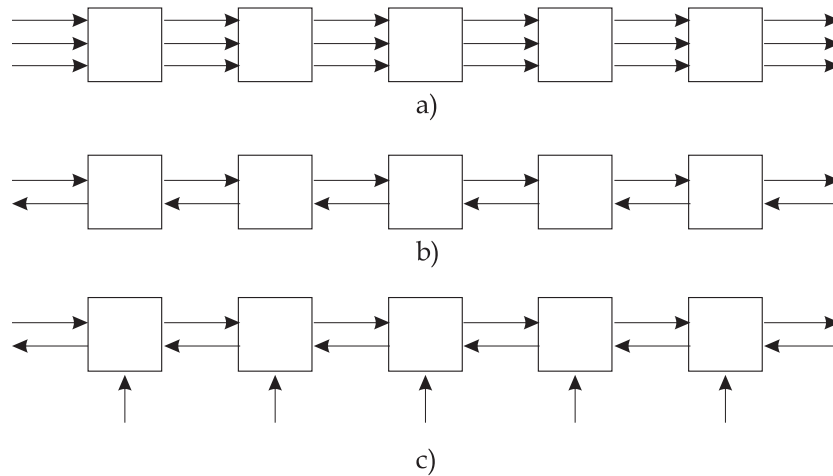
- jednosimenzionalana (1D), odnosno *linearna* polja,
- dvodimenzionalna (2D), ravna ili *planarna* polja, i
- višedimenzionalna polja.

Linearna polja u zavisnosti od broja tokova podataka dalje delimo na jednosmerna, dvosmerna i višesmerna (slika 2.14) [2].

### Planarna polja

Planarna polja su polja kod kojih su za opis veza između procesorskih elemenata potrebne dve dimenzije. Projektovanje topologije ovakvih polja ima čvrstu matematičku osnovu [43, 48]. Naime, problem povezivanja procesnih elemenata polja u ravni svodi se na problem pokrivanja površine različitim geometrijskim oblicima [49]. Jedan od prvih istraživača koji se još pre skoro četiri veka bavio problemom pokrivanja površina bio je nemački matematičar i astronom Johannes Kepler (1571.-1630.).





Slika 2.14: Linearna sistolička polja: (a) jednosmerno sa tri toka podataka, (b) dvosmerno - dva toka podataka u suprotnim smerovima, (c) trosmerno.

**Definicija 2.11 (Zakrpe)** *Pokrivanje površine* (eng. tiling) je graf ( $T$ ) koji se sastoji od familije skupova čvorova  $v_{i,j}$ , u oznaci  $S_i \in (v_{i,1}, v_{i,2}, \dots)$ . Skupovi  $S_i$  nazivaju se zakrpe.

**Definicija 2.12 (Pokrivanje površine)** *Za površinu se kaže da je pokrivena* ukoliko su zakrpe takve da se međusobno ne preklapaju, a pokrivanje je takvo da na površini ne postoje pukotine.

Za svako pokrivanje  $T$  važi da svaki čvor pokrivanja mora pripadati bar trima zakrpama, a grana koja povezuje dva čvora predstavlja granicu između tačno dve zakrpe. Ovakva pokrivanja se nazivaju i pokrivanja "grana-do-grane" (eng. *edge-to-edge*).

Postoji neograničeno mnogo različitih kombinacija za pokrivanje površine, što se jednostavno može pokazati izvođenjem jednog pokrivanja iz drugog transformacijama grafa [49]. Međutim, ukoliko se pokrivanje ograniči na zakrpe koje su konveksni jednakostranični poligoni, situacija se značajno menja. Ovakva pokrivanja su još iz radova Keplera poznata kao "**manje-više regularna**" (eng. *more-less regular*) pokrivanja [49, str. 227].

Da bi površina bila pokrivena, za svaki čvor  $v_{i,j}$  mora da važi trivijalno pravilo geometrije da je zbir uglova koji zaklapaju sve stranice poligona koje dodiruju čvor  $v_{i,j}$  jednak  $360^\circ$  ( $2\pi$ ). Kod regularnih poligona ugao koji stranice zaklapaju je

$$\theta = \frac{(n-2)\pi}{n}. \quad (2.12)$$

Na primer, ukoliko se za pokrivanje površina koriste isključivo jednakos-tranični trouglovi, kod koga su uglovi između stranica na osnovu izraza (2.12) jednaki  $\pi/3$ , može se zaključiti da je za pokrivanje površine potrebno da svaki čvor dodiruje 6 trouglova. Uobičajena oznaka za ovakvo pokrivanje je 3.3.3.3.3.3, ili skraćeno  $3^6$ , gde broj 3 označava da se radi o trouglu. Naredna mogućnost je četiri trougla i jedan šestougao, u oznaci 3.3.3.3.6, ili skraćeno  $3^4.6$ . Jednostavno se može proveriti da postoji ukupno 17 različitih tipova čvorova koji mogu zadovoljiti uslov punog kruga, uz uslov jednakos-traničnosti poligona [49]. Ako se uzme u obzir i redosled ređanja poligona oko čvora, tada imamo 4 nove kombinacije, tako da ukupno postoji 21 kombinacija rasporeda poligona oko čvora kojima se može pokriti površina. Sve kombinacije su prikazane na slici 2.15.

Ukoliko dalje ograničimo pokrivanje tako da u obzir uzmemo ona pokrivanja koja se sastoje samo od jednog tipa čvora, može se pokazati da postoji 11 tipova čvorova kojima je uniformno moguće pokriti površinu. Dokaz za ovo nije trivijalan, a postupak dokazivanja opisno je dat u [49]. Ovakva pokrivanja nazivaju se **regularna**, ili **Arhimedova** pokrivanja. Čvorovi Arhimedovih pokrivanja naznačena su slovom **A** na slici 2.15. Arhimedova pokrivanja prikazana su na slici 2.16.

Daljim uvođenjem strožih kriterijuma uvodi se pojam strogo regularnih pokrivanja [49]. **Strogo regularna** pokrivanja su grafovi koji se izvode jedan iz drugog zamenom uloge ivica grafa i čvorova. Strogo regularni grafovi su grafovi  $3^6$ ,  $4^4$  i  $6^3$ .

**Definicija 2.13 (Regularna sistolička polja)** *Regularna sistolička polja su sistolička polja kod kojih pozicija procesnih elemenata odgovara poziciji čvorova grafa, a pozicija veza odgovara granama Arhimedovog pokrivanja.*

**Definicija 2.14 (Strogo regularna sistolička polja)** *Strogo regularno sistoličko polje (SRSP) je sistoličko polje čija topologija odgovara strogo regularnom grafu.*

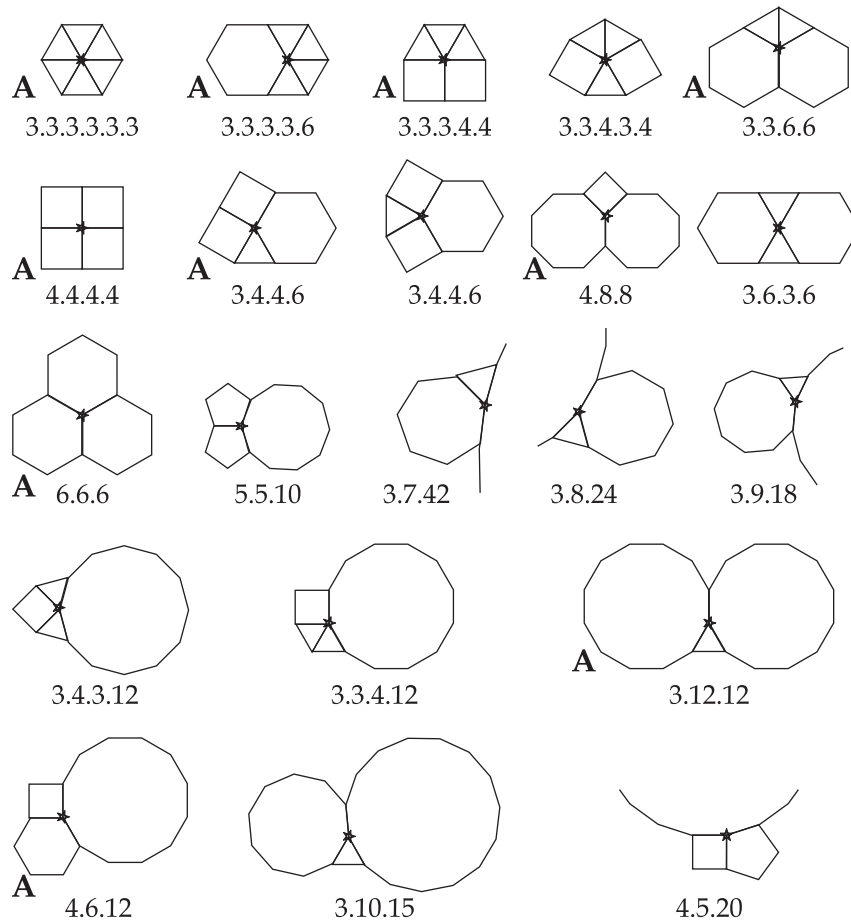
SRSP  $3^6$  naziva se heksagonalno sistoličko polje (šest veza iz svakog procesnog elementa),  $4^4$  ortogonalno, ili *mesh* polje, dok se polje  $6^3$  naziva trougaono polje. SRSP zbog svoje regularnosti i skalabilnosti nalaze veliku primenu u implementaciji DSP algoritama u VLSI tehnologiji [1, 44, 50]. Rezultati ovog rada biće ilustrovani na 1D poljima i na strogo regularnim 2D poljima.

### 2.3.3 Semi-sistolička polja

Regularnost polja je veoma strog zahtev koji je prisutan kod sistoličkih polja. Međutim, u određenim slučajevima ovaj zahtev se blago narušava. Klasa sis-

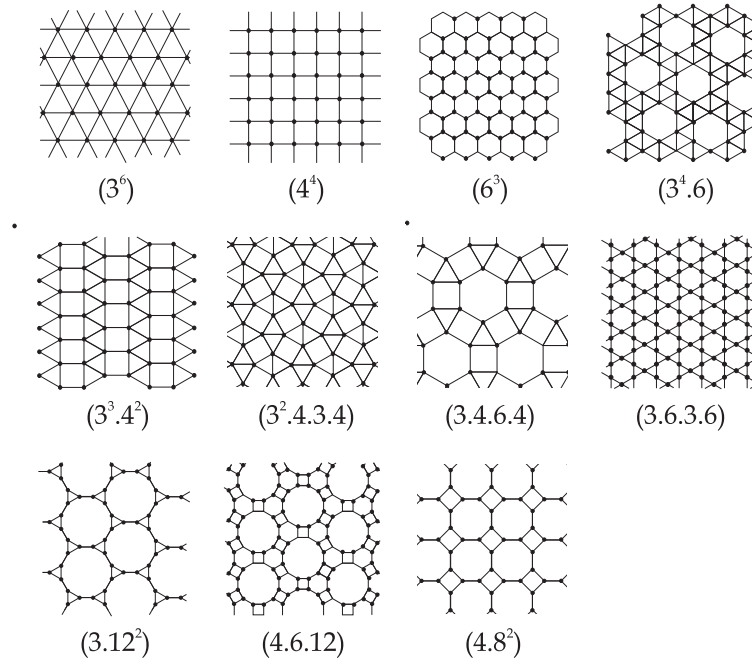
toličkih polja sa blago narušenim zahtevom po pogledu regularnosti poznata je kao klasa polu- ili semi-sistoličkih polja, i obično se ogleda u neregularnosti komunikacionih kanala u vidu emisione linije, ili potrebom za dodatnim sabiračem za formiranje krajnjeg rezultata.

Primer semi-sistoličkog polja prikazan je na slici 2.17. Na slici 2.17 prikazana je topologija polja koja na prvi pogled podseća na topologiju strogo regularnog  $3^6$  polja (slika 2.16). Međutim, veze prikazane isprekidanom linijom zapravo predstavljaju neregularnost sistoličkog polja u vidu emisione linije, kojom se podaci istovremeno uvode u procesne elemente. Na ovaj način dobijamo srtego regularno ortogonalno semi-sistoličko polje  $4^4$  sa mogućnošću uvođenja podataka u sve procesne elemente istovremeno.

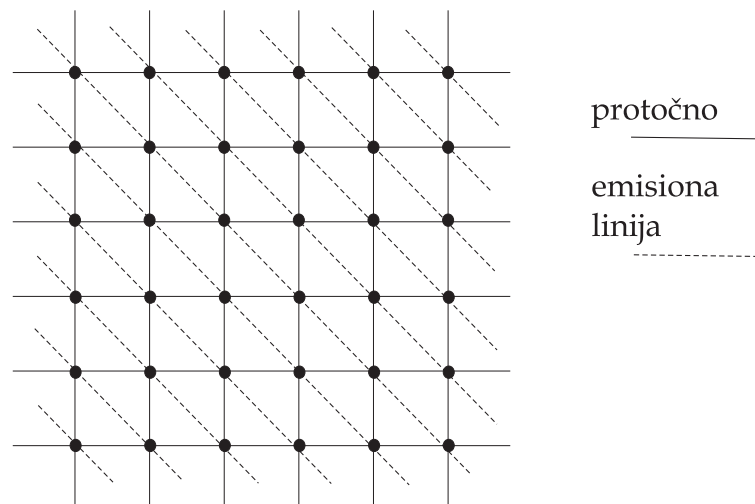


Slika 2.15: Kombinacije poligona kojima je moguće izvršiti pokrivanje površine

S jedne strane, semi-sistolička polja su nešto nepogodnija za VLSI implementaciju od sistoličkih poja [1, 44, 47]. Uslov regularnosti je narušen, a i emisiona linija predstavlja jedinstvenu tačku otkaza, tako da su nepogodnija



Slika 2.16: Arhimedova pokrivanja



Slika 2.17: Semi-sistoličko  $4^4$  polje

i po pitanju otpornosti sistema na defekte. S druge strane, ovakva topologija pruža određen nivo fleksibilnosti u implementaciji DSP algoritama, i nalazi veliku primenu, pored ostalog, u digitalnom filtriranju [22, 42, 21, 51, 52].

### 2.3.4 Primeri mapiranja DSP algoritama

Mnogi DSP algoritmi mogu biti, manje ili više, efikasno mapirani na sistolička polja [1, 43, 53, 54]. Ovi algoritmi uključuju FIR i IIR filtriranje, konvoluciju, korelaciju, interpolaciju, diskretnu Furieovu transformaciju, diskretnu kosinusnu transformaciju, i mnoge druge. Za implementaciju algoritama najčešće se koriste operacije poput množenja matrice i vektora, množenja dve matrice, rešavanja trougaonog sistema jednačina, LU dekompoziciju, i sl. [1].

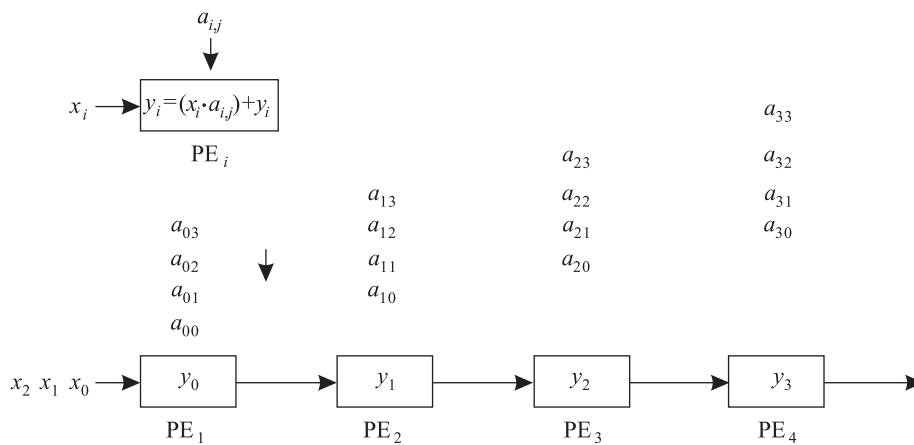
Postoji veliki broj radova koji se bave problemom projektovanja efikasnih polja [55, 56, 57, 58, 59]. U daljem tekstu biće prikazana samo ona polja koja su korišćena za proveru i ilustraciju rezultata ovog rada.

#### DSP algoritmi na 1D poljima

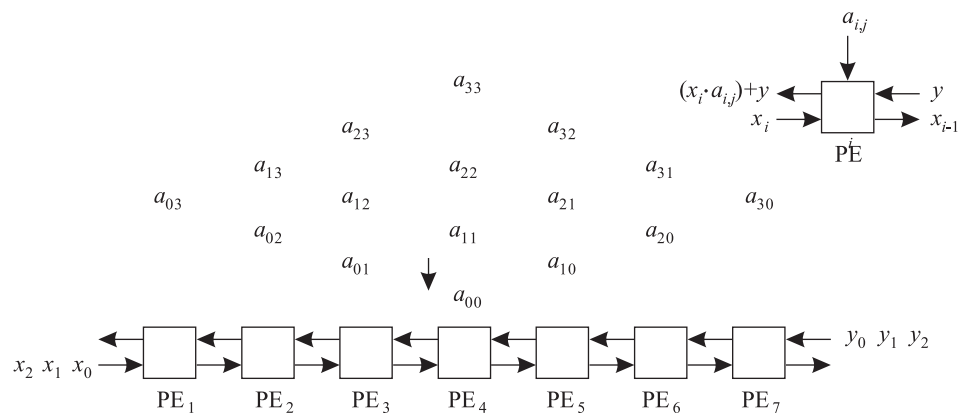
Implementacija algoritma za množenje matrice i vektora na linearnom jednosmernom polju, koja koristi jedan tok podataka u desno i jedan stalan tok za akumulaciju međurezultata kroz svaku ćeliju, prikazana je na slici 2.18 [2]. Kako se za izračunavanje koristi samo jedan tok podataka (vektor  $\{x_i\}$ ), tok elemenata matrice  $\{a_{i,j}\}$  je stacionaran u odnosu na ćeliju (slika 2.18). Procesni elementi polja su MAC (eng. *Multiply-ACcumulate*) jedinice koje u svakom taktom intervalu izračunavaju proizvod jednog elementa vektora  $\{x_i\}$  i jedog elementa matrice  $\{a_{i,j}\}$ , i dobijeni rezultat dodaju prethodno određenom međurezultatu. Tako, u prvom taktom intervalu procesni element  $PE_1$  (slika 2.18) određuje proizvod  $(x_0 \cdot a_{0,0})$ , element  $x_0$  prosleđuje narednom elementu desno, a među rezultat akumulira na inicijalnu vrednost 0. U narednom taktom intervalu  $PE_1$  određuje proizvod  $(x_1 \cdot a_{0,1})$  i dodaje ga prethodno određenom međurezultatu, tako da je sada u akumulatoru procesnog elementa  $PE_1$  vrednost  $(x_0 \cdot a_{0,0}) + (x_1 \cdot a_{0,1})$ . U istom taktom intervalu  $PE_2$  određuje i akumulira sumu  $(x_0 \cdot a_{1,0})$ . Nakon 4 taktna intervala u  $PE_1$  nalazi se prvi element rezultujućeg vektora  $y_0 = \sum_{i=0}^3 x_i \cdot a_{0,i}$ . U narednom, petom taktom intervalu  $PE_2$  sadrži drugi element rezultujućeg vektora  $\{y_1\}$ , itd. Kompletno izračunavanje proizvoda vektora sa 4 elementa i matrice dimenzija 4x4 na polju sa  $n = 4$  elementa sa slike 2.18 traje 7 taktnih intervala. Za polje sa slike 2.18 kaže se da ima stacionarne rezultate, jer je rezultate potrebno pročitati iz procesnih elemenata.

Kung i Leiserson su u [41] predložili linearno dvosmerno sistoličko polje za množenja matrice i vektora. Ovo polje prikazano je na slici 2.19.

Originalna ideja bazirana je na dva regularna toka podataka,  $\{x_i\}$  i  $\{y_i\}$ , za  $i = 1, 2, \dots, n$ , u suprotnim smerovima (slika 2.19). Kod ovog polja, dva susedna elementa niza  $\{x_i\}$  moraju biti razdvojena u vremenu za jednu vremensku jedinicu. U suprotnom, ukoliko susedni elementi nisu razdvojeni u vremenu, već se uvode sukcesivno tako da se u svakom vremenskom intervalu u polje uvodi po jedan element, element  $x_i$  nikada neće susresti elemente  $y_{i+1}$ ,  $y_{i+3}$ ,  $\dots$ , itd. Potrebna izračunavanja izvode se konkurentno u okviru ćelija, koje su i u ovom slučaju MAC jedinice. Razlika u ćelijama je ta da kod ovog polja ćelija ne zadržava akumulirani međurezultat već ga prosleđuje levom susedu (tok  $\{y_i\}$ , slika 2.19) [2]. Tok podataka kroz polje prikazan je na slici 2.20.



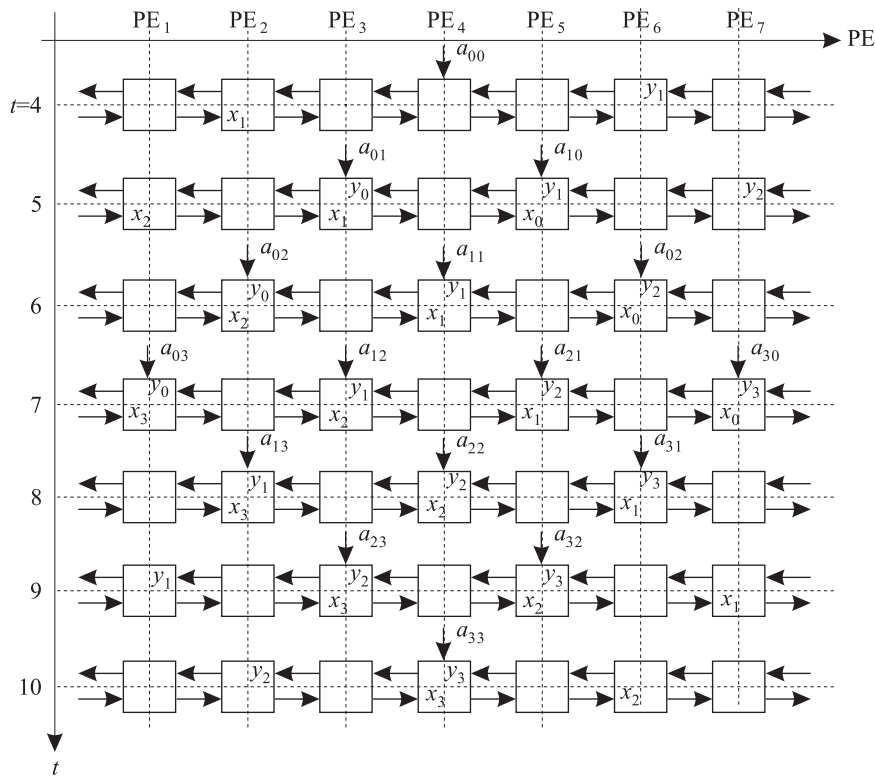
Slika 2.18: Linearno jednosmerno polje za množenje matrice i vektora ( $n=4$ )



Slika 2.19: Linearno dvosmerno polje za množenje matrice i vektora

Prvo korisno izračunavanje je u 4 taktom intervalu, ukoliko se za prvi interval uzme trenutak uvođenja  $x_0$  u procesni element PE<sub>0</sub>. Izračunavanje u ovom trenutku obavlja PE<sub>4</sub>, koji određuje proizvod ( $x_0 \cdot a_{0,0}$ ) (slika 2.20). Prvi podatak rezultujućeg vektora  $y_0$ , proizvoda vektora  $\{x_i\}$  sa 4 elementa i matrice  $\{a_{i,j}\}$  dimenzija  $4 \times 4$  na polju sa slike 2.19 sa 7 procesnih elemenata, dostupan je na izlazu polja nakon 7 taktovih intervala (slika 2.20). Elementi rezultujućeg vektora dostupni su na izlazu na svaka 2 taktovna intervala. Poslednji element  $y_3$  dostupan je nakon  $7 + (2 \cdot 3) = 13$  taktovnih intervala od početka uvođenja vektora  $\{x_i\}$  (još 3 elementa na svaka 2 taktovna intervala, počev od 7. intervala kada se dobija prvi rezultat).

Procesori jednosmernog polja sa slike 2.18 rade efikasnije u odnosu na dvosmernu implementaciju algoritma zbog postojanja aktivnih i neaktivnih vremenskih trenutaka kod dvosmerne implementacije sa slike 2.19. Kod dvosmernog polja je, međutim, jednostavnije i jeftinije čitanje podataka iz polja, a vezano za izlazni tok može se reći i da je polje bolje po pitanju skalabilnosti.

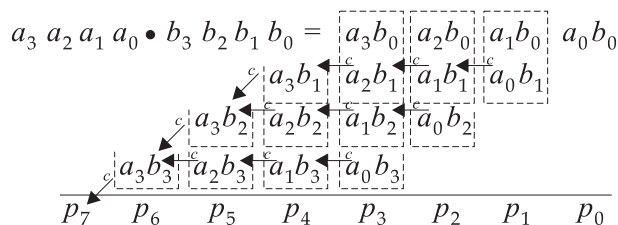


Slika 2.20: Tok podataka kroz linearno dvosmerno polje za množenje matrice i vektora

### Primer planarnog sistoličkog polja

U cilju ilustracije planarnih sistoličkih polja, u ovom poglavlju prikazana je implementacija algoritma za množenje brojeva na sistoličkom polju sa heksagonalnom ( $3^6$ ) topologijom i emulacija ovog algoritma na trougaonom ( $6^3$ ) polju.

Za algoritam za množenje brojeva izabran je *ripple carry* algoritam [1]. Množenje dva četvorobitna binarna broja  $p = a \cdot b$  prikazano je na slici 2.21. Oznakom  $a_i, i = 0, 1, 2, 3$ , na slici 2.21 predstavljen je bit težine  $2^i$  broja  $a$ . Isprekidanim linijama na slici 2.21 grupisana su sabiranja kod kojih se javlja prenos. Sabiranja se obavljaju između odgovarajućih parcijalnih prenosa po vertikali, tako da svaka vertikala rezultuje jednim bitom rezultata  $p_i, i = 0, 1, \dots, 7$ . Tok prenosa predstavljen je strelicama sa oznakom  $c$  na strelici. Množenje  $(a_0 \cdot b_0)$ , kako ovde ne postoji i sabiranje, ne daje prenos.



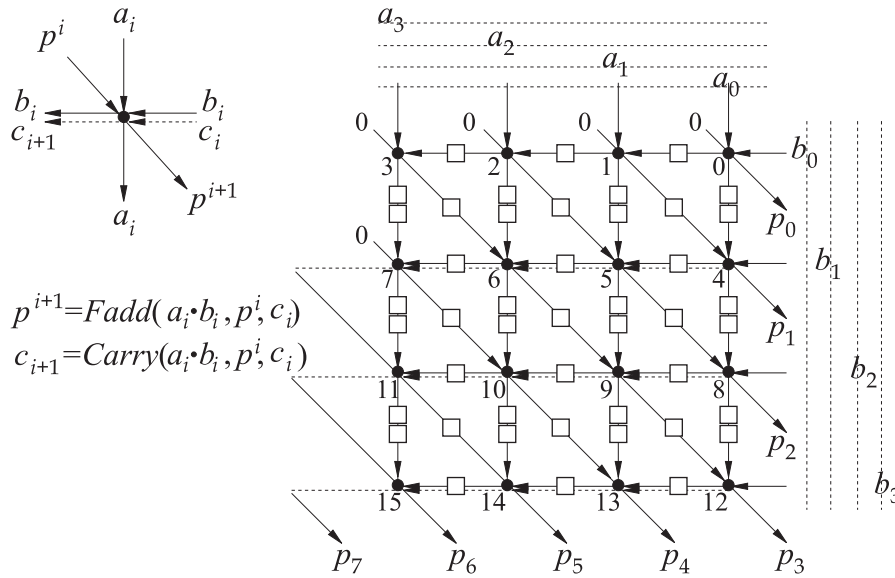
Slika 2.21: Algoritam za množenje brojeva

Polje  $3^6$  za množenje četvorobitnih binarnih brojeva prikazano je na slici 2.22. U polje sa slike 2.22 vrednosti  $a_i$  uvode se sa gornje strane, dok se  $b_i$  uvode sa desne strane. Tok bitova  $a_i$  je odozgo na dole. Bitovi  $b_i$  se kroz polje prenose s desna u levo. Čelije su na slici 2.22 predstavljene tačkama i zbog kasnijeg referenciranja označene su rednim brojevima od 0 do 15. Redni brojevi nalaze se dole levo u odnosu na ćeliju. Kašnjenja, radi jednostavnosti prikaza, predstavljena su kvadratima bez oznaka. Jednom kvadratu odgovara kašnjenje od jednog taktnog intervala. Jedna ćelija polja vrši množenje dva ulazna bita  $a_i$  i  $b_i$ , i akumulira dobijeni proizvod na prethodno određeni parcijalni proizvod ( $p^i$ ). Čelija uzima u obzir i prenos iz prethodne ćelije, tako da rezultat koji određuje ćelija predstavlja potpunu sumu  $p^{i+1} = a_i \cdot b_i + p^i + c_i$ . Pored sume  $p^{i+1}$  svaka ćelija generiše i narednoj ćeliji prosleđuje prenos dobijen sumiranjem ( $c_{i+1}$ ), slika 2.22. Ukoliko proizvod  $a_i \cdot b_i$  predstavimo sa  $ab_i$ , funkcije ćelije su

$$p^{i+1} = ab_i \oplus p^i \oplus c_i, \quad c_{i+1} = (ab_i \cdot p^i) + (ab_i \cdot c_i) + (p^i \cdot c_i).$$

Ukoliko bi iz polja sa slike 2.22 uklonila kašnjenja, rezultujuće polje predstavlja *wavefront* implementaciju algoritma za množenje brojeva, kod koga



Slika 2.22: DFG sistoličkog  $3^6$  polja za množenje brojeva

se talas (eng. *wave*) prostire paralelno glavnoj dijagonali polja odozgo na dole [1]. Dodati memorijski elementi omogućavaju uvođenje protočnosti u polje, koje se ogleda u tome da je naredno množenje moguće započeti pre nego se prethodno započeto množenje završi. Pored ove osobine, kritični put protočnog polja sadrži samo jednu ćeliju. Za razliku od *wavefront* implementacije, gde je kritični put od ulaza  $a_0$  i  $b_0$  do izlaza  $p_7$  (slika 2.22), taktni interval sistema u koji se ugrađuje protočni množač može biti značajno kraći.

U prvom taktnom intervalu ćelija 0 sistoličkog polja sa slike 2.22 određuje proizvod  $a_0 \cdot b_0$  i prosleđuje ga na izlaz  $p_0$ . Zakašnjen za jedan taktni interval,  $b_0$  se uvodi u ćeliju sa rednim brojem 1 u drugom taktnom intervalu, u vreme kada se na ulazu  $a_1$  u ćeliju 1 uvodi drugi po težini bit broja  $a$ . Parcijalni proizvod  $a_1 \cdot b_0$  se sa kašnjenjem od jednog taktnog intervala prosleđuje ćeliji sa rednim brojem 4, a bit  $b_0$ , takođe sa kašnjenjem od jednog taktnog intervala ćeliji 2. Parcijalni proizvod je dostupan ćeliji 4 u trećem taktnom intervalu. U ovom taktnom intervalu se u ćeliju uvodi i bit  $a_0$  koji je na putu sa dva kašnjenja ćeliji 4 prosleđen od strane ćelije 0 u prvom taktnom intervalu. Parcijalnom proizvodu  $a_1 \cdot b_0$  ćelija 4 dodaje parcijalni proizvod  $a_0 \cdot b_1$ , tako da se u trećem taktnom intervalu na izlazu  $p_1$  nalazi drugi bit rezultata  $p_1 = a_1 \cdot b_0 + a_0 \cdot b_1$ . Prenos nastao sumiranjem u ćeliji 4 se sa kašnjenjem od jednog taktnog intervala prosleđuje ćeliji 5, itd.

Bitovi rezultata na desnoj strani polja ( $p_0, p_1, p_2$  i  $p_3$  na slici 2.22) dostupni su na svaka dva taktna intervala (1, 3, 5, 7, respektivno). Bitovi rezultata sa

donje strane polja dobijaju se u svakom taktom intervalu po jedan, počev od taktog intervala nakon dobijanja bita rezultata  $p_3$ . Bitovi rezultata  $p_4$ ,  $p_5$ ,  $p_6$  i  $p_7$  sa slike 2.22 biće dostupni na odgovarajućim izlazima u taktim intervalima 8, 9, 10 i 11, respektivno. Period izračunavanja ovog polja jednak je  $3n$ , gde je  $n$  broj bitova brojeva  $a$  i  $b$ . Ovo se jednostavno može pokazati na osnovu prethodne analize. Naime, ukupan broj bitova rezultata  $p$  je  $2n$ . Polovina bitova dobija se na svaka dva taktna intervala, a polovina u svakom taktom intervalu ( $\frac{2n}{2} \cdot 2 + \frac{2n}{2} \cdot 1 = 3n$ ). Naredno izračunavanje se u protočni sistem može uvesti u narednom taktom intervalu u odnosu na vreme uvođenja  $a_0$  i  $b_0$ . Tako, nakon inicijalnog kašnjenja od  $3n$  taktih intervala, pun protočni sistem rezultate će davati u svakom taktom intervalu, po ceni implementacije dodatnih memorijskih elemenata.

U [43] autori su pokazali da je emulacija bilo koje topologije regularnog polja moguća na bilo kojoj drugoj topologiji. Na slici 2.23 prikazana je emulacija *ripple carry* algoritma za množenje brojeva implementiranog na  $3^6$  polju sa slike 2.22 na polju sa topologijom  $6^3$ . Efikasnost ove emulacije je 4, mereno u broju potrebnih čvorova za predstavljanje topologije [43]. Zauzeće resursa na čipu je u ovom slučaju nepromenjeno, jer je broj aritmetičkih operacija nakon transformacije grafa ostao isti. Naime, čvorovi T3 i T4 sa slike 2.23 ne vrše nikakva izračunavanja, već samo prosleđuju rezultate, dok čvor T1 vrši samo množenje, a čvor T2 sabiranje.

### Primer planarnog semi-sistoličkog polja

Regularna struktura algoritma za FIR filtriranje (poglavlje 2.1.3) pogodna je za implementaciju na sistoličkim poljima. Iskustva u implementaciji procesorskih polja u VLSI tehnologiji pokazala su da se veća propusnost sistema postiže kod semi-sistoličkih polja, jer je moguće ostvariti finiju protočnost u odnosu na implementacije sa potpuno-sistoličkim poljima [22, 23]. Neke od arhitekture koje se mogu svrstati u ovu grupu su: arhitektura sa paralelnim ulazom i serijskim izlazom, arhitektura bez akumulacije, bit-plane (BP) arhitektura i modifikovana bit-plane arhitektura. Zajednička karakteristika ovih arhitekture je da se mogu realizovati tako da imaju programabilne koeficijente [23].

U cilju ilustracije planarnih semi-sistoličkih polja, u ovom poglavlju prikazana je implementacija DSP algoritma za FIR filtriranje na ortogonalnom ( $4^4$ ) semi-sistoličkom bit-plane polju.

Neka je dat niz koeficijenata  $c_0, c_1, \dots, c_{k-1}$  i neka je dat niz ulaznih reči  $\{x_i\}$  koji se dovodi na ulaze FIR filtra. Tada se, u skladu sa izrazom (2.9), na izlazu FIR filtra dobija niz reči  $\{y_i\}$  određen sa

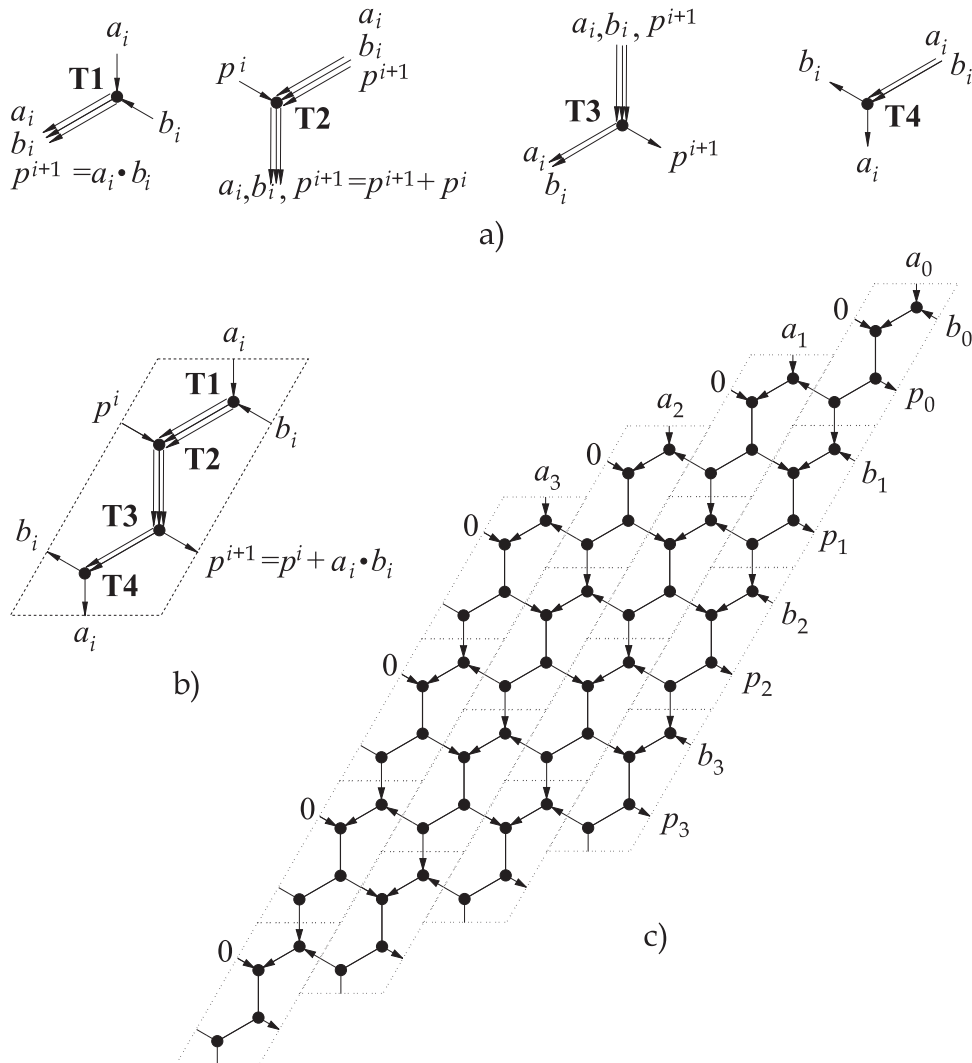
$$y_i = c_0 \cdot x_i + c_1 \cdot x_{i-1} + \dots + c_{k-1} \cdot x_{i-k+1}. \quad (2.13)$$

Blok dijagram FIR filtra realizovanog BP arhitekturom prikazan je na slici 2.24 ([22, 60]), gde je:

$m$  – dužina koeficijenta (broj bitova u koeficijentu);

$k$  – broj koeficijenata ( $c_0, c_1, \dots, c_{k-1}$ );

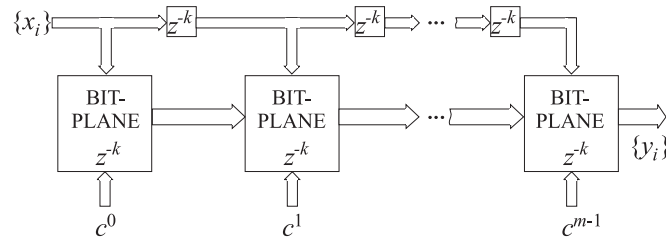
$c_j$  – vektor od  $k$  bitova težine  $2^j$  uzetih iz koeficijenata, odnosno ako važi  $c^i \equiv c_i^{m-1} c_i^{m-2} \dots c_i^0$ , gde su  $c_i^0, c_i^1, \dots, c_i^{m-1}$  bitovi koeficijenta  $c_i$  težina  $2^0, 2^1, \dots, 2^{m-1}$  redom, tada je  $c^j \equiv c_{k-1}^j, c_{k-2}^j, \dots, c_0^j$ , gde su  $c_0^j, c_1^j, \dots, c_{k-1}^j$



Slika 2.23: DFG wavefront  $6^3$  polja za množenje brojeva: a) tipovi ćelija; b) mreža za simulaciju jedne ćelije heksagonalnog polja; c) polje za množenje brojeva

bitovi težine  $2^j$  koeficijenata  $c_0, c_1, \dots, c_{k-1}$ , respektivno;

$z^{-k}$  – oznaka da je funkcionalnoj jedinici pridruženo kašnjenje od  $k$  taktnih intervala.



Slika 2.24: Blok dijagram bit-plane arhitekture

BP arhitektura je protočna, visokopropusna semi-sistolička arhitektura sa regularnim vezama između elemenata polja. Proizvodi  $c_j \cdot x_{i-j}$  se izračunavaju paralelno i istovremeno se jednom učitana reč  $x_i$  višestruko koristi.

Karakteristično za BP arhitekturu sa slike 2.24 je preuređenje parcijalnih proizvoda, tako da su množači koji realizuju proizvode  $c_j \cdot x_{i-j}$  zamenjeni BP elementima. U jednom BP elementu se dobija zbir parcijalnih proizvoda ulazne reči i bitova iste težine svih koeficijenata. Rezultat se prosleđuje na naredni BP gde se dodaje zbiru parcijalnih proizvoda naredne težine. Broj BP elemenata je jednak broju bitova u koeficijentu. Po ovoj osobini arhitektura je dobila naziv *bit-plane* (prev. ravan bitova).

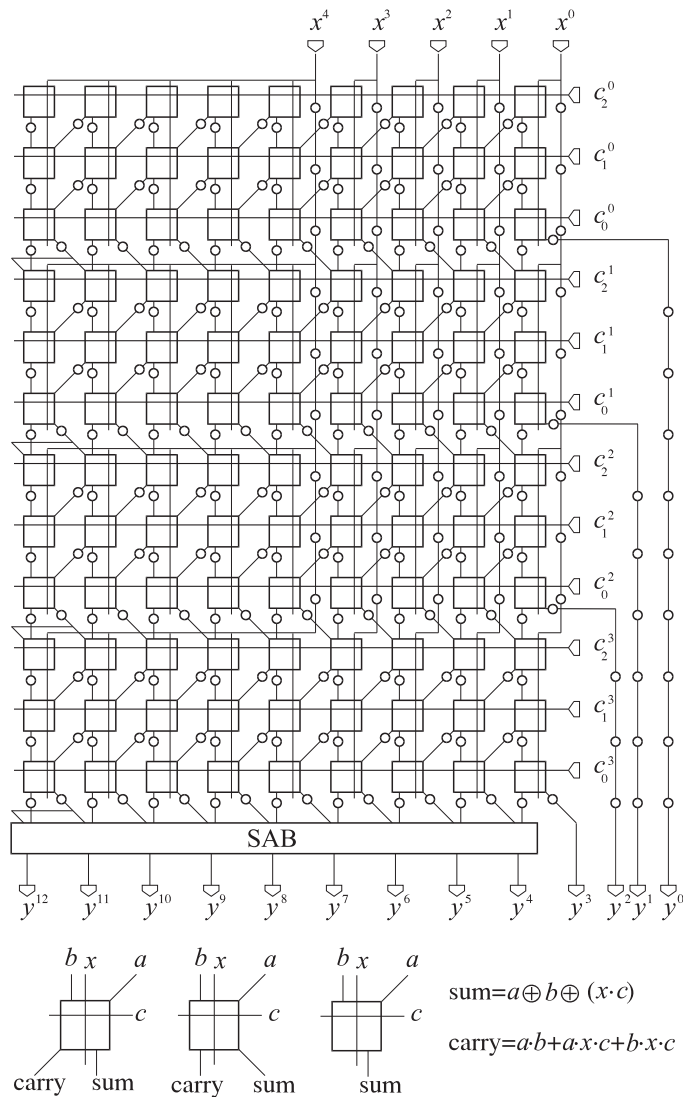
Funkcionalni blok dijagram BP FIR filtra sa 3 koeficijenta dužine 4 bita, i ulaznim rečima dužine 5 bitova prikazan je na slici 2.25. Kao što se sa slike 2.25 vidi, ova arhitektura je visokog stepena regularnosti i ima jednostavne veze između ćelija. Prenosna funkcija ovog filtra je

$$\begin{aligned}
 G(z) = & z^{-1}(c_0^3 2^3 z^{-9} + z^{-1}(c_1^3 2^3 z^{-9} + z^{-1}(c_2^3 2^3 z^{-9} + \\
 & + z^{-1}(c_0^2 2^2 z^{-6} + z^{-1}(c_1^2 2^2 z^{-6} + z^{-1}(c_2^2 2^2 z^{-6} + \\
 & + z^{-1}(c_0^1 2^1 z^{-3} + z^{-1}(c_1^1 2^1 z^{-3} + z^{-1}(c_2^1 2^1 z^{-3} + \\
 & + z^{-1}(c_0^0 2^0 + z^{-1}(c_1^0 2^0 + z^{-1}(c_2^0 2^0))))))))))
 \end{aligned}$$

U opštem slučaju, za  $k$  koeficijenta dužine  $m$  bitova i  $n$ -tobitne ulazne

reči prenosna funkcija BP FIR filtra je

$$\begin{aligned}
 G(z) &= z^{-1}(c_0^{m-1}2^{m-1}z^{-(m-1)k} + z^{-1}(c_1^{m-1}2^{m-1}z^{-(m-1)k} + \dots + \\
 &+ z^{-1}(c_{k-1}^{m-1}2^{m-1}z^{-(m-1)k} + \\
 &+ z^{-1}(c_0^{m-2}2^{m-2}z^{-(m-2)k} + z^{-1}(c_1^{m-2}2^{m-2}z^{-(m-2)k} + \dots + \\
 &+ z^{-1}(c_{k-1}^{m-2}2^{m-2}z^{-(m-2)k} + \\
 &\dots \\
 &+ z^{-1}(c_0^02^0 + z^{-1}(c_1^02^0 + z^{-1}(\dots + z^{-1}(c_{k-1}^02^0))\dots)).
 \end{aligned}$$



Slika 2.25: BP FIR filter se tri 4-bitna koeficijenta i 5-bitnim označenim ulaznim rečima ( $k = 3$ ,  $m = 4$ ,  $n = 5$ )

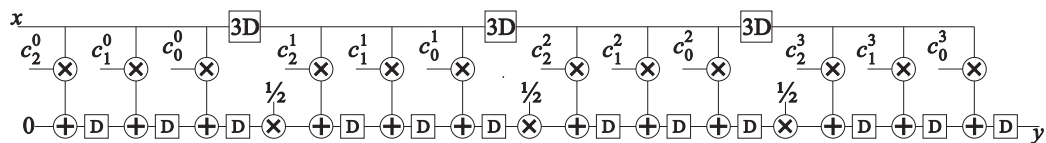
V. Ćirić, "Projektovanje parcijalno visokopouzdatih procesorskih polja", Doktorska disertacija

Inicijalno kašnjenje arhitekture je  $m \cdot k$  taktних intervala.

Funkcionalni blok dijagram BP FIR filtra sa slike 2.25 je projektovan za ulazne reči  $x_i$  predstavljene u dvojnomo komplementu, dok su koeficijenti neoznačeni brojevi. Kao što se sa slike vidi, ulazna reč je proširena bitom najveće težine (bitom znaka).

Svaka ćelija polja sa slike 2.25 ima 4 jednobitna ulaza ( $a, b, x, c$ ) i 2 jednobitna izlaza ( $s$  i  $c$ ). Bitovi ulazne reči  $x$  i koeficijenta  $c$  se u polje uvode emisionom linijom. Ulazni bitovi  $a$  i  $b$ , i izlazni bitovi  $s$  i  $c$  se kroz polje prenose protočno, tako da je polje sa slike 2.25 moguće klasifikovati kao semi-sistolicko polje, *bit-plane* po redosledu izračunavanja, sa topologijom  $4^4$ , prikazanom na slici 2.17. Za razliku od topologije sa slike 2.17, bit-plane arhitektura ima i drugu, dodatnu emisionu liniju (po jedna za  $x$  i  $c$ ) normalnu na emisionu liniju prikazanu isprekidanom linijom na slici 2.17 [22, 23].

Funkcije koje realizuje ćelija su suma ( $s = a \oplus b \oplus (x \cdot y)$ ) i prenos ( $c = a \cdot b + a \cdot x \cdot y + b \cdot x \cdot y$ ). Horizontalni niz ćelija na slici 2.25 čini jednu vrstu polja, a  $k$  vrsta formiraju jedan BP element. Svaka vrsta polja je zadužena za množenje ulaznog podatka jednim bitom koeficijenta, kao i za akumulaciju parcijalnih proizvoda. Prvi BP element obrađuje bitove najmanje težine, tako da je na njegovom izlazu moguće uzeti bit najmanje težine izlazne reči. Drugi BP element obrađuje bitove težine  $2^1$ , i na njegovom izlazu je određen izlazni bit težine  $2^1$ , itd. BP polje se sastoji od  $k \cdot m$  vrsta osnovnih ćelija, odnosno  $m$  BP elementa. SAB je brzi sabirač. Na slici 2.26 je DFG BP arhitekture za slučaj  $k = 3$  i  $m = 4$ .



Slika 2.26: Graf toka podataka BP arhitekture za  $k = 3$  i  $m = 4$

Jedan BP element se sastoji od  $k = 3$  množača i isto toliko sabirača i kola za kašnjenje. Svaki od množača predstavlja jednu vrstu osnovnih ćelija arhitekture sa slike 2.25. BP elementi su razdvojeni sa  $kD$  kašnjenja na putu ulaznih podataka i množačem na putu rezultata, koji množi među-rezultat sa  $\frac{1}{2}$ . Ovaj množač je realizovan pomeranjem među-rezultata za jedno mesto u desno (slika 2.25) [22].

Na slici 2.27 prikazan je tok podataka kroz arhitekturu BP FIR filtra. Sa  $V_0, V_1$  i  $V_2$  na slici 2.27 su označene prva, druga i treća vrsta osnovnih ćelija svakog BP elementa, respektivno. Na ulaz filtra se u prvom taktom intervalu dovodi ulazna reč  $x_0$  (slika 2.27). Prva vrsta ćelija prvog BP elementa

množi ovu ulaznu reč bitom težine  $2^0$  koeficijenta  $c_2$  (slika 2.27), tako da se u prvom taktom intervalu na izlaznim linijama prve vrste ćelija dobija parcijalni proizvod  $c_2^0 \cdot x_0$ . U narednom taktom intervalu na ulaz filtra se dovodi ulazna reč  $x_1$ . Tada druga vrsta ćelija prvog BP elementa određuje parcijalni proizvod  $c_1^0 \cdot x_1$  i ovom parcijalnom proizvodu dodaje među-rezultat koji je u prethodnom taktom intervalu odredila prva vrsta ćelija istog BP elementa. Dalje, u narednom taktom intervalu se na izlaznim linijama druge vrste ćelija nalazi suma parcijalnih proizvoda  $c_2^0 \cdot x_0 + c_1^0 \cdot x_1$ . Treća vrsta prvog BP elementa ovoj sumi dodaje parcijalni proizvod  $c_0^0 \cdot x_2$ , koji određuje u trećem taktom intervalu. Na izlaznim linijama iz treće vrste se u trećem taktom intervalu nalazi međurezultat  $c_2^0 \cdot x_0 + c_1^0 \cdot x_1 + c_0^0 \cdot x_2$ . Time je prvi BP završio izračunavanje međurezultata bitovima koeficijenata težine  $2^0$  ( $c^0$ ). Međurezultat  $c_2^0 \cdot x_0 + c_1^0 \cdot x_1 + c_0^0 \cdot x_2$  se tada množi sa  $\frac{1}{2}$ , odnosno pomera u desno za jedno mesto i u četvrtom taktom intervalu dovodi na ulaze prve vrste drugog BP elementa koji dodaje parcijalni proizvod  $c_2^1 \cdot x_0 + c_1^1 \cdot x_1 + c_0^1 \cdot x_2$ , itd. Kao što je i ranije rečeno, inicijalno kašnjenje BP arhitekture je  $m \cdot k$  taktom intervala. Za slučaj  $k = 3$  i  $m = 4$  inicijalno kašnjenje kroz BP FIR filter je  $m \cdot k = 12$  taktom intervala. Prva izlazna reč filtra,  $y_2 = c_2 \cdot x_0 + c_1 \cdot x_1 + c_0 \cdot x_2$ , dostupna je na izlaznim linijama arhitekture nakon dvanaestog taktom intervala (slika 2.27). BP FIR filter izlaznu reč  $y_2$  određuje na sledeći način:

$$\begin{aligned} y_2 &= (c_2^0 x_0 + c_1^0 x_1 + c_0^0 x_2) + (c_2^1 x_0 + c_1^1 x_1 + c_0^1 x_2) + \\ &+ (c_2^2 x_0 + c_1^2 x_1 + c_0^2 x_2) + (c_2^3 x_0 + c_1^3 x_1 + c_0^3 x_2) = \\ &= c_2 \cdot x_0 + c_1 \cdot x_1 + c_0 \cdot x_2. \end{aligned}$$

Paralelno sa izračunavanjem izlazne reči  $y_2$  (protočno), počev od drugog taktom intervala, počinje izračunavanje izlazne reči  $y_3$  (slika 2.27). U drugom taktom intervalu prva vrsta ćelija prvog BP elementa računa parcijalni proizvod  $c_2^0 \cdot x_1$ . U trećem taktom intervalu druga vrsta prvog BP elementa na ovaj parcijalni proizvod dodaje parcijalni proizvod  $c_1^0 \cdot x_2$ , itd. Na kraju trinaestog taktom intervala na izlaznim linijama je dostupan rezultat  $y_3 = c_2 \cdot x_1 + c_1 \cdot x_2 + c_0 \cdot x_3$  (slika 2.27).

Broj BP elemenata jednak je broju bitova u koeficijentu. Evidentno je da na drugu dimenziju polja, širinu, pored broja bitova u koeficijentu utiče i sam broj koeficijenata. U [61] je pokazano da se jedna vrsta BP polja sastoji od

$$l_0 = \left\lfloor \frac{(m+2 + \log_2 k)(k-1)}{k} \right\rfloor + n - 1, \quad (2.14)$$

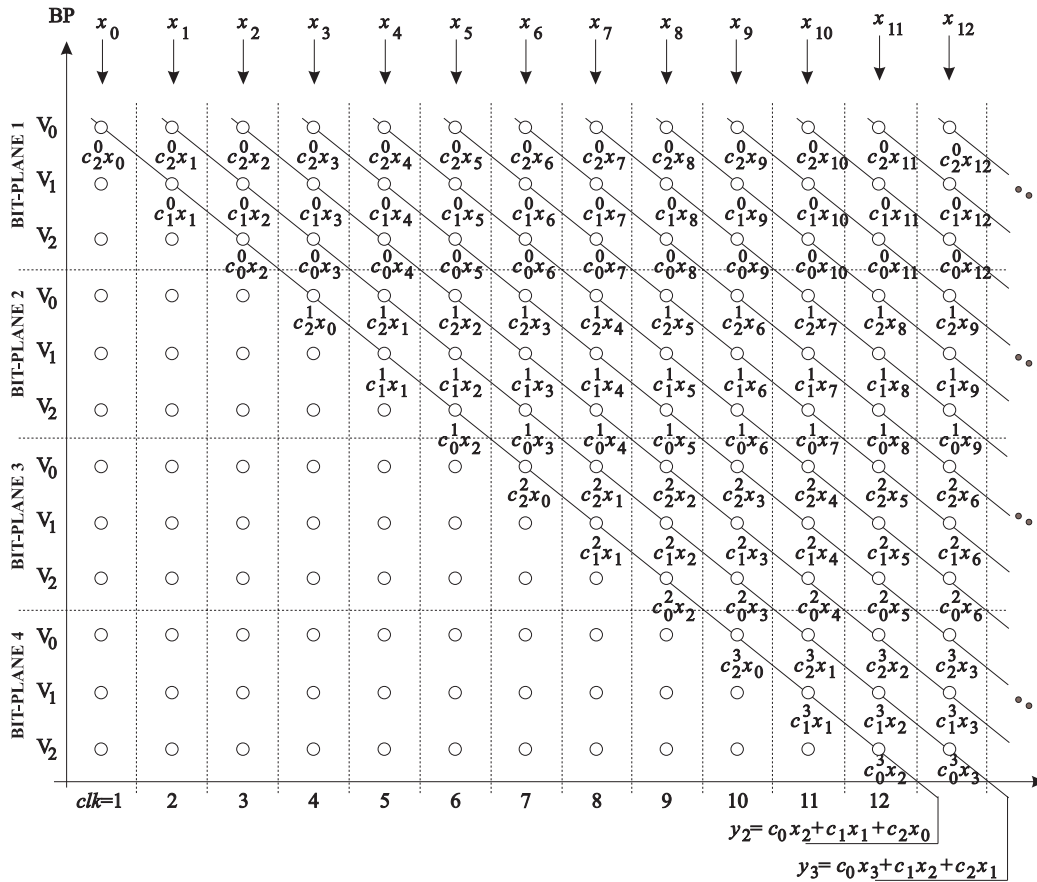
ćelija, uzimajući u obzir i ćelije koje su dodate u cilju proširenje ulaznog podatka u prvim BP elementima. Broj ćelija u polju je

$$T = l_0 \cdot m \cdot k. \quad (2.15)$$

Ukupan broj flip-flopora je  $N_{FF} = N_{FF1} + N_{FF2}$ , gde je  $N_{FF1}$  broj flip-flopora koji služe za postizanje protočnosti, za bitove sume  $s$  i prenosa  $c$ , a  $N_{FF2}$  broj flip-flopora koji služe za ulazne reči  $x_i$ , odnosno

$$N_{FF} = 2 \cdot \left( \left\lfloor \frac{m + 2 + \lceil \log_2 k \rceil (k - 1)}{k} \right\rfloor + n - 1 \right) k \cdot m - (k - 1) \cdot m + (m - 1) k \cdot n.$$

Regularna struktura sistoličkih i semi-sistoličkih polja, kako je ranije pomenuto, pogodna je za implementaciju u VLSI tehnologiji. U cilju provere teoretskih rezultata ovog rada, arhitekture obrađene u radu implementirane su na sistemu za izradu prototipova arhitekture. Naredno poglavlje posvećeno je ovom okruženju.



Slika 2.27: Tok podataka kroz BP FIR filter ( $k = 3$ ,  $m = 4$ ,  $n = 5$ )



## 2.4 Implementacija prototipa DSP sistema

Integrirana kola specifične namene (eng. *Application Specific Integrated Circuits - ASIC*) su integrirana kola projektovana namenski za određenu aplikaciju. Proces projektovanja ASIC kola uključuje nivo apstrakcije počev od aplikativnog, pa do nivoa rasporeda tranzistora na čipu. Fabrikacija ovakvih čipova je skupa, jer zahteva pokretanje celog proizvodnog postupka [62]. Zbog prethodno navedenog ova tehnologija je isplativa samo kod velikih serija.

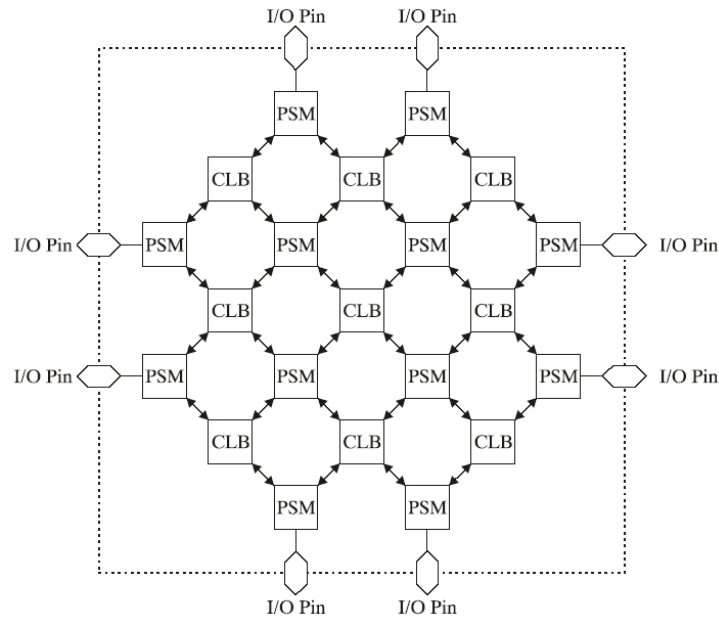
Kako je ASIC skupa i vremenski zahtevna tehnologija za slučaj malih serija i brzu izradu prototipa koriste se programabilna logička kola FPGA (eng. *Field-Programmable Gate Arrays*) [32]. FPGA su čipovi kod kojih je moguće programirati funkcije gradivnih elemenata, kao i veze između njih. Ove čipove karakteriše niska cena projektovanja i znatno brže vreme potrebno za projektovanje. FPGA čipovi su veoma pogodni za digitalnu obradu signala. Rade na većim brzinama od DSP čipova opšte namene, i po kvalitetu odgovaraju DSP čipovima specijalne namene, a uz sve to imaju i nižu cenu. Pored toga, FPGA omogućava da se projekat menja nakon što je proizveden. FPGA je potrebno programirati kako bi se postigla željena funkcionalnost.

Postoji više proizvođača i mnogo različitih familija FPGA kola. Svaku familiju karakteriše izgled osnovne gradivne ćelije, kao i organizacija programabilnih veza. Pored funkcionalne provere ispravnosti dizajna, FPGA implementacija može, sa određenim odstupanjem, kroz zauzeće resursa na FPGA čipu nagovestiti zauzeće površine silicijuma u slučaju ASIC dizajna. Alati za FPGA sintezu ovaj podatak nakon mapiranja dizajna na resurse čipa daju u vidu "ekvivalentnog broja gejtova" (eng. *equivalent gate count*). Kako se pojam "resursa" od familije do familije razlikuje, za proveru rezultata i poređenja u ovom radu biće korišćen ekvivalentni broj gejtova.

U daljem tekstu opisana je arhitektura FPGA čipova, sa posebnim osvrtom na familiju koja je korišćena za proveru i ilustraciju rezultata ovog rada.

### 2.4.1 Arhitektura FPGA kola

Opšta struktura FPGA je prikazana na slici 2.28. Strukturu FPGA čini dvodimenzionalno polje logičkih blokova CLB (Configurable Logic Block) povezanih programabilnim međuvezama PSM (Programmable Switch Matrices) [63]. Na periferiji čipa se nalaze ulazno/izlazni blokovi IOB (Input/Output Blocks). PSM povezuje izlaze i ulaze susednih CLB-ova. FPGA imaju mnogo više CLB-a od I/O pinova, tako da svaki CLB ne može da ima kontakt sa spoljašnjim svetom.



Slika 2.28: Uopštena struktura FPGA čipa

### 2.4.2 Spartan familija FPGA kola

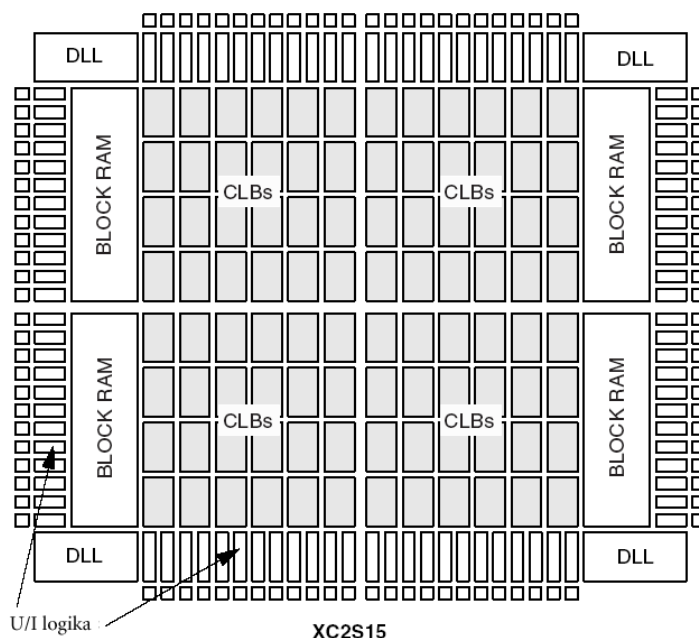
Za implementaciju projektovanih arhitektura u ovom radu korišćeni su FPGA čipovi firme Xilinx. Xilinx nudi dve familije čipova, Spartan i Virtex, gde svaku familiju čini veći broj arhitektura. Čipovi u okviru familije razlikuju se kako po tehnologiji u kojoj su implementirani, tako i po dodacima u vidu memorije, množača, specijalizovanih kola i sl. [63].

Spartan FPGA su čipovi sa visokim performansama i velikim kapacitetom. Ovu familiju FPGA čipova Xilinx razvija počev od 1998. godine, i do danas je čine Spartan, Spartan-II, Spartan-XL i Spartan-3. Xilinx je u vreme pisanja ovog rada najavio i predstavio novu seriju Spartan-6 FPGA čipova. Spartan-6 je serija izrađena u 45-nm tehnologiji sa devetoslojnim dizajnom čipa i par stotina hiljada CLB-ova na čipu, zavisno od modela. Xilinx FPGA čipovi podržani su softverom koji pokriva sve faze projektovanja (Xilinx-ISE), od unosa šeme kola ili opisa ponašanja sistema, preko planiranja razmeštaja, logičke i funkcionalne simulacije, automatskog razmeštaja blokova i trasiranja veza, do učitavanja konfiguracionog niza bitova na čip, unošenje podataka na čip i čitanje sa njega [63].

Arhitekture projektovane u ovom radu implementirane su na Spartan-II familiji FPGA čipova. Gustina pakovanja Spartan-II kreće se od 15,000 gejtova za najmanji čip, do 200,000 sistemskih gejtova za čip najveće gustine.

## Arhitektura Spartan-II FPGA čipova

Čipovi iz Spartan-II familije imaju regularnu, fleksibilnu, programabilnu strukturu konfigurabilnih logičkih blokova (CLB), okruženih programabilnim ulazno-izlaznim blokovima (IOB). Na svakom uglu čipa (slika 2.29) se nalazi po jedna petlja kašnjenja takta (Delay-Locked Loop - DLL). Dve kolone blok RAM-a se nalaze na različitim stranama čipa, između CLB-ova i IOB kolona. Ovi funkcionalni elementi su povezani hijerarhijski organizovanim kanalima za rutiranje (slika slika 2.29) [63].



Slika 2.29: Blok dijagram osnovnih gradivnih blokova Spartan-II čipova

Konfiguracija čipova iz Spartan-II familije vrši se tako što se konfiguracioni podaci učitaju u ćelije interne statičke memorije. Ovakvim pristupom omogućen je neograničen broj reprogramiranja čipova. Konfiguracioni podaci, smešteni u ove ćelije, određuju logičke funkcije i veze koje su implementirane na FPGA. Konfiguracioni podaci mogu biti čitani i direktno sa nekog spoljašnjeg medijuma sa serijskim pristupom (npr. PROM).

## Funkcionalne jedinice Spartan-II FPGA

Spartan-II familija FPGA čipova je implementirana kao regularna, fleksibilna i programabilna arhitektura sačinjena od polja CLB-ova, okruženih

mnoštvom programabilnih U/I blokova. U arhitekturu Spartan II FPGA čipa su uključeni i dodatni resursi kao što su blok RAM i blokovi za upravljanje taktom [63].

Polje Spartan-II čipova (slika slika 2.29) se sastoji od pet osnovnih elemenata:

1. CLB (Configurable Logic Block) – konfigurabilni logički blok,
2. blok RAM – interna memorija FPGA čipa,
3. IOB – ulazno/izlazni interfejs između pinova i unutrašnje logike kola,
4. DLL (Delay Locked Loop) – petlje kašnjenja za distribucija taktnog signala,
5. mreža za rutiranje i veze.

Sa slike 2.29 vidi se da CLB-ovi imaju pristup svim dodacima i strukturi za rutiranje. U cilju lakšeg i bržeg prenosa signala u čip kao i lakšeg i bržeg prenosa signala iz čipa, IOB su locirani oko CLB-ova i memorijskih elemenata.

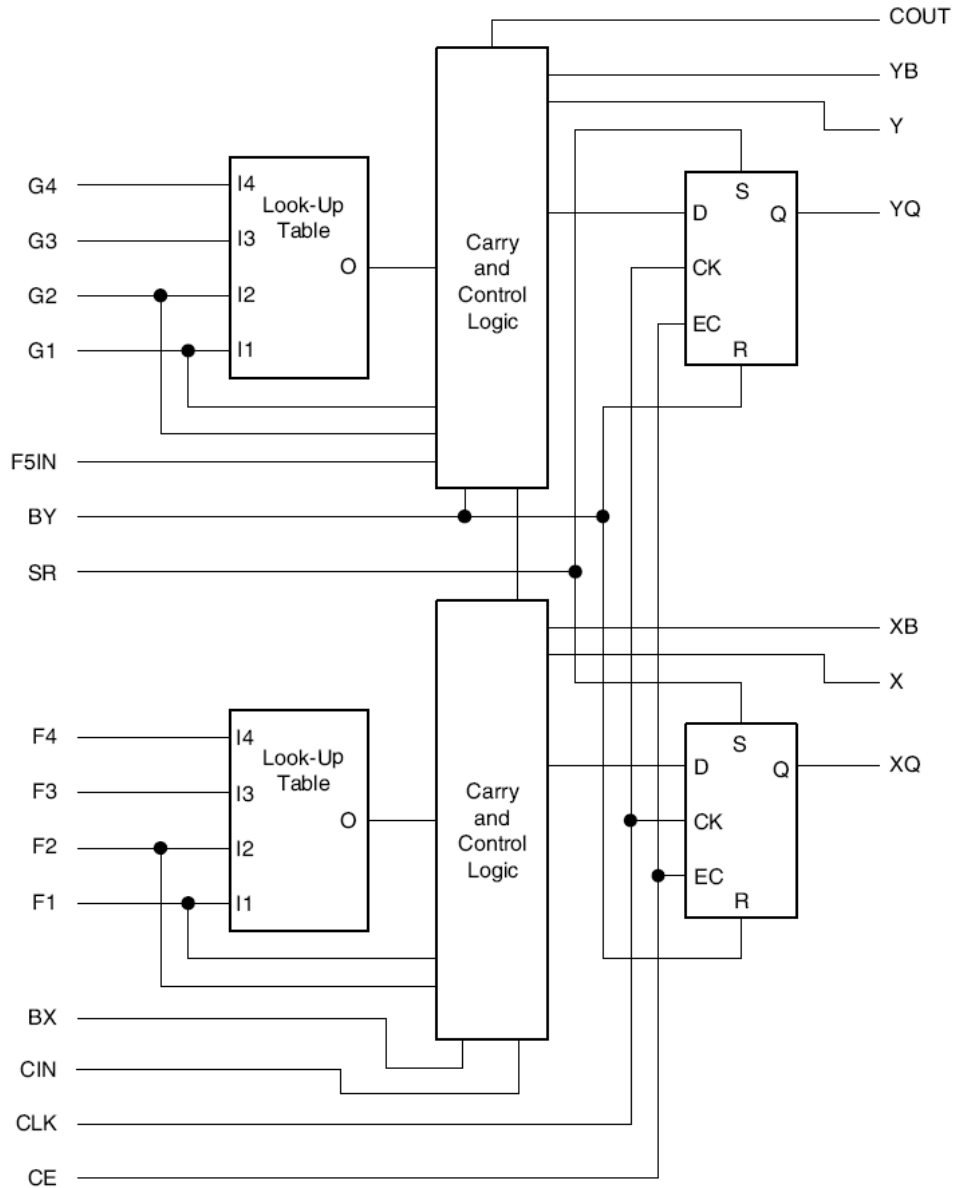
Osnovni gradivni blok Spartan-II FPGA kola logička ćelija (LC). LC uključuje četvoroulaznu *Look-Up* tabelu (LUT), *carry* logiku i memorijske elemente. Svaki CLB Spartan-II čipa se sastoji od četiri logičke ćelije, smeštene u dve identične celine (eng. Slice - SL). Na slici 2.30 je prikazana blok šema SL Spartan-II FPGA čipa. Dodatak logičkim ćelijama je mreža koja kombinuje generatore funkcija čime je moguće implementirati logičke funkcije sa više ulaza.

*Carry* logika (slika 2.30) omogućava implementaciju brzih aritmetičkih funkcija. CLB Spartan-II čipova podržava dva nezavisna lanca prenosa (eng. *carry chain*). Dužina lanca prenosa je dva bita po CLB-u. Podrška za aritmetičke operacije podrazumeva XOR kolo koje se koristi za implementaciju jednobitnog potpunog sabirača po logičkoj ćeliji. Dodatno AND kolo povećava efikasnost pri implementaciji množača. Lanac prenosa, u obliku kaskadnog lanca generatora funkcija, može biti korišćen za implementaciju logičkih funkcija sa velikim brojem ulaza.

Koncept blok RAM memorije dopunjuje relativno mali kapacitet distribuirane memorije, koja je implementirana u LU tabelama CLB elemenata. Blokovi RAM memorije su organizovani kao kolone. Svaki čip iz Spartan-II familije sadrži po dve kolone blok RAM-a (jedna uz svaku vertikalnu ivicu čipa). Kolone se protežu celom dužinom čipa. Visina jednog memorijskog bloka jednaka je zbiru visina četiri CLB-a. Tako, Spartan-II čip sa osam CLB

vrsta, ima implementirana dva bloka uz svaku vertikalnu ivicu, tj. ukupno četiri memorijska bloka. U tabeli 2.2 prikazana je količina raspoložive memorije za pojedine čipove iz Spartan-II familije.

Frekvencu taktnog signala, tj. maksimalnu brzinu rada implementiranog kola, definiše najduži put između dva memorijska elementa u kolu. Proces za



Slika 2.30: Blok šema SL Spartan-II čipa

Spartan-II čip	Broj Blokova	Blok RAM [kB]
XC2S15	4	16
XC2S30	6	24
XC2S50	8	32
XC2S100	10	40
XC2S150	12	48
XC2S200	14	56

Tabela 2.2: Kapacitet raspoloživog blok RAM-a za Spartan-II

optimizaciju rutiranja, ugrađen u softver za projektovanje kola, ima za cilj da minimizira kašnjenje najdužeg puta signala u kolu. Mreža za rutiranje signala u okviru Spartan II FPGA čipa implementirana je u dva nivoa. Rutiranje signala između fizički bliskih CLB-ova se obavlja preko lokalne mreže za rutiranje, dok se rutiranje između udaljenih CLB-ova obavlja preko globalne mreže za rutiranje.

Resursi za lokalno rutiranje, prikazani na slici 2.30, dozvoljavaju tri tipa veza. Prvi tip veza su veze između LU tabela, flip-flopora i matrice za rutiranje signala (eng. General Routing Matrix - GRM). Drugi tip veza su povratne, brze veze u okviru CLB-a sa minimalnim kašnjenjem pri rutiranju. Treći tip su direktne veze između horizontalno bliskih CLB-ova. Ove veze imaju ulogu da eliminišu kašnjenja GRM-a.

Većina signala u okviru Spartan-II čipa se rutira preko globalne mreže za rutiranje signala. Drugim rečima, većina resursa za rutiranje signala pripada ovom nivou hijerarhije. Resursi globalne mreže za rutiranje su implementirani kao horizontalni i vertikalni kanali, vezani za vrste, odnosno kolone CLB-ova. Resursi globalne mreže za rutiranje su: globalna matrica za rutiranje, 24 linije za rutiranje GRM signala do obližnjih GRM-a u sva četiri pravca, 96 linija sa baferima za prenos signala na rastojanje koje nije veće od 6 CLB-ova, 12 dugih linija sa beferima.

Programsko okruženje koje je korišćeno za opis i simulaciju u ovom radu je Xilinx ISE WebPack 8.2i. Za implemetaciju polja korišćen je Spartan II FPGA čip sa oznakom XC2S200.

### 2.4.3 Rezultati implementacije sistoličkih polja u FPGA tehnologiji

U ovom poglavlju dati su rezultati imlementacije jednosmernog i dvosmernog 1D polja za množenje matrice i vektora (slike 2.18 i 2.19), planarnog  $3^6$  polja za množenje brojeva sa slike 2.22 i semi-sistoličkog  $4^4$  bit-plane polja za FIR filtriranje (slika 2.25). Rezultati implementacije dati u ovom poglavlju

biće korišćeni u narednim poglavljima za proveru teoretskih rezultata koji se odnose na kompromis između površine i stepena parcijalne pouzdanosti.

### Implementacija 1D sistoličkih polja za množenje matrice i vektora

U tabeli 2.3 dati su rezultati implementacije jednosmernog polja za množenje matrice, prikazanog na slici 2.18, gde je sa  $N$  označen broj ćelija polja, a sa  $n_x$  broj bitova kojim su kodirane vrednosti ulaznog niza  $\{x_i\}$ . Elementi izlaznog niza  $\{y_i\}$  implementirani su sa  $n_y = 2n_x$  bitova. Sabirači u ćelijama polja sa slike 2.18 imaju  $n_y$ -to bitne ulazne magistrale, dok su ulazne magistrale u množač širine  $n_x$ , a izlazna  $n_y$  bitova. Polje je opisano u VHDL-u, a mapiranje komponenata i rutiranje veza izvršeno je automatski u programskom okruženju Xilinx ISE WebPack 8.2i. Sistoličko polje implementirano je na Spartan II FPGA čipu sa oznakom XC2S200.

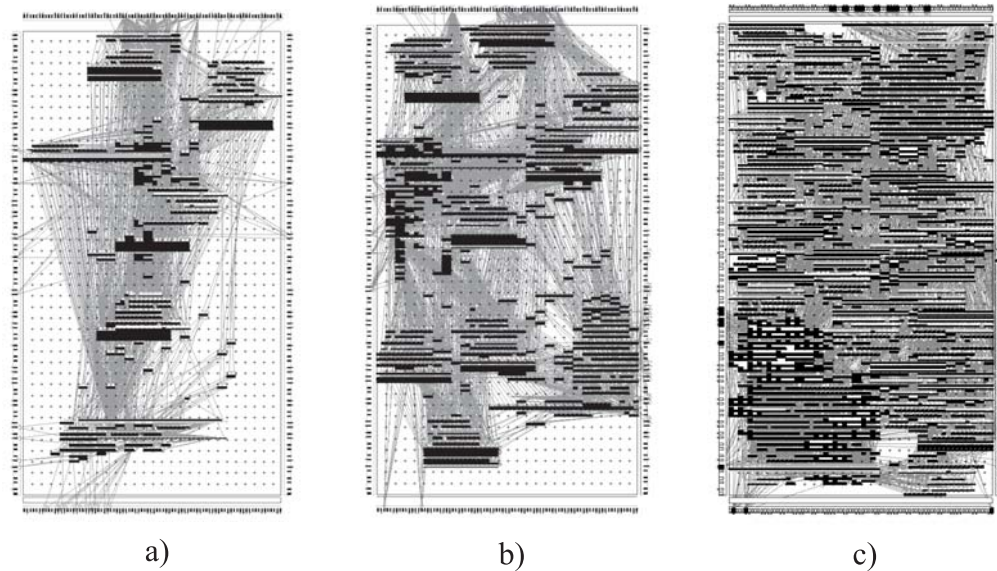
$N$	$n_x = 8, n_y = 16$		$n_x = 16, n_y = 32$	
	resursi [kG]	takt [ns]	resursi [kG]	takt [ns]
4	9,5	15,0	24,8	24,3
8	17,8	16,7	47,8	24,0
16	36,6	20,0	/	/

Tabela 2.3: Rezultati FPGA implementacije jednosmernog polja za množenje matrice i vektora na Spartan II XC2S200 FPGA čipu

Na slici 2.31 prikazan je raspored zauzetih resursa na XC2S200 FPGA čipu za slučaj implementacije jednosmernog polja za množenje matrice i vektora dimenzija datih u tabeli 2.3. Na slici 2.31, generisanoj iz okruženja, zauzeti resursi prikazani su crnom bojom, dok su veze između CLB-ova u pozadini prikazane sivom bojom. Rutiranje veza na slici nije prikazano. Veze su prikazane kao prave linije od izvorne do određene tačke, ili do spoljnog pina čipa.

Na slici 2.31c prikazano je zauzeće čipa kada se na njemu implementira polje dimenzija  $N = 16$  i  $n_x = 8$ , od 36.6 kG (tabela 2.4). Relativno u odnosu na veličinu izabranog FPGA čipa, ovo iznosi 73% resursa, odnosno 1739 od ukupno 2352 *slice*-a koliko ima XC2S200 FPGA čip (slika 2.31c).

Iz rezultata datih u tabeli 2.3 može se uočiti linearni rast zauzeća resursa sa povećanjem broja ćelija polja. Ovo je očekivano, s obzirom na topologiju polja datu na slici 2.18. Mala odstupanja od savršene proporcionalnosti nastala su zbog optimizacije prilikom implementacije funkcija na CLB-ovima FPGA čipa.



Slika 2.31: Izgled SpartanII XCS200 čipa i raspored zauzetih resursa na čipu za slučaj implementacije jednosmernog polja za množenje matrice i vektora dimenzija: a)  $N = 4, n_x = 8$ , b)  $N = 8, n_x = 8$ , c)  $N = 16, n_x = 8$ .

Polje sa slike 2.18 zahteva veliki broj pinova za uvođenje elemenata matrice i za paralelno čitanje rezultata iz svake ćelije. Kako FPGA čip ima ograničen broj spoljnih pinova, implementacija je izvršena tako da je polju pridodat serijsko-paralelni registar za uvođenje elemenata matrice u polje, i paralelno-serijski registar za čitanje rezultata iz polja. Ovo je takođe malo narušilo proporcionalnost rezultata datih u tabeli 2.3.

Rezultati implementacije dvosmernog jednodimenzionalnog polja za množenje matrice i vektora, prikazanog na slici 2.19, dati su u tabeli 2.4.

$N$	$n = 8$		$n = 16$	
	resursi [kG]	takt [ns]	resursi [kG]	takt [ns]
3	5,2	16,3	14,7	26,5
7	10,7	14,6	33,2	25,5
15	23,5	15,2	71,0 <sup>(*)</sup>	/

<sup>(\*)</sup> implementacija zahteva više resursa nego što čip poseduje

Tabela 2.4: Rezultati FPGA implementacije dvosmernog polja za množenje matrice i vektora na Spartan II XC2S200 FPGA čipu



Kao i kod jednosmernog polja, i ovom polju je u cilju smanjenja broja potrebnih pinova dodat serijsko-paralelni registar za uvođenje elemenata matrice. Za drugim registrom u ovom slučaju nema potrebe, jer je polje projektovano tako da rezultati nisu stacionarni u ćelijama, već se prenose odvojenim tokom ka izlazu (slika 2.20).

U tabeli 2.4 vidi se skoro linearna promena zauzeća resursa čipa sa promenom broja ćelija polja. Mala odstupanja nastala su, kao i kod prethodno opisanog polja, zbog optimizacije prilikom mapiranja funkcija kola na CLB-ove FPGA čipa i uvedenog registra. Iz istog razloga, sa promenom broja ćelija dolazi i do male razlike u dužini kritičnih puteva, što se može videti u koloni sa taktnim intervalom u tabeli 2.4. Teoretski, taktni interval ne zavisi od broja ćelija polja.

### Implementacija 2D polja za množenje brojeva

Planarno polje za množenje brojeva, prikazano na slici 2.22, opisano je u VHDL-u, tako da je u obzir uzeta samo kombinaciona mreža. Implementacija je izvršena na SpartanII FPGA čipu. Rezultati implementacije, bez uzimanja u razmatranje memorijskih elemenata, dati su u tabeli 2.5. Dimenzija polja označena je sa  $N$ . Potrebno je napomenuti da polje  $N = 4$  ima ukupno  $T = N^2 = 16$  ćelija.

$N$	ukupan br. ćelija	resursi [kG]	propagacija [ns]
4	16	0,35	12,98
8	64	1,36	21,80
16	256	5,20	39,44
32	1024	20,74 <sup>(1)</sup>	/
64	4096	83,75 <sup>(2)</sup>	/

<sup>(1)</sup> implementacija zahteva više I/O portova nego što ima na čipu

<sup>(2)</sup> implementacija zahteva više resursa nego što čip poseduje

Tabela 2.5: Rezultati FPGA implementacije  $3^6$  polja za množenje brojeva na Spartan II XC2S200 FPGA čipu

Kako je implementacija polja izvršena bez lečeva, iz tabele se može videti da kritični put zavisi od dimenzije polja, što je u skladu sa definicijom 2.8. Kritični put kod ovog polja jednak je vremenu propagacije signala od ćelije označene sa 0 na slici 2.22, do ćelije koja generiše najviši bit rezultata. U primeru na slici 2.22 ova ćelija ima oznaku 15.

Prilikom implementacije ovog polja nisu korišćeni dodatni registri za uvođenje podataka u polje, pa je već za  $N = 32$ , zbog nedostatka ulazno/iz-

laznih pinova, polje nemoguće implementirati na XC2S200. Za ovo polje u tabeli je dat jedino ekvivalentni broj gejtova, a kašnjenje je izostavljeno zbog nemogućnosti mapiranja dizajna na ovaj čip.

### Implementacija semi-sistoličkog bit-plane FIR filtra

Semi-sistoljičko polje bit-plane FIR filtra sa slike 2.25 opisano je u VHDL-u i implementirano na XC2S200 FPGA čipu. Rezultati implementacije polja različitih dimenzija dati su u tabeli 2.6.

$k$	$m$	$n$	$l_0$	$m \cdot k$	broj ćelija	zauzeće [kG]	takt [ns]
3	4	5	9	12	108	3,4	9,2
4	8	8	16	32	512	19,4	9,1
4	8	16	24	32	768	28,6	9,7
4	8	24	32	32	1024	37,8	10,2
4	16	8	22	64	1408	50,2	9,8
4	32	8	34	128	4352	148,3*	/

(\*) implementacija zahteva više resursa nego što čip poseduje

Tabela 2.6: Rezultati implementacije semi-sistoličkog BP FIR filtra

Polje je opisano tako da su dimenzije polja zadate parametrima (*generic* konstantama u VHDL-u):  $k$  - broj koeficijenata,  $m$  - broj bitova koeficijenta,  $n$  - broj bitova ulaznog niza,  $l_0$  - broj kolona polja. Parametrima se definišu  $k$ ,  $m$ , i  $n$ , a  $l_0$  se određuje na osnovu jednačine 2.14.

U prvoj vrsti tabele, ispod zaglavlja, dati su rezultati implementacije polja sa dimenzijama kao na slici 2.25. Potrebno je napomenuti da sabirač sa slike 2.25 utiče sa oko 10% resursa u odnosu na ukupna zauzeća data u tabeli.

U narednom poglavlju date su definicije pojmova i kratak pregled tehnika za postizanje visoke pouzdanosti sistema, koje su korišćene kasnije u ovom radu.



## Glava 3

# Pouzdani sistemi i tolerancija na greške

Sistemi koji se koriste u hazardnim uslovima, kao na primer u nuklearnim reaktorima, avionima i svemirskim stanicama su, pre svega zbog uticaja zračenja, u velikoj meri podložni greškama. Održavanje i popravka takvih sistema je vrlo skupa i zahteva dosta vremena. Zbog toga, pored performansi, otpornost na defekte, koja predstavlja mogućnost sistema da korektno funkcioniše uprkos pojavi defekata, postaje vrlo važno pitanje [8]. S druge strane, mere veličine pojedinih komponenti izražavaju se u broju atoma, odnosno molekula koje zauzimaju. Sa međuatomskim rastojanjem od oko  $0.5nm$ , tranzistor u 65-nanometarskoj tehnologiji širok je nešto više od 100 atoma. Ključni elementi, kao što je gejst tranzistora su efektivno trećina pomenute veličine. Ovakav razvoj tehnologije i drastično smanjenje komponenti integrisanih kola dovešće do toga da se u narednih par dekada veličina integrisanih komponenti izražava jednocifrenim brojem atoma [11]. Najznačajniji problemi u procesu fabrikacije ovakvih čipova su varijacije u proizvodnji koje dovode do razlika u istoj seriji proizvoda, stepen defekata koji se javlja, škart, kao i osetljivost samog procesa na interne i eksterne smetnje [4, 5]. Ovakav trend će po svemu sudeći smanjiti efektivnu ekonomsku isplativost budućih tehnologija.

Visoka pouzdanost je osobina koja omogućava sistemu da nastavi da funkcioniše ispravno u slučaju otkaza neke od njegovih komponenti. Ako kvalitet funkcionisanja sistema opada, on je proporcionalan broju defekata, za razliku od sistema koji ne realizuju visoku pouzdanost kod kojih i najmanji otkaz može da dovede do kompletnog otkaza sistema [8, 64, 65].

Visoka pouzdanost je trenutno najtraženija u sistemima od kojih zavisi zdravlje i životi ljudi, sistemima za vojne primene i sistemima koji rade u specijalnim uslovima [7, 13, 66]. Međutim, ukoliko se ostvare predviđanja

da će došadašnja mikroelektronska VLSI kola zameniti nanoelektronska kola, zbog velikog stepena otkaza koji se predviđa za ovu tehnologiju, visoka pouzdanost će postati imperativ pri projektovanju arhitektura sistema [67]. Veliki stepen otkaza nanoelektronskih komponenti je cena koja se plaća za ekstremno velike gustine pakovanja, a koja se mora smanjiti metodama za postizanje visoke pouzdanosti, kako bi proizvodnja ovakvih čipova bila isplativa [68]. Zbog značaja koji tehnika projektovanja predložena u ovom radu može imati kod projektovanja arhitektura za nano-implementaciju, na početku ovog poglavlja biće dat kratak osvrt na razvoj nanotehnologije.

Cena uvođenja visoke pouzdanosti je povećanje kompleksnosti sistema. Postoji veliki broj metoda za projektovanje sistema sa visokom pouzdanošću. Sve metode se mogu klasifikovati u tri grupe: hardverske, softverske i informacione metode. U drugom delu ovog poglavlja biće prikazane postojeće metode za povećanje pouzdanosti sistema. U trećem delu poglavlja biće data kvantitativna analiza dobiti u procesu proizvodnje koja se može postići uvođenjem visoke pouzdanosti. Na kraju poglavlja biće razmotren alternativni pravac u projektovanju arhitektura u vidu tolerancije aplikacije na postojanje grešaka.

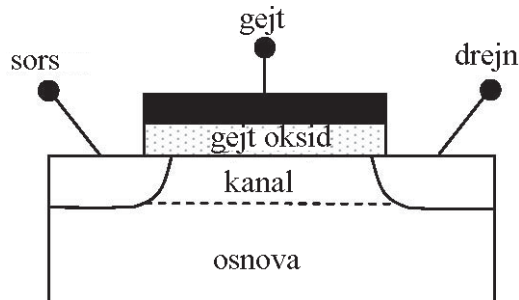
## 3.1 Nanotehnologija

Gordon Moore, koosnivač Intela, je 1965. godine predvideo da će se broj tranzistora koji se mogu integrisati na čip duplirati svake dve godine [69]. Proizvođači čipova su održavali eksponencijalni rast broja tranzistora na čipu par desetina godina, međutim, ovakav trend neće biti prisutan još dugo. Murov zakon ne može trajati večno [67]. Tri glavna razloga koja ne dozvoljavaju dalji rast broja tranzistora su: rast cene proizvodnje, ograničenja litografije, i ograničenja vezana za veličinu tranzistora. Na primer, tranzistori su već smanjeni na veličinu od par desetina atoma, a pri veličini tranzistora od 1-2 atoma prelazak na novu tehnologiju biće neizbežan [67].

### 3.1.1 Ograničenja litografije

Za proteklih nekoliko decenija osnovna gradivna jedinica VLSI integrisanih kola bio je MOSFET tranzistor (eng. *Metal Oxide Semiconductor Field Effect Transistor*). MOSFET je četvoroterminalna komponenta sa priključima sors, drejn, gejt i osnovom, koja se kod digitalnih integrisanih kola u najčešće koristi kao prekidač. Struktura MOSFET-a prikazana je na slici 3.1. Sors i drejn su dva kraja prekidača, čiji se kanal "otvara" polarizacijom gejta. Gejt je odvojen od kanala tankim slojem oksida [70]. Komplementarni MOS

(CMOS) se najčešće upotrebljava kod logičkih kola, i sastoji se od po jednog pMOS i nMOS tranzistora [70].



Slika 3.1: Struktura MOSFET tranzistora

Trenutne projekcije razvoja integrisanih kola predviđaju kraj MOSFET tehnologije do 2022. godine, sa  $11nm$  procesom proizvodnje. Čak i sa  $22nm$  procesom postoje veliki i za sada nerešeni problemi. Među ovim problemima su povećanje potrošnje izazvano strujama curenja, manja tolerancija na varijacije procesa proizvodnje, i veliki rast cene proizvodnje [5].

Kroz idealni tranzistor struja teče samo kada je gejt tranzistora polarizovan na određeni način. Kada je idealni tranzistor, uslovno rečeno, "isključen", kroz tranzistor ne teku struje. Nažalost, tranzistori nisu idealni. Može se čak reći da što je tranzistor manji, manje je idealan [67]. Sa smanjenjem veličine tranzistora struje curenja postaju dominantne. Kod  $22nm$  tehnologije ove struje u ukupnoj potrošnji čipa utiču sa preko 50% [67]. Struje curenja nastaju kao posledica prelaska elektrona od gejta do kanala kroz gejt oksid (slika 3.1). Smanjenjem tranzistora smanjuje se i debljina gejt oksida, što vodi povećanju struje curenja kroz gejt oksid.

Druga komponenta struje curenja je struja između sorsa i drejna, koja se javlja u odsustvu polarizacije gejta. Ova struja postaje veća sa smanjenjem kanala i smanjenjem napona praga provođenja [70]. Dinamička potrošnja čipa se smanjuje sa smanjenjem komponenti, jer je manje energije potrebno prevesti tranzistor iz stanja u stanje. Međutim, zbog povećanja statičke potrošnje, ukupna potrošnja se povećava. Na primer, Intelov procesor u  $250nm$  tehnologiji, Pentium, u piku zahteva oko 15W, dok Intelov Itanium 2 u  $90nm$  tehnologiji ima potrošnju od oko 177W [67]. Sa povećanjem gustine pakovanja, lokalizovano zagrevanje predstavlja veliki problem, koji može da prouzrokuje otkaz čipa.

Pored povećanja potrošnje problem je i povećana osetljivost na varijacije u proizvodnji prilikom smanjenja komponenti. Na primer, debljina gejt oksida

je trenutno oko pet atoma. Ukoliko se debljina razlikuje za 1 atom, varijacija u proizvodnji je 20%.

Veliki problem za trenutnu MOSFET tehnologiju predstavlja i eksponencijalni rast cene proizvodnih postrojenja sa smanjem veličine tranzistora. Rast cene proizvodnih postrojenja je direktno u vezi sa povećanjem zahteva za preciznošću prilikom izrade čipova [67].

### 3.1.2 VLSI nanoelektronika

Fizičko ograničenje pri daljem smanjenju komponenti, kao i nesrazmeran rast cene proizvodnje verovatno će dovesti do fundamentalnog zaokreta u načinu proizvodnje čipova [67]. Mnogi istraživači veruju da će rešenje problema doneti nanotehnologija. Individualne komponente u nanotehnologiji su velike svega par nanometara, i moguće ih je proizvesti po relativno niskoj ceni. Stepem integracije koji je moguće postići u nanotehnologiji je  $10^{12}$  komponenti po  $cm^2$ , dok *International Technology Roadmap for Semiconductors* [5] predviđa kraj MOSFET tehnologije 2018. godine sa stepenom integracije od  $10^{10}$  komponenti po  $cm^2$ .

Osnovne gradivne komponente u nanoelektronici, koja će zameniti bakarne provodnike i poluprovodničke komponente biće grafitne nanocevi (eng. *Carbon NanoTube - CNT*) i silikonski nanoprovodnici (eng. *Silicon NanoWires - SNW*) [67].

CNT je cilindrična struktura izgrađena od atoma ugljenika, otkrivena 1991. godine kao sporedni efekat u proizvodnji sferičnih molekula  $C_{60}$  [71]. Struktura CNT je prikazana an slici 3.2a. Tipične dimenzije CNT su od  $0.7nm$  do  $2nm$  za prečnik cevi, i od  $10nm$  do  $20nm$  za dužinu [67].

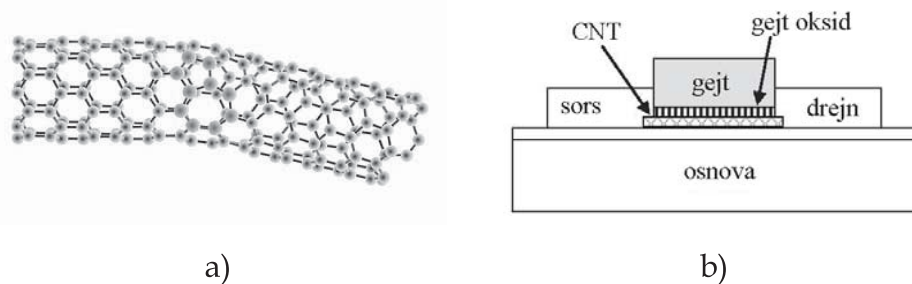
Jedna varijanta FET tranzistora sa CNT prikazana je na slici 3.2b. Na slici 3.2b prikazana je tzv. druga generacija CNT tranzistora, kod koje se gejst polaže preko CNT. Struja kroz CNT se kontroliše polarizacijom gejta. Zbog same strukture i karakteristika CNT, kod ove tehnologije ne postoje prethodno navedeni problemi koji postoje kod MOSFET tranzistora i litografije [67]. Moguća primena CNT prikazana je na slici 3.3. Na slici 3.3 prikazana je memorija sa četiri ćelije izgrađene u nanotehnologiji [67].

Silikonski nanoprovodnicu su takođe mogući pravac u razvoju nanotehnologije. Za razliku od CNT, pored uloge provodnika, SNW mogu imati i ulogu aktivnih komponenti [67].

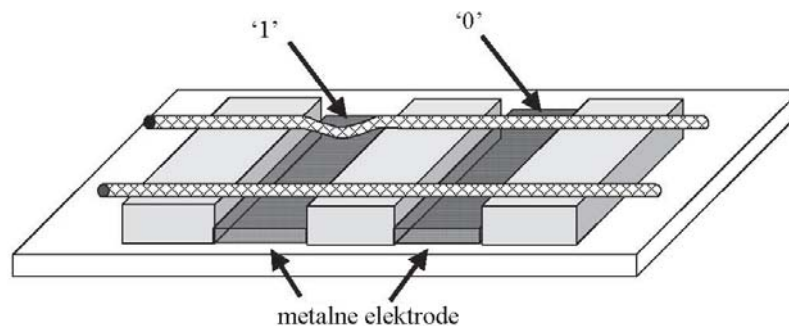
Litografski postupak, kojim se komponente kod trenutno dostupne VLSI tehnologije raspoređuju na čip, dozvoljava skoro bilo kakav raspored komponenata na čipu. Nanotehnologija će, po svemu sudeći, dozvoljavati postavljanje isključivo regularnih struktura na čip, sa strogo definisanim pravilima ponavljanja. Umesto litografije, za raspoređivanje komponenti u nanote-

hnologiji će se verovatno koristiti stohastički samoasemblirajući procesi, ili drugim rečima, delimično kontrolisani rast komponenti baziranih na CNT ili SNW [67]. Za razliku od determinističkog samoasembliranja, kod stohastičkog samoasembliranja će čipovi biti asemblirani metodama tako da se na čipu gradi regularna struktura komponenata sa malo, ili bez spoljne intervencije [67].

Trenutno ne postoji proizvodna linija koja proizvodi čipove u nanotehnologiji. Nanotehnologija je u eksperimentalnoj fazi, a prototipovi sa malim brojem komponenata se razvijaju u laboratorijama. Do sada je predloženo nekoliko metoda za postavljanje nanokomponenti na površinu čipa [67]. Ilustracije radi, nanokomponente se mogu rasporediti na površini čipa Langmur - Blodžetovom metodom. Ova metoda razvijena je 1900. godine za potrebe postavljanja sloja debljine jednog molekula na filmsku traku [72]. Metoda je za slučaj CNT, odnosno CNW ilustrovana na slici 3.4. Nanokomponente se postavljaju u tečnost koja se nalazi na podlozi na koju je potrebno



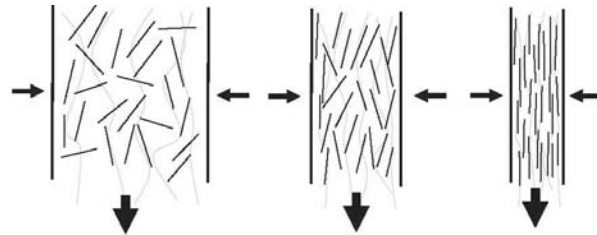
Slika 3.2: a) struktura grafitne nanocevi, b) struktura tranzistora druge generacije u CNT tehnologiji



Slika 3.3: Struktura memorije u nanotehnologiji



postaviti komponente (slika 3.4). Oko komponenti se hemjskim reakcijama u tečnosti stvara sloj oksida, dok se tečnost istiskuje, kako bi se komponente poravnale. Debljinom oksida kontroliše se rastojanje između komponenti. Kada se postavi jedan sloj komponenti na podlogu, na isti način je moguće postaviti još jedan sloj ponavljanjem postupka pod pravim uglom [72]. Na ovaj način je moguće kreirati rešetku komponenti na podlozi. Ovom metodom je veoma precizno moguće kontrolisati ugao pod kojim se postavljaju nanocevi i nanoprovodnici, ali je teško poravnati krajeve nanokomponenti. Ovaj nedostatak preciznosti će imati dosta uticaja na proces kada se čipovi budu proizvodili u ovoj tehnici. Ovom tehnikom je do sada postignuto poravnanje sa greškom od oko  $90nm$ , ali se veruje da će ovom tehnikom biti moguće postići 3 – 4 puta manju grešku [67].



Slika 3.4: Langmur-Blodžetova tehnika za postavljanje paralelnih nanocevi na čip

Trenutno dostupna VLSI elektronska kola bazirana na liteografiji koriste takav model upravljanja defektima da retko kada dozvoljavaju i jedan defekt na čipu. Ovaj model nije primenljiv na kola u nanotehnologiji. Trenutne procene su takve da će čipovi u nanotehnologiji imati između 1% i 15% defektnih komponenti. U takvom okruženju neophodno je metode za postizanje visoke pouzdanosti ugraditi u arhitekturu kako bi se povećao broj upotrebljivih čipova i omogućila isplativost fabrikacije. Zbog postojećih problema, bar prvih nekoliko generacija nano-čipova će biti hibridi mikro- i nano-tehnologije [67].

### 3.1.3 Otpornost na defekte u nanotehnologiji

U trenutno aktuelnoj CMOS tehnologiji, postupkom litografije moguće je proizvesti čip sa velikim stepenom pouzdanosti, odnosno sa veoma malom verovatnoćom pojave defekta. Malo je verovatno da će biti moguće postići takvu preciznost i pouzdanost u nanotehnologiji [67]. Preciznost od par atoma istraživači postižu u laboratorijskim uslovima, ali je malo verovatno

da će to biti moguće praktično postići u procesu proizvodnje u kom se na čip postavlja  $10^{12}$  komponenti [67].

Najznačajnije karakteristike nanotehnologije su:

1. samoasembliranje čipa,
2. manipulacija komponentama na visokom nivou (navođenje i postavljanje velike grupe komponenti npr. Langmur-Blodžetovom tehnikom),
3. primena kompletno slučajnih procesa (npr. postavljanje dovoljnog broja elemenata dok ih statistički ne bude dovoljno za zadovoljavajuće funkcionisanje čipa).

Ovakav pristup se značajno razlikuje od mikrotehnologije, kod koje je litografskim postupkom precizno moguće rasporediti komponente na željene lokacije.

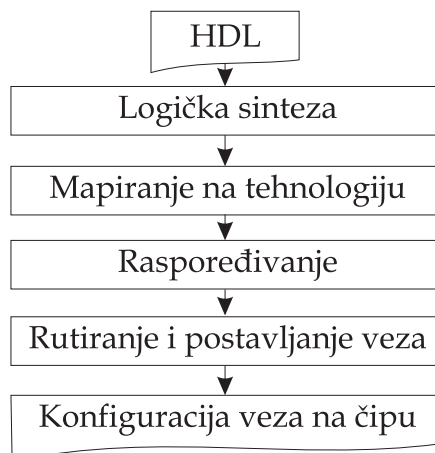
Iako je nanoelektronika na početku razvoja, jasno je da će nivo defekata biti značajno veći u poređenju sa CMOS mikrotehnologijom. Tačan broj defektnih komponenti je teško predvideti, ali se pretpostavlja da će taj broj biti između 1% i 15% resursa, što je i do 100.000.000.000 defekata na  $10^{12}$  komponenti. Za razliku od CMOS mikroelektronskih kola, kod kojih se nivo defekata od  $10^{-9}$  smatra *state-of-art* (do 10 defekata na  $10^{10}$  komponenti), kod nanotehnologije posebna pažnja mora se posvetiti problemu pouzdanosti [67].

Veliki broj defekata u nanotehnologiji je direktna posledica ekstremno male veličine komponenti u nanotehnologiji. Bez obzira na atraktivnost veličine komponenti u nanotehnologiji, pouzdanost predstavlja veliki problem iz tri razloga: nepredvidivost rasporeda, nepostojanost zbog malog broja atoma, manja tolerancija na izobličenja signala. Pouzdanost se mora razmatrati na arhitekturnom nivou, umesto na nivou komponenti [67].

Tranzicija na nanoelektroniku će zahtevati i modifikacije trenutnih CAD (eng. *Computer-Aided Design*) alata, kako bi se odgovorilo zahtevima tehnologije. Na slici 3.5 prikazani su koraci pri projektovanju FPGA kola. Kao dodatak prikazanom toku, pre raspoređivanja resursa na čip kod nanotehnologije mora se dodati korak kojim će se upravljati eventualnom pojavom defekata [67].

Autori u [68] predviđaju razvoj tehnika kojima bi se određivale "mape ranjivosti" arhitekture (eng. *vulnerability maps*), kako bi se detektovali regioni od najvećeg interesa za ispravno funkcionisanje čipa. Autori ovaj hipotetički postupak nazivaju "bojenje rasporeda", kojim se ukazuje na "vruće zone" (eng. *color-coded plan indicating hotspot regions*) [68].

Cilj ovog rada je određivanje "vrućih zona" arhitekture na sistematski način, korišćenjem matematičkih metoda. Da bi formalno definisali metod za



Slika 3.5: Koraci pri CAD projektovanju FPGA kola

određivanje interesnih zona kod regularnih arhitektura, u narednom poglavlju biće data definicija osnovnih termina.

## 3.2 Defekt, greška, otkaz

Kod visokopouzdanih sistema razlikujemo tri osnovna termina: defekt, greška i otkaz.

**Definicija 3.1 (Defekt)** *Defekt (eng. defect) je fizička neispravnost, nepotpunost, ili oštećenje koje se javlja unutar neke hardverske ili softverske komponente. Defekt je takođe mana, slabost ili zastarelost određene hardverske ili softverske komponente.*

Primeri defekata uključuju kratke spojeve između električnih provodnika, otvorene ili prekinute veze, fizička oštećenja ili nedostatke u poluprovodničkim komponentama, i drugo [73]. Primer defekta u programu je pogrešan kod koji uzrokuje neželjeno ponašanje softvera.

**Definicija 3.2 (Greška)** *Greška (eng. error) je posledica defekta ili nekog štetnog spoljnjeg uticaja. Ukoliko se greška propagira od mesta nastanka do izlaza sistema, kažemo da greška ima globalni karakter. U sprotnom, greška je lokalna.*

Konkretno, greška je odstupanje od tačnosti ili ispravnosti. Na primer, pretpostavimo da na vezi postoji spoj koji rezultuje time da je linija uvek na

nivou logičke 1. Ako se pojavi stanje koje zahteva da se linija prebaci na nivo logičke 0, vrednost na liniji će biti pogrešna. Drugim rečima, ispravna vrednost na liniji bi bila logička 0, ali postojanje defekta je rezultovalo pogrešnom vrednošću na liniji. S druge strane, ako je stanje kola takvo da na konkretnoj liniji treba da bude nivo logičke 1, postojanje defekta ne uzrokuje greku. Dakle, čak i ako postoji defekt, greška ne mora da se ispoljava u svim situacijama [73].

Konačno, ako greška rezultira time da sistem neku svoju funkciju izvršava pogrešno, došlo je do otkaza (eng. *failure*) sistema.

**Definicija 3.3 (Otkaz)** *Otkaz je neizvršavanje funkcije sistema usled pojave grešaka.*

Otkaz takođe predstavlja i izvršavanje neke funkcije u neodgovarajućem kvalitetu ili kvantitetu [73]. Drugim rečima, ukoliko se govori o uzročno - posledičnim vezama, defekt može (a i ne mora) izazvati grešku, koja može dovesti do otkaza sistema (slika 3.6).

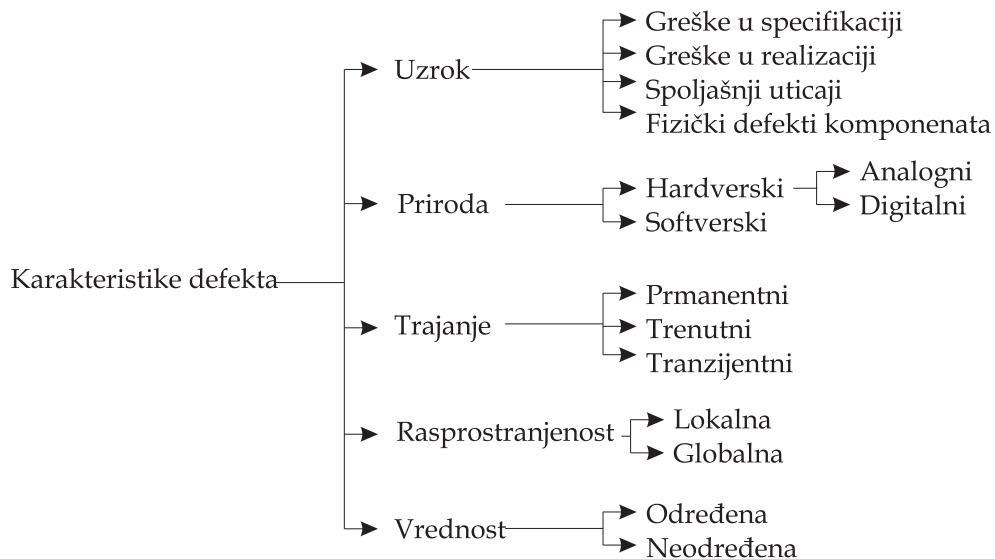


Slika 3.6: Uzročno - posledične veze defekta, greške i otkaza sistema

Defekti se mogu javiti zbog uticaja raznovrsnih pojava kako unutar samih elektronskih komponenata, tako i u njihovom okruženju, ili mogu biti inicirani još u fazi projektovanja ili proizvodnje komponente ili sistema. Takođe, defekti se razlikuju po trajanju, rasprostranjenosti i manifestaciji. Klasifikacija defekta se može obaviti prema različitim kriterijuma, na način koji je prikazan na slici 3.7 [12, 73, 74].

Rezultati izloženi u ovom radu odnose se na digitalne defekte, permanentne po trajanju, čiji je uzrok fizički defekt komponenti na čipu.

Stalni ili permanentni defekt (eng. *permanent fault*) traje neograničeno dugo i manifestuje se sve dok se ne preduzmu odgovarajuće mere korekcije. Permanentni defekti posledica su ireverzibilnih fizičkih promena. Poboljšanja u procesu projektovanja kao i tehnike izrade poluprovodnika u značajnoj meri smanjuju stopu pojave permanentnih defekata. S druge strane, sve veći stepen integracije dovodi do češćih pojava grešaka u toku proizvodnje [4]. Za komponentu ili sistem definiše se stopa otkaza (eng. *failure rate*) koja predstavlja broj otkaza koji se javljaju po jedinici površine [7].



Slika 3.7: Karakteristike defekta

### 3.3 Tehnike za povećanje pouzdanosti sistema

Sa aspekta projektovanja, za poboljšavanje ili zadržavanje nominalnih performansi, odnosno za postizanje visoke pouzdanosti u radu sistema, u okruženju u kome su defekti mogući, koriste se tri osnovna pristupa: izbegavanje defekata, maskiranje defekata, i otpornost na defekte [6, 7, 73].

**Definicija 3.4 (Izbegavanje defekta)** *Izbegavanje defekta (eng. fault avoidance) je skup tehnika koje imaju za cilj sprečavanje pojave hardverskih defekata ili softverskih grešaka.*

Ovakve tehnike se pre svega baziraju na korišćenju boljih materijala i poboljšanju tehnoloških procesa i metoda projektovanja kola. Defekte je moguće izbeći procesima kao što su preispitivanje projekta, zaštita komponenti, testiranje, i drugim metodama kontrole kvaliteta. U skladu sa slikom 3.6, cilj tehnika za izbegavanja defekata je prekinuti uzročno - posledične veze na samom začetku.

**Definicija 3.5 (Maskiranje defekta)** *Maskiranje defekta (eng. fault masking) je skup tehnika čiji je cilj sprečavanje defekata da uzrokuju greške (slika 3.6).*

Primeri maskiranja defekata su: memorije sa mogućnošću korekcije grešaka, ili većinsko izglasavanje u sistemima gde postoje više istovetnih modula, kod

kojih se na osnovu većine mogu maskirati neispravni moduli [11]. Sistemi koji koriste maskiranje defekata ostvaruju osobinu otpornosti "sakrivanjem" defekata koji se javljaju. Cilj je da uticaj svih defekata bude lokalnog karaktera. Tehnike zadržavanja defekta, koje se zasnivaju na maskiranju, sprečavaju širenje njihovih efekata kroz sistem.

**Definicija 3.6 (Otpornost na otkaze)** *Otpornost na otkaze (eng. Fault Tolerant - FT) je sposobnost sistema da nastavi sa izvršavanjem svojih zadataka i posle pojave defekata, odnosno osobina sistema da sve greške u sistemu imaju isključivo lokalni karakter.*

Za sistem se kaže da je otporan na otkaze ukoliko može da nastavi korektno izvršavanje specificiranih zadataka i u prisustvu hardverskog defekta. Osnovni cilj otpornosti na otkaze je sprečavanje otkaza sistema usled pojave defekta [75]. Otpornost na otkaze može se postići mnogim tehnikama. Jedna od njih se sastoji u tome da se defekt detektuje i locira, a da se zatim, sistem rekonfiguriše kako bi se eliminisao uticaj neispravne komponente ili modula. Uopšteno govoreći, rekonfiguracija je proces eliminisanja iz sistema entiteta sa defektom i vraćanja sistema u operativno stanje. Rekonfiguracija obuhvata aktivnosti detekcije, lokalizacije, izolacije i sanacije defekta [12].

Otpornost sistema na otkaze, koja je do skoro bila korišćenja kao pristup samo kod profesionalnih uređaja koji su bili namenjeni za rad u hazardnim uslovima i kritičnim misijama, postala je neminovnost koja se uvodi u komercijalne proizvode [6]. VLSI tehnologija je omogućila dovoljnu kompaktnost kola, tako da se mogu primeniti redundantne tehnike. Redundansa predstavlja dodavanje informacija, resursa, ili vremena pored onoga koje je potrebno za normalan rad sistema. Redundansa može imati više oblika, a to su: hardverska, informaciona, vremenska i softverska redundansa.

### 3.3.1 Hardverska redundansa

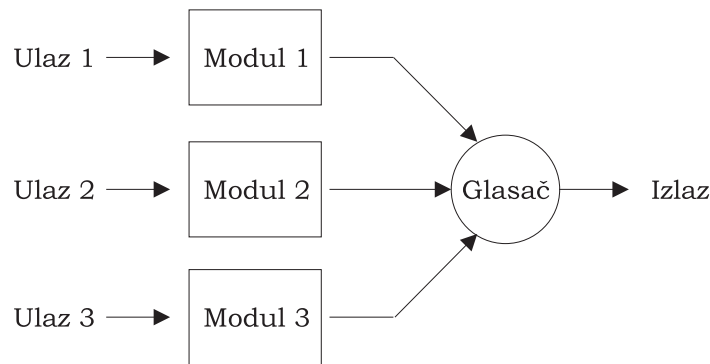
Hardverska redundansa predstavlja dodavanje hardvera, obično u cilju detekcije ili otklanjanja defekata. U svom osnovnom obliku ovaj metod obuhvata dupliciranje funkcije i poređenje rezultata dve kopije specificiranog kola pomoću kola za proveru. Prednost ovakvog postupka je u tome što se može primeniti na bilo koju funkciju. Fizičko umnožavanje hardvera je najčešći oblik redundanse koji se danas koristi u cilju postizanja otpornosti na defekte. Kako poluprovodničke komponente postaju sve manje i jeftinije hardverska redundansa postaje prihvatljivija [8].

Postoje tri osnovna oblika hardverske redundanse: pasivna, aktivna i hibridna. Pasivne tehnike koriste princip maskiranja defekta kako bi sakrile

pojavu defekta i sprečile da defekt prouzrokuje pojavu greške. Ovakvim tehnikama postiže se otpornost na otkaze bez zahteva za određenim akcijama na delovima sistema, odnosno bez potrebe za detekcijom grešaka ili rekonfiguracijom sistema. Pasivna hardverska redundansa oslanja se na mehanizam glasanja kako bi maskirala pojavu defekata. Većina pasivnih tehnika zasnovana je na principu većinskog glasanja.

### N-modularna redundansa

Osnovni princip trostruke modularne redundanse (eng. *Triple Modular Redundancy - TMR*), kao što se sa slike 3.8 vidi, je utrostručavanje hardvera i primena većinskog glasanja pri određivanju izlaza sistema. U slučaju jednog defektnog modula, dva preostala ispravna modula pri većinskom glasanju maskiraju rezultate neispravnog modula [8, 9, 12].

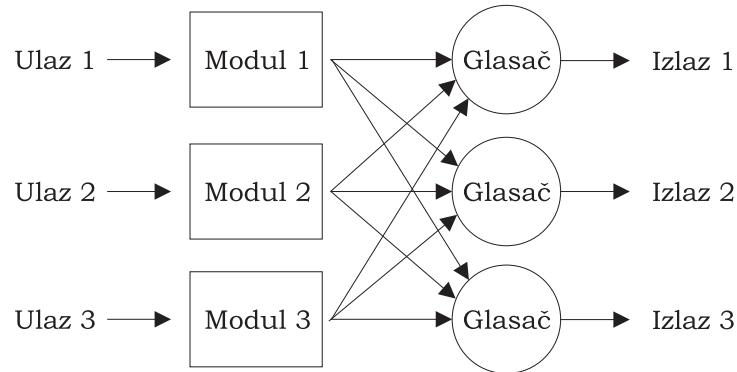


Slika 3.8: Model pasivne hardverske redundanse sa tri istovetna modula i jednim glasačem

Problem sa TMR-om je jedinstvena tačka otkaza koju uvodi glasač. Naime, ako glasač otkaze, ceo sistem će otkazati. Drugim rečima, pouzdanost najjednostavnijeg oblika TMR-a nije bolja od pouzdanosti glasača. Bilo koja pojedinačna komponenta unutar sistema čiji otkaz dovodi do otkaza sistema zove se jedinstvena tačka otkaza, ili singularitet.

Nekoliko tehnika se može upotrebiti za prevazilaženje efekta otkaza glasača. Jedna od njih, prikazana na slici 3.9, je utrostručavanje samog glasača i obezbeđivanje tri nezavisna izlaza (pretpostavka je da se tri izlaza vode na tri ulaza podsistema koji takođe koristi TMR).

Neka je  $p$ , ( $0 \leq p \leq 1$ ) verovatnoća otkaza modula, gde  $p = 0$  znači da je modul sigurno ispravan, a  $p = 1$  da je modul sigurno neispravan.



Slika 3.9: Model za eliminaciju jedinstvene tačke otkaza korišćenjem tri nezavisna glasača

**Lema 3.1 (Pouzdanost TMR sistema)** *Sistem nad kojim je primenjena osnovna TMR tehnika (slika 3.8) pouzdaniji je od sistema koji sadrži samo jednu kopiju modula kada je verovatnoća otkaza pojedinačnog modula manja od  $p = \frac{1}{2}$ .*

**Dokaz** Verovatnoća da je modul ispravan data je sa

$$R_{mod} = (1 - p). \quad (3.1)$$

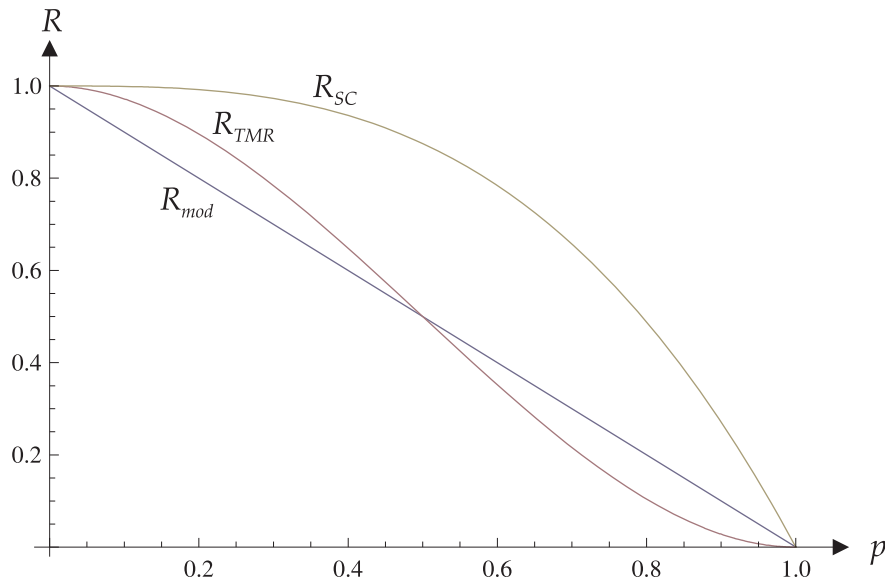
TMR sistem sa slike 3.8 ispravan je ukoliko su sva tri modula, ili ukoliko su bilo koja dva modula ispravna. Verovatnoća da su sva tri modula ispravna jednaka je proizvodu verovatnoća da je svaki od modula ispravan  $(1 - p)^3$ . Verovatnoća da su dva modula ispravna, a jedan neispravan je  $(1 - p)^2 \cdot p$ . Kako imamo tri takve kombinacije, ukupna verovatnoća da je TMR sistem ispravan, [7], je

$$R_{TMR} = (1 - p)^3 + 3 \cdot (1 - p)^2 \cdot p \quad (3.2)$$

Jednakost  $R_{mod} = R_{TMR}$  je polinom trećeg stepena čija su rešenja  $p = \{0, \frac{1}{2}, 1\}$ . Za vrednosti  $p$  iz skupa  $\{p \in \mathbb{R} \mid 0 < p < \frac{1}{2}\}$ , verovatnoća da je TMR sistem ispravan ( $R_{TMR}$ ) veća je od verovatnoće da je pojedinačni modul ispravan ( $R_{mod}$ ), i obrnuto za vrednosti  $p$  iz intervala  $(\frac{1}{2}, 1)$ , što dokazuje lemu. Grafik funkcija  $R_{mod}(p)$  i  $R_{TMR}(p)$  prikazan je na slici 3.10.  $\square$

N-Modularna redundansa (N-Modular Redundancy - NMR) predstavlja generalizaciju TMR-a. NMR koristi iste principe kao i TMR, ali koristi  $N$  modula. Uglavnom, za  $N$  se bira neparna vrednost, tako da se mogu primeniti principi većinskog glasanja. Prednost korišćenja  $N$  modula umesto samo tri je da se može tolerisati kvar više od jednog modula.





Slika 3.10: Uporedni prikaz pouzdanosti osnovnog modula  $R_{mod}$ , TMR sistema  $R_{TMR}$  i sistema sa rezervnim modulima  $R_{SC}$

### Aktivna redundansa i metod rezervnih komponentata

Kod aktivnih, odnosno dinamičkih tehnika, otpornost na defekte postiže se detektovanjem postojanja defekta i preduzimanjem akcija uklanjanja (ili izolovanja) neispravnog hardvera iz sistema, odnosno rekonfiguracijom sistema. Aktivna hardverska redundansa koristi detekciju defekta, lokaciju defekta, i sanaciju defekta u cilju postizanja tolerancije na defekte. Aktivna hardverska redundansa ne poseduje osobinu maskiranja defekta i primenjuje se kod sistema koji mogu tolerisati privremene pogrešne rezultate za vreme koje je potrebno da se sistem rekonfiguriše i povrati svoj operativni status [8, 12].

Osnovni princip udvostručavanja sa poređenjem, prikazan na slici 3.11, je ugraditi dva identična hardverska modula, pustiti ih da simultano rade ista izračunavanja, i zatim uporediti njihove rezultate. U slučaju neslaganja, generiše se signal greške. U svom najosnovnijem obliku, princip udvostručavanja može samo detektovati postojanje kvara, a ne i otkloniti ga, jer ne postoji način za određivanje koji od dva modula je u kvaru.

Aktivnost ove metode ogleda se u tome da je jedan modul operativan, dok drugi modul/moduli mogu služiti kao rezerva (eng. *Spare Component - SC*). Termin "rezervni modul" koristi se kod aktivnih tehnika za opis redundantnih modula, koji, za razliku od redundantnih modula kod pasivnih tehnika,

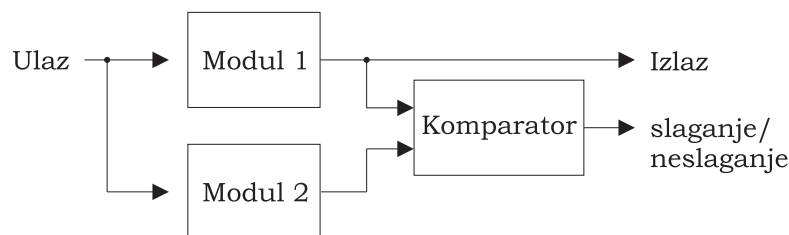
ne moraju sve vreme biti u funkciji. Naime, kod aktivnih tehnika rezervni moduli mogu postati funkcionalni tek nakon detektovanja greške. Ako je defekt detektovan i lociran, неисправan modul se uklanja iz sistema i zamenjuje se rezervnim. Ova metoda može vratiti sistemu punu operativnu sposobnost posle pojave kvara, ali izaziva trenutno narušavanje performansi dok se izvodi rekonfiguracija. Ako ovo narušavanje rada mora biti minimizirano, primenjuje se "vruća spremna rezerva". Naime, kod ove tehnike, rezervni moduli rade sinhrono sa aktivnim modulima, i spremni su da preuzmu sistem bilo kada. Nasuprot ovome, imamo "hladnu spremnu rezervu" kod koje su rezervni moduli neaktivni, sve dok se ne javi potreba da zamene неисправni modul.

**Lema 3.2 (Pouzdanost SC sistema)** *Sistem nad kojim je primenjena SC tehnika sa dva rezervna modula pouzdaniji je od osnovnog modula za sve vrednosti verovatnoće  $p$  iz intervala  $[0, 1]$ .*

**Dokaz** Verovatnoća ispravnosti pojedinačnog modula opisana je izrazom (3.1). SC sistem je ispravan ukoliko je bar jedan od modula (osnovni ili rezervni) ispravan. Uzmimo SC sistem sa dve rezervne kopije. Verovatnoća da su sva tri modula ispravna (osnovni i dve kopije) jednaka je proizvodu verovatnoća da je svaki od modula ispravan  $(1 - p)^3$ . Verovatnoća da su dva modula ispravna, a jedan неисправan je  $(1 - p)^2 \cdot p$ . Verovatnoća da su dva modula неисправna, a jedan ispravan je  $(1 - p) \cdot p^2$ . Za poslednje dve navedene verovatnoće imamo po tri kombinacije, pa je ukupna verovatnoća da je SC sistem ispravan, [7], je

$$R_{TMR} = (1 - p)^3 + 3 \cdot (1 - p)^2 \cdot p + 3 \cdot (1 - p) \cdot p^2 \quad (3.3)$$

Grafik funkcija (3.1) i (3.3) prikazan je na slici 3.10.  $\square$



Slika 3.11: Model aktivne hardverske redundanse sa jednim rezervnim modulom

## Hibridne tehnike

Hibridna tehnika koristi dobre osobine pasivnog i aktivnog prilaza. Maskiranje greške se koristi u hibridnim sistemima da spreči generisanje pogrešnih rezultata. Onda kada maskiranje nije više u stanju da obezbedi ispravan rad sistema, primenjuje se neka aktivna tehnika kako bi se defekt locirao i sanirao zamenom neispravnog hardvera rezervnim. U osnovi, hibridna hardverska redundansa koristi dobre osobine i pasivne i aktivne hardverske redundanse. Maskiranje greške se koristi da spreči sistem da proizvede pogrešne rezultate, a detekcija, lokacija i sanacija greške koriste se za rekonfiguraciju sistema u slučaju pojave greške [8, 9, 12, 73].

Princip NMR sa rezervama je obezbediti konfiguraciju od  $N$  modula uključenih u glasanje. Glasanje obezbeđuje maskiranje neispravnog modula za vreme dok se neispravn modul ne zameni rezervnim.

Osnovni princip redundanse sa samoizbacivanjem sličan je principu koji imamo kod NMR-a sa rezervama. Razlika je u tome što kod tehnike samoizbacivanja sve jedinice aktivno učestvuju u radu sistema, dok kod NMR-a rezervne jedinice nisu aktivne sve dok se ne pojavi greška. Svaki od  $N$  identičnih modula je projektovan sa sposobnošću da ukloni sebe iz sistema u slučaju da se njegov izlaz razlikuje od izglasanog izlaza sistema [12].

Redundansa sa "prosejavanjem" modula je tehnika koja takođe koristi  $N$  identičnih modula koji su povezani u sistem korišćenjem specijalnih blokova: komparator, detektor i kolektor. Uloga komparatora je da upoređuje izlaz svakog od modula sa izlazima preostalih modula, i pri tome proizvodi po jedan signal za svako poređenje koje je izvršio. Signal koji generiše komparator je 1 ako se dve poređene jedinice ne slažu, a 0 u suprotnom slučaju. Uloga detektora je da odredi koja neslaganja je komparator prijavio, i da onesposobi jedinicu koja se ne slaže sa većinom preostalih modula. Detektor proizvodi po jedan signal za svaki modul, i vrednost tog signala je 1 ako se dati modul ne slaže sa većinom preostalih modula, i 0 u suprotnom. Uloga kolektora je da proizvede izlaz sistema, koristeći pri tome izlaze pojedinačnih modula i signale iz detektora koji indiciraju koji je od modula neispravan. Modulu koji je identifikovan kao neispravan nije dozvoljeno da utiče na izlaz sistema [12].

Tehnika utrostručeno-dvostruka arhitektura kombinuje udvostručavanje sa poređenjem i trostrukom modularnu redundansu. Po dva modula rade u paru. Komparator pridružen paru modula poredi njihove rezultate i isključuje par iz sistema ukoliko ustanovi neslaganje. Upotreba TMR-a dozvoljava maskiranje greške, i neprekidan, pravilan rad sa jednim neispravnim modulom [12].

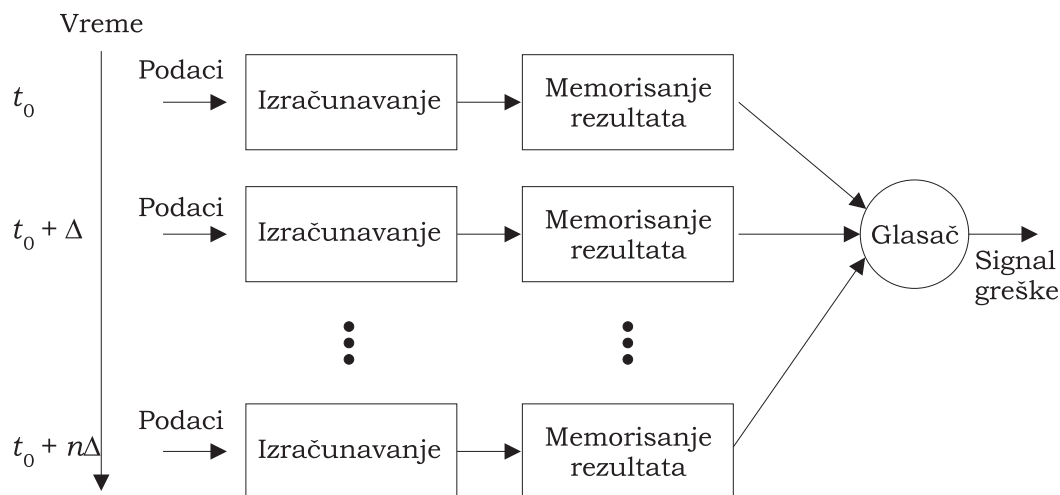
Za proveru rezultata ovog rada biće korišćeni metodi hardverske redun-

danse: TMR, kao predstavnik pasivnih tehnika, i SC kao predstavnik aktivnih tehnika. U cilju dobijanja rezultata u slučaju SC tehnika koji su poredivi sa rezultatima TMR tehnike, biće korišćena SC tehnika sa dva rezervna modula, tako da ceo sistem ima ukupno tri kopije modula (skrać. SC3). Radi celovitosti izlaganja, u narednom poglavlju dat je kratak pregled tehnika informacione, vremenske i softverske redundanse.

### 3.3.2 Informaciona, vremenska i softverska redundansa

Kako bi se ostvarili zahtevi za visokom pouzdanošću sistema i obezbedio integritet podataka, kod informacione redundanse bitovima se pridružuju redundantni bitovi, koji se koriste za detekciju i/ili korekciju greške. Dobri primeri informacione redundanse su kodovi za detekciju, kao i kodovi za korekciju grešaka, koji se formiraju dodavanjem redundantnih informacija rečima, ili prevođenjem reči u neki novi oblik koji sadrži redundantne informacije [76, 77, 78].

Osnovni princip vremenske redundanse je, korišćenjem istog hardvera i istih ulaza, izvršavati ista izračunavanja u dva ili više vremenskih trenutaka i upoređivati rezultate u cilju određivanja eventualnog neslaganja, kao što je prikazano na slici 3.12. Ako se detektuje greška, izračunavanje se može ponoviti kako bi se videlo da li neslaganje ostaje, ili nestaje. Ovakav pristup je dobar za detekciju grešaka nastalih usled nekog prolaznog, tzv. tranzijentnog defekta, ali se u osnovnoj varijanti ne može iskoristiti za detekciju grešaka izazvanih stalnim defektom.



Slika 3.12: Osnovni princip vremenske redundanse

U literaturi se mogu naći rešenja koja prevazilaze ovaj problem manipulišući operandima u različitim iteracijama algoritma sa slike 3.12 [79].

Softverska redundansa predstavlja dodavanje softvera, pored onog neophodnog za obavljanje zadatah funkcija, a u cilju detekcije i tolerancije defekata. Mnoge tehnike za detekciju i toleranciju defekata mogu biti implementirane u softveru. Redundantni softver se može realizovati u više oblika, i nije potrebno replicirati celokupne programe da bi dobili redundantni softver. Softverska redundansa se može realizovati dodatnim kodom koji se koristi za proveru rezultata obrade.

Postoje tri glavne tehnike softverske redundanse: kontrola doslednosti, kontrola sposobnosti, i metoda replikacije softvera, odnosno N-verziona programiranje [8].

U narednom poglavlju biće data kvantitativna analiza pouzdanosti sistema kod koga se metodi za postizanje visoke pouzdanosti sistema primenjuju nad delovima sistema.

## 3.4 Analiza pouzdanosti sistema

Lemom 3.1 pokazano je da je pouzdanost TMR sistema veća od pouzdanosti pojedinačnog modula za verovatnoće otkaza  $p < 1/2$ , dok je lemom 3.2 pokazano da je SC3 sistem pouzdaniji za ceo skup verovatnoća  $p \in [0, 1]$ . Granični slučajevi su verovatnoće  $p = 0$ , kada sigurno ne postoji defekt, i  $p = 1$ , kada sigurno postoji bar jedan neispravan modul. U oba ova slučaja verovatnoća da je osnovni sistem ispravan je jednaka verovatnoći da je visokopouzdan sistem ispravan. Realno je da u procesu proizvodnje čipova verovatnoća otkaza bude daleko manja od  $p = 0.5$ . U suprotnom takav proces proizvodnje ne bi bio ekonomski isplativ [11, str. 833].

Lemama 3.1 i 3.2 pokazana je pouzdanost sistema ukoliko se tehnike za postizanje visoke pouzdanosti sistema primenjuju nad sistemom kao celinom. Tehnike projektovanja visokopouzdanih sistema moguće je primeniti na različitim nivoima sistema. U ovom poglavlju data je kvantitativna analiza pouzdanost sistema kod koga su pomenute tehnike primenjene nad delovima koji čine sistem.

### 3.4.1 Minimalna jedinica zamene

Neki defekti mogu biti kritični za ceo sistem. Za razliku od kombinacione mreže i memorijskih elemenata koji čine sistem, infrastruktura čipa je obično jedinstvena. Infrastrukturu čini distribucija napajanja i mase, kao i mreža za taktovanje delova sistema. Mnogi infrastrukturni delovi su po površini čipa

koju zauzimaju mnogo veći od ostalih komponenti. Kao takvi, obično su i otporniji na defekte od manjih komponenti koje čine sistem [11, str. 832].

Prilikom projektovanja sistema korisno je razdvojiti zamenljive od nezamenljivih resursa. Za svaki dizajn potrebno je definisati minimalnu jedinicu zamene, koja određuje granulaciju zamene rezervnim komponentama [11, str. 832].

**Definicija 3.7 (Minimalna jedinica zamene)** *Minimalna jedinica zamene je najmanji podsistem koji se može zameniti u slučaju otkaza.*

Na primer, kod memorijskih modula ne može se izvršiti zamena pojedinih ćelija. Kod memorija se koristi tehnika poznata kao tehnika rezervnih vrsta (eng. *row sparing*), gde se rezervnom vrstom zamenjuje cela vrsta koja sadrži neispravnu ćeliju [11, str. 832], [80, str. 85].

Najjednostavnija šema koje uključuju zamenu odbaciće ceo sistem u slučaju pojave defekta bilo gde u sistemu. Međutim, glanuralnost dizajna i veličina minimalne jedinice zamene imaju ključnu ulogu u projektovanju visokopouzdanog sistema.

### 3.4.2 Pouzdanost sistema sa N jedinica zamene

U ovom poglavlju data je jednostavna analiza pouzdanosti sistema sa minimalnom jedinicom zamene manjom od veličine samog sistema.

Pretpostavimo da je uniformna i nezavisna raspodela verovatnoće pojave defekta na celoj površini čipa. Korišćenjem jednostavnih modela moguće je ilustrovati glavne trendove koji definišu stepen pojave defekata koji visokopouzdan sistem može tolerisati, a samim tim i steći utisak o kvalitetu procesa proizvodnje koji mora biti zadovoljen kako bi proizvodnja sistema bila ekonomski isplativa.

**Definicija 3.8 (Savršena pouzdanost)** *Sistem koji ne sadrži rezervne komponente ispravan je samo ako su svi elementi koji čine sistem ispravni [11, str. 833].*

Neka se sistem sastoji od  $N$  podistema, i neka je  $p$  verovatnoća pojave defekta u svakom od podistema. Po definiciji 3.8 verovatnoća da je sistem u celosti ispravan je

$$R(p, N) = (1 - p)^N. \quad (3.4)$$

Jednačinu (3.4) moguće je razviti u binomni red

$$R(p, N) = \sum_i \binom{N}{i} (-p)^i = 1 - N \cdot p + \binom{N}{2} p^2 - \dots$$

Ukoliko je  $N \times p \ll 1$  svaki naredni član sume sadrži verovatnoću  $p$  sa većim stepenom i zanemarljivo je mali u odnosu na prethodni član, pa prethodnu jednačinu možemo aproksimirati kao

$$R(N, p) \approx 1 - N \cdot p, \quad (3.5)$$

što predstavlja aproksimaciju pouzdanosti sistema sa  $N$  podsistema.

Jednačina 3.5, grubo rečeno, govori da je verovatnoća otkaza sistema sa  $N$  podsistema jednaka

$$p_N = N \cdot p.$$

Iz ovog jednostavnog izraza moguće je zaključiti sledeće [11, str. 833]:

- Kod današnjih sistema sa  $N > 10^9$  komponentata, verovatnoća otkaza pojedinih komponenti mora biti ispod  $10^{-10}$  kako bi bilo moguće očekivati 90 procenata, ili više ispravnih čipova.
- Da bi održali pouzdanost sistema konstantnim u slučaju povećanja broja komponenti sistema, neophodno je verovatnoću pojave defekta umanjiti za isti faktor.
- Manji sistemi sa većim brojem komponentata imaju veću verovatnoću otkaza.
- Da bi se pouzdanost sistema zadržala u slučaju povećanja verovatnoće pojave defektne komponente, granulacija sistema mora biti finija (minimalna jedinica zamene manja).

**Lema 3.3 (Pouzdanost sistema sa  $N$  podsistema)** *Pouzdanost sistema sa  $N$  identičnih podsistema, može se povećati dodavanjem rezervnih podsistema, takvih da svaka rezerva može zameniti bilo koji od  $N$  podsistema [11, str. 834].*

### Dokaz

Pretpostavimo da se sistem sastoji od ukupno  $N$  podsistema, od kojih je  $M$  osnovnih i  $(N - M)$  rezervnih, i važi ( $M \leq N$ ). Verovatnoća da će sistem imati tačno  $i$  ispravnih podsistema je

$$R(i, N) = \left( \binom{N}{i} p^i (1 - p)^{N-i} \right).$$

To znači da postoji ukupno  $\binom{N}{i}$  načina kako se može rasporediti  $i$  nedefektnih podsistema, a verovatnoća svakog od ovih slučajeva je  $p^i (1 - p)^{N-i}$ .

Verovatnoća da će sistem imati manje od  $M$  neispravnih modula, neophodnih za ispravno funkcionisanje sistema jednaka je zbiru verovatnoća da sistem ima  $M$  ili više ispravnih modula

$$R_G(M, N) = \sum_{M \leq i \leq N} R(i, N) = \sum_{M \leq i \leq N} \left( \binom{N}{i} p^i (1-p)^{N-i} \right). \quad (3.6)$$

Ukoliko je  $N > M$ , tj. sistem ima bar jedan rezervni modul, iz (3.6) sledi da je  $R_G(M, N) > R_G(N, N)$ , bar za  $R(N-1, N)$ , što potvrđuje lemu.  $\square$

**Primer** Razmotrimo primer FPGA čipa na kome je implementiran sistem sa ukupno 10 LUT [11, str. 834]. Neka su 2 dodatne LUT pridružene sistemu u ulozi rezervnih komponenti, i neka ove LUT mogu zameniti bilo koju od 10 osnovnih LUT. Neka je verovatnoća otkaza komponente  $10^{-4}$ . Tada, za sistem bez rezervnih komponenti iz (3.6) sledi

$$R_G(10, 10) = (10^{-4})^{10} (1 - 10^{-4})^0 \approx 0.9990005.$$

Pouzdanost sistema sa rezervnim komponentama na osnovu (3.6) je

$$\begin{aligned} R_G(10, 12) &= (10^{-4})^{12} (1 - 10^{-4})^0 + \\ &+ (10^{-4})^{12} (1 - 10^{-4})^1 + (10^{-4})^{12} (1 - 10^{-4})^2 \approx \\ &\approx 0.9999999998 > 1 - 10^{-9}. \end{aligned}$$

$\triangle$

Bez korišćenja rezervnih komponenti teško je postići zadovoljavajuću pouzdanost i kod sistema koji imaju i do 1.000 komponenti. Sa rezervnim komponentama moguće je postići pouzdanost i do 99,9% u slučajevima kada broj komponenti prelazi 1.000.000 [11, str. 834]. Međutim, pretpostavka da svaka rezervna komponenta može da zameni bilo koju komponentu sistema je veoma teško ostvariva u praksi. Da bi se ovo postiglo potrebno je projektovati veoma kompleksnu mrežu veza na čipu i dodati podsistem za upravljanje tom mrežom. U praksi se uglavnom ovaj problem rešava kompromisom, tako da se rezervne kopije lokalizuju, kako bi mogle zameniti bilo koju komponentu, ali samo u određenom delu sistema. Granični slučaj ovakve lokalizacije je primena sistema sa slike 3.11 nad svakom komponentom posebno.

Više detalja o uticaju veličine oblasti na koju je lokalizovana rezervna kopija moguće je naći u *Bernoullijevom zakonu velikih brojeva* (eng. *Law of Large Numbers, the Central Limit Theorem*) [81], gde se kaže da vrednost sume slučajno izabranih promenljivih opada sa porastom broja promenljivih. Za proveru rezultata ovog rada koristićemo jednostavan model gde je veličina minimalne jedinice zamene jednaka jednoj ćeliji polja.



## 3.5 Tolerancija grešaka

Od samog začetka digitalne elektronike postoji interesovanje za ponašanje sistema u slučaju otkaza pojedinih komponenti. Godine 1956. von Neumann je u [10] razmatrao problem dobijanja tačnog rezultata u slučaju otkaza pojedinih komponenti (releja u to vreme). Naumann je u [10] predstavio NAND redundantno multipleksiranje i dokazao da se ograničavanjem mogućnosti za prostiranje greške povećava verovatnoća dobijanja tačnog rezultata. Od tog vremena radovi u ovoj oblasti su uglavnom posvećeni tehnikama projektovanja FT sistema, sa ciljem dobijanja rezultata bez ijedne greške.

### 3.5.1 Neprecizna izračunavanja

Vrednost određenih atributa, kao što su dužina, težina, površina, zapremina i sl., hiljadama godina unazad određuju se na veoma neprecizan način. Nekada je kamen služio kao mera za težinu, a dužina ruke od vrha prstiju do lakta kao mera za dužinu. Razvoj tehnike je, međutim, nametnuo u velikom broju slučajeva nepotreban zahtev za velikom preciznošću. Ovaj problem se poslednjih godina dosta razmatra u literaturi, a neki autori idu dotle da procenjuju koliko ljudi veruje kalkulatoru da je vrednost izraza  $(1 + 4/3) \cdot 3$  jednaka 6.9999 [13].

Zahtev za sve preciznijim izračunavanjima, koji je nametnut razvojem digitalne tehnike, nije uvek i u svakoj situaciji neophodan. Naime, par vekova unazad istraživači poput Kristofera Kolumba su tokom dugih plovidbi imali potrebu za nešto preciznijim određivanjem vremena od posmatranja pozicije sunca i posmatranja senki. Iako su satovi postojali u to vreme, greška koja se javljala u pokazivanju vremena u toku trajanja dugih plovidbi bila je neprihvatljiva. U to vreme se došlo na ideju da se na duge plovidbe nosi više od jednog sata. Deo satova kasni, deo žuri, ali je prosek dovoljno dobar za tako specifičnu namenu, bez obzira što nije u potpunosti tačan [13]. U savremenom računarstvu ova tehnika je poznata pod nazivom "dogovorna aproksimacija" (eng. *approximate agreement*), i ogleda se u određivanju srednje vrednosti izračunavanja, koja je u slučaju otkaza delova sistema dovoljno dobra za konkretnu primenu [82].

Postoji veliki broj radova koji se bave tzv. "nepreciznim izračunavanjima" [13, 83, 84, 85, 86]. U [83] predstavljen je koncept smanjenja pouzdanosti sistema sa ciljem smanjenja potrošnje kola. Rad je baziran na pretpostavci da je energija potrebna da bi se izračunavanje obavilo proporcionalna tačnosti izračunavanja koju sistem nudi. Smanjenjem pouzdanosti izračunavanja, koje je moguće postići dodatnim upravljačkim kolima, moguće je smanjiti potrošnju sistema u slučajevima kod kojih je opravdano uvesti ovakav tip

kompromisa.

Sličan koncept primenjen je u [84], gde su autori istraživali uticaj smanjenja napona napajanja, uz konstantnu frekvencu, na broj grešaka koje sistem proizvodi. Drastičnim smanjenjem napona napajanja broj grešaka izračunavanja raste. Napon napajanja se smanjuje dok je amplituda i učestanost pojavljivanja grešaka prihvatljiv za aplikaciju. Autori su u [84] pokazali da se na ovakav način, po ceni smanjenja tačnosti izračunavanja, značajno može smanjiti potrošnja u kolu.

Još jedan primer sistema kod koga postojanje defekta koji prouzrokuje greške ne mora nužno voditi pogrešnom rezultatu su neuronske mreže. Neuronske mreže su veoma robusne po pitanju šumova kod ulaznih podataka, kao i po pitanju defektnog hardvera. M. Saidiku i M. Mazzara navode u radu [85] da neuronske mreže tipično nude veći stepen otpornosti na defekte od von Neumannovih sekvencijalnih računara zbog velikog broja procesnih elemenata sa, uglavnom, lokalnim vezama. Postojanje nekoliko defektnih procesnih elemenata i veza ne utiče nužno na performanse celog sistema [85].

Ljudska čula, kao što su vid i sluh su u velikoj meri tolerantna na spoljne šumove. Audio i video aplikacije su velika klasa aplikacija kod kojih se često na algoritamskom nivou koristi činjenica da ljudski sluh i vid mogu tolerisati i prepoznati signale sa velikim odnosom signal-šum. Korišćenjem ove činjenice audio i video aplikacije postižu velike stepene kompresije podataka [13, 86]. Chung i Ortega su u [86] razmatrali uticaj jednostrukih i višestrukkih grešaka u bloku za određivanje pomeraja (eng. *Motion Estimator - ME*) za H.264 video kodek [16]. Rezultati do kojih su došli ukazuju na to da veoma mali broj grešaka bitno utiče na performanse sistema. Greške u ME bloku utiču na smanjenje stepena kompresije, što je u vezi sa kapacitetom veze koja je potrebna za prenos video sekvence [86].

Generalno, postoji nekoliko razloga zbog kojih treba razmatrati digitalna izračunavanja kod kojih se može javiti greška u izračunavanju. Neki od razloga pomenuti su u prethodnom tekstu i odnose se na smanjenje potrošnje, kao i heurističke algoritme, gde je aproksimacija sa greškom i dalje "dovoljno dobra" aproksimacija. Međutim, nove tehnologije poput kvantnih i bio-računara, ali i novih trendova razvoja CMOS tehnologije, zbog same prirode tehnologije, nameću potrebu za sistematskim tehnikama koje razmatraju pojavu i prostiranje greške kroz sistem [13]. Kod novih CMOS tehnologija, gde na čipu postoji i do par milijardi komponenata, koncept "ispravnog čipa" postaje besmislen, ili u krajnjem slučaju neopravdano skup.

### 3.5.2 Sistemi tolerantni na greške

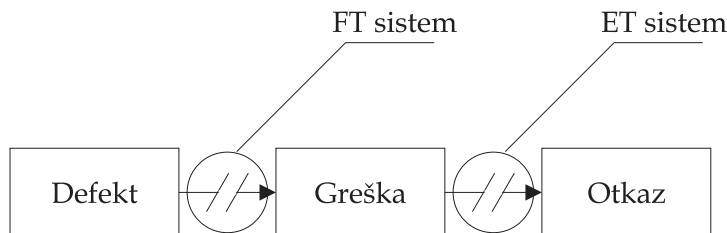
Istraživanja sistema kod kojih projektant svesno toleriše postojanje grešaka u sistemu zarad postizanja nekog cilja intenzivirana su 2001. godine, nakon publikacije o trenutnom stanju u tehnologiji izrade čipova [5].

U najznačajnije probleme u procesu fabrikacije čipova se u [5] svrstavaju varijacije u proizvodnji koje dovode do razlika u istoj seriji proizvoda, stepen defekata koji se javlja, škart, kao i osetljivost samog procesa na interne i eksterne smetnje. "Ovakav trend će po svemu sudeći smanjiti efektivnu ekonomsku isplativost budućih tehnologija. Trend razvoja tehnologije nameće zaključak da će proizvodnja integrisanih kola uskoro dostići nivo od više desetina milijardi tranzistora po čipu, među kojima je par hiljada neispravnih.", kako navode u [5], "Proizvodnja čipa sa 100% ispravnim komponentama i vezama postaće veoma skupa." Uz trend razvoja integrisanih kola, koji je sve bliži fizičkim granicama koje je teoretski trenutnom tehnologijom moguće postići, projektanti će morati da ovu činjenicu uvrste u projektne zahteve i na arhitekturnom nivou [5, 7, 6].

U odnosu na **sisteme otporne na otkaze** (FT), opisane definicijom 3.6 u poglavlju 3.3, čiji je cilj povećanje pouzdanosti uvođenjem nekog vida redundanse (poglavljje 3.3), projektovanje **sistema tolerantnih na greške** je dijametralan pristup i može se definisati na sledeći način [20, 14, 15]:

**Definicija 3.9 (Sistem tolerantan na greške)** *Za sistem sa defektima, koji prouzrokuju propagaciju grešaka do izlaza sistema, kaže se da je tolerantan na greške (eng. Error Tolerant - ET) ukoliko su rezultati takvi da su za datu primenu sistema dovoljno dobri [7].*

Za razliku od FT sistema, ET sistemi dopuštaju pojavu grešaka, ali ublažavaju pojam "otkaz" sistema. Kako je na slici 3.13 ilustrovano, cilj FT sistema je raskid uzročno-posledične veze između defekta i greške, dok je cilj ET sistema raskid (ublažavanje) uzročno-posledične veze između greške i otkaza.



Slika 3.13: Pozicija ET i FT sistema u modelu defekt-greška-otkaz

U daljem tekstu dat je primer primene sistema tolerantnih na greške.

### 3.5.3 Primer sistema tolerantnog na greške

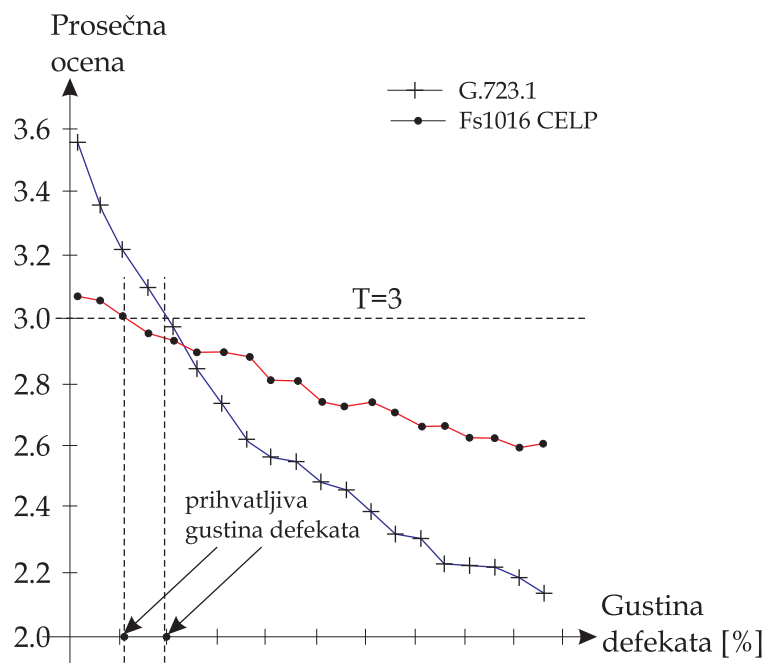
Primena sistema tolerantnih na greške i nivo same tolerancije na greške definisana je funkcijom koju sistem obavlja. Drugim rečima, jedan isti DSP algoritam u različitim aplikacijama može imati različite tolerancije na greške. Očigledan je primer debloking filtra u H.264 video koderu/dekoderu [16, 17, 18, 87, 21]. Naime, ukoliko sistem realizuje funkciju debloking filtra u TV prijemniku u HD (eng. *High Definition*) rezoluciji sa 40" dijagonalom, bloking efekat koji se javlja na slici je uočljiviji i neprihvatljiviji u odnosu na isti sistem kod prenosivog uređaja sa daleko manjom rezolucijom i dijagonalom ekrana.

U ovom poglavlju prikazan je primer ugradnje defektne memorije u automatske telefonske sekretarice.

Okruženje za testiranje prihvatljivosti ET sistema prikazano je u [15] i ilustrovano na primeru koncepta koji je patentiran 1995. godine [88]. U apstraktu patenta se navodi: "U audio sistemima koji koriste memorijske čipove za smeštanje digitalizovanih audio signala, prihvatljivo je koristiti memorije sa jednom ili više neispravnih memorijskih vrsta. Grešku koja se javlja prilikom reprodukcije ovakvog audio signala korisnik teško može primetiti, kako zbog brzine uzorkovanja signala, tako i zbog maksimalnog dozvoljenog broja grešaka."

U cilju analize sistema sa defektnom memorijom u [15] usvojeno je pet različitih ocena kvaliteta audio signala: 1 (loš), 2 (slab kvalitet), 3 (prosečan), 4 (dobar), i 5 (odličan). Analiziran je sistem sa ITU-T G.723.1 audio kodekom, kao delom H.324 standarda za prenos multimedijalnog sadržaja. Izabrani nivo kompresije u ovom primeru je takav da originalnu audio sekvencu sa 64 Kbps svodi na 5.3 Kbps [89].

Veličina simulirane memorije je 100 Kbit. Defekti su modelirani kao defekti tipa "kratak spoj na 0" i "kratak spoj na 1" (eng. *stuck-at model*). Gustina defekata je definisana kao odnos između broja defekata i veličine memorije. U radu [15] su prikazani rezultati dvadeset različitih gustina defekata u opsegu od 0% do 1%. Ocena prihvatljivosti uzeta je kao prosečna ocena data od strane 26 ispitanika. Rezultati su prikazani na slici 3.14. Prihvatljiva gustina defekata je definisana kao vrednost za koju je prosečna ocena jednaka, ili veća od praga tolerancije  $T$ . Na slici 3.14, prag tolerancije je postavljen na  $T = 3$  i prikazan za G.723.1 i FS1016 CELP algoritme [15].



Slika 3.14: Rezultati subjektivne ocene kvaliteta digitalne telefonske sekretarice sa defektnom memorijom

# Glava 4

## Parcijalno visokopouzdan sistemi

Kako je u prethodnom poglavlju rečeno, von Neumann je u [10] razmatrao problem dobijanja tačnog rezultata u slučaju otkaza pojedinih komponenti. Neumann je u [10] predstavio NAND redundantno multipleksiranje i dokazao da se ograničavanjem mogućnosti za prostiranje greške povećava verovatnoća dobijanja tačnog rezultata. Od tog vremena radovi u ovoj oblasti su uglavnom posvećeni tehnikama projektovanja FT sistema, sa ciljem određivanja rezultata bez ijedne greške.

Nepouzdanost savremenih tehnologija je poslednjih godina indukovala razvoj novih metoda za povećanje ekonomske isplativosti fabrikacije računarskih arhitektura. U [5] je publikovana konstatacija da razvoj i usložnjavanje tehnologije vodi tome da je sve teže i skuplje proizvesti čip sa 100% ispravnim komponentama. Serija radova publikovanih u proteklih par godina, u cilju povećanja ekonomske isplativosti fabrikacije računarskih arhitektura evaluira i redefiniše stanje otkaza sistema [4, 6, 19, 20, 13, 14, 83]. Tako, otkaz sistema postaje relativna kategorija, pa se za defekte u jednom slučaju kaže da su doveli do otkaza sistema, a u drugom slučaju, za primenu sistema gde greške uzrokovane tim istim defektima nisu od velikog značaja, kaže se da nije došlo do otkaza sistema (def. 3.9).

Ukoliko se u razmatranje uzme prethodno opisana relativnost otkaza, otkaz sistema je potrebno je redefinisati u odnosu na definiciju 3.3 i posmatrati ga u odnosu na grešku koja se javlja na izlazu sistema, u skladu sa definicijom 3.9.

## 4.1 Metrika otkaza sistema

Otkaz (def. 3.3) se može meriti subjektivnim ili objektivnim metodama. Jedan primer subjektivnih metoda prikazan je u poglavlju 3.5.3, gde se kvalitet sistema određuje na osnovu subjektivnih ocena određenog broja ispitnika. Objektivnim metodama, znajući karakteristike sistema, moguće je meriti, ili matematički odrediti kvalitet sistema. Kvalitet sistema u tehnici se često izražava kao odnos signal-šum (eng. *Signal to Noise Ratio - SNR*) i meri u decibelima. Po definiciji, odnos signal-šum je

$$SNR_{dB} = 10 \cdot \log_{10} \frac{P_{signal}}{P_{sum}} = 10 \cdot \log_{10} \left( \frac{A_{signal}}{A_{sum}} \right)^2 = 20 \cdot \log_{10} \frac{A_{signal}}{A_{sum}}, \quad (4.1)$$

gde su  $P_{signal}$  i  $P_{sum}$  snaga signala i šuma, respektivno, a  $A_{signal}$  i  $A_{sum}$  amplitude odgovarajućih signala [90, str. 11]. Na početku ovog poglavlja definisaćemo otkaz sistema tolerantnih na greške, u skladu sa definicijom sistema tolerantnih na greške (def. 3.9).

Neka je  $\mathbf{A} \subset \mathbb{R}$  skup rezultata koji se mogu javiti na izlazu sistema, i neka je  $A_{err} \in \mathbf{A}$  rezultat izvršenja operacije  $o_i$  koji se može razlikovati od tačnog rezultata  $A_t \in \mathbf{A}$ .

**Definicija 4.1 (Otkaz ET sistema)** *Otkaz sistema tolerantnog na greške je stanje u kome sistem, prilikom izvršenja operacije  $o_i$ , na svom izlazu daje rezultat  $A_{err}$ , koji se razlikuje od tačnog rezultata  $A_t$  za*

$$d(A_t, A_{err}) > \Delta, \quad (4.2)$$

gde  $d$  preslikavanje  $d : \mathbf{A}^2 \rightarrow [0, \infty)$  takvo da zadovoljava uslove metrike ([91, str. 86, def. 4.1.1]), a  $\Delta$  maksimalna dozvoljena vrednost odstupanja od tačnog rezultata za datu primenu sistema.

Vrednost  $\Delta$  u (4.2) je razlika tačnog rezultata i rezultata koji sadrži grešku, što predstavlja šum sistema. Najčešće upotrebljavane metrike u tehnici su Euklidska i Hamingova metrika [20, 91].

**Definicija 4.2 (Otkaz ET sistema u odnosu na Euklidsku metriku)**

*Za sistem se kaže da je otkazao ukoliko važi*

$$d_E(A_t, A_{err}) > \Delta_E = 2^{L-\alpha} - 1,$$

gde je metrika  $d_E(A_i, A_j) = |A_i - A_j|$ , sa  $L$  označen ukupan broj bitova rezultata, a  $\alpha$  broj viših bitova rezultata  $A_{err}$  na kojima pojava greške znači otkaz sistema u konkretnoj primeni [20].

Na osnovu definicije 4.2, sistem nije otkazao sve dok je amplituda greške  $\Delta_E$  po apsolutnoj vrednosti manja ili jednaka  $2^{L-\alpha} - 1$ , ili drugim rečima, dok je  $\alpha$  najznačajnijih bitova tačno. SNR dat izrazom (4.1) se u tehnici često koristi za predstavljanje relativnog odnosa Euklidske metrike na logaritamskoj skali.

Problem u izražavanju greške Euklidskom metrikom u tehnici nastaje kod sistema kod kojih bitovi rezultata ne nose težinske vrednosti  $2^i$ , već označavaju kodirane simbole sa drugačijim rasporedom bitova. Primer ovakvih sistema su sistemi za Hafmanovo kodiranje, MD5 enkripciju, i sl. [20, 16].

**Primer** Razmotrimo, na veoma specifičnom numeričkom primeru, jednu osobinu Euklidske metrike. Neka je amplituda tačnog signala  $A_t = 0111 \dots 1$ . Pod uslovom da su rezultati  $A_i \in \mathbf{A}$  predstavljeni u težinskom sistemu, tako da  $j$ -ti bit ima težinu  $2^j$ , uzmimo primer greške na bitu najmanje težine  $2^0$ , tj.  $\Delta_E = +000 \dots 01$ . Izlaz sistema  $A_{err}$  će u ovom slučaju biti  $A_{err} = 100 \dots 0$ .

Posmatrano u težinskom sistemu, ispravno je reći da je greška relativno mala i da iznosi  $\Delta_E = 2^0$ . Međutim, da su u pitanju kodovi, ova greška ne bi bila beznačajna. Naprotiv, svi bitovi rezultata bili bi pogrešni, odnosno greška u ovom slučaju bila bi  $\Delta = L$ .  $\triangle$

Metrika koja ne uzima u obzir težinu je Hamingova metrika.

### Definicija 4.3 (Otkaz ET sistema u odnosu na Hamingovu metriku)

Za sistem se kaže da je otkazao ukoliko važi

$$d_H(A_t, A_{err}) > \Delta_H = 0,$$

gde je metrika

$$d_H(A_t, A_{err}) = \sum_{i=L-\alpha}^{L-1} (a_t^i \oplus a_{err}^i),$$

a  $a_j^i$  bit težine  $2^i$  rezultata  $A_j \in \mathbf{A}$  [20].

Hamingovo rastojanje definiše se kao broj različitih bitova dva broja, pa otkaz ET sistema u odnosu na Hamingovu metriku možemo definisati kao stanje u kome je  $\alpha$  najznačajnijih bitova tačno, odnosno identični su odgovarajućim bitovima tačnog rezultata [20, 91].

**Lema 4.1** *Sistem koji je ispravan po Hamingovoj metrici ispravan je i po Euklidskoj metrici, odnosno*

$$\Delta_H = 0 \Rightarrow d_E(A_t, A_{err}) \leq 2^{L-\alpha} - 1,$$

ali ne i obrnuto.



**Dokaz** Iz uslova  $\Delta_H = 0$ , na osnovu definicije 4.3, imamo da je  $\alpha$  viših bitova sigurno tačno. Greška se može javiti na nižih  $\alpha$  bitova, t.j.

$$d_E(A_t, A_{err}) \leq 2^{L-\alpha},$$

pa iz definicije 4.2 sledi tvrđenje teoreme.

Da obrnuto tvrđenje generalno ne važi vidi se iz prethodno datog numeričkog primera.  $\square$

Lemom 4.1 pokazano je da je Hamingova metrika stroža od Euklidske, odnosno da će sistem projektovan po Hamingovoj metrici sigurno zadovoljiti uslov Euklidske metrike. Propagaciju greške jednostavnije je pratiti po Hamingovoj metrici, jer se u obzir ne uzima težina greške kroz prenosnu funkciju podsistema. Zato će u ovom poglavlju projektovanje PDT sistema biti ilustrovano na primeru Hamingove metrike.

Definisanjem maksimalne dozvoljene greške na izlazu sistema ( $\Delta$ ), u zavisnosti od izabrane metrike, ET sistemi mogu povećati ekonomsku isplativost fabrikacije arhitekture, jer se određeni broj defektnih čipova ne proglašava za škart i ne odbacuje [5, 19, 20]. Kvantitativna analiza dobiti ovakvih sistema biće izložena u narednom poglavlju.

Kod ET sistema, bez obzira na metriku, mogu se razlikovati dve celine. Jedan deo sistema je onaj kod koga defekt prouzrokuje grešku manju od maksimalno dozvoljene, a drugi je onaj kod koga pojava defekta koji uzrokuje grešku dovodi do otkaza [14].

**Definicija 4.4 (Tolerantni deo arhitekture)** *Tolerantnim delom arhitekture nazivamo onaj deo kod koga pojava greške ne prouzrokuje otkaz sistema. Ostatak sistema predstavlja netolerantni deo arhitekture za datu primenu i metriku.*

U ovom radu, u cilju uštede resursa na čipu, autor predlaže primenu FT metoda samo na netolerantni deo arhitekture. Pretpostavka da ovakav pristup selekcije najznačajnijeg dela sistema i parcijalna primena FT metoda nad tim delom može dovesti do povećanju ekonomske dobiti pri fabrikaciji arhitekture biće razmatrana u narednoj glavi ovog rada. U narednom poglavlju predstavljen je koncept projektovanja parcijalno visokopouzdanih sistema, data je formalna definicija sistema, i prikazan je, i na primeru ilustrovan matematički aparat potreban za projektovanje PDT sistema.

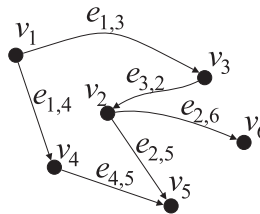
## 4.2 Definicija parcijalno visokopouzdanih sistema

U cilju definisanja parcijalno visokopouzdanih sistema, na početku ovog poglavlja biće definisani osnovni pojmovi.

Neka je arhitektura predstavljena grafom toka podataka  $\mathbf{G}$ . Neka graf  $\mathbf{G}$  čine dva podskupa,  $\mathbf{G} = (V, E)$ , gde je skup  $V = \{v_1, v_2, \dots, v_\nu\}$  konačni skup čvorova, a  $E = \{e_1, e_2, \dots, e_\varepsilon\}$  konačni skup grana. Grana  $e \in E$  je uređeni par  $(v_i, v_j)$ , gde su  $v_i, v_j \in V$ , a  $(v_i, v_j)$  znači da su čvorovi  $v_i$  i  $v_j$  povezani.

**Primer** Na slici 4.1 prikazan je primer grafa sa skupom čvorova  $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$  i skupom grana

$$\mathbf{E} = \{(v_1, v_3), (v_3, v_2), (v_1, v_4), (v_2, v_5), (v_4, v_5), (v_2, v_6)\}.$$



Slika 4.1: Primer grafa

△

**Definicija 4.5 (Propagacija greške [14])** Neka su  $v_i$  i  $v_j$  čvorovi i neka je sa  $e_{i,j}$  označen uređen par  $(v_i, v_j) \in E$ . Propagaciju greške definišemo kao relaciju

$$\xi \subseteq V^2, \quad (v_i, v_j) \in \xi,$$

ako za rezultat ispravnog izvršenja operacije u čvoru  $v_j$  važi  $d_H(A_t, A_{err}) > \Delta_H$ , ili  $d_E(A_t, A_{err}) > \Delta_E$ .

Činjenicu  $(v_i, v_j) \in \xi$  označavamo sa  $\xi_{i,j}$ , što važi ukoliko greška koja se javlja u čvoru  $v_i$  izaziva grešku u nedefektnom čvoru  $v_j$ .

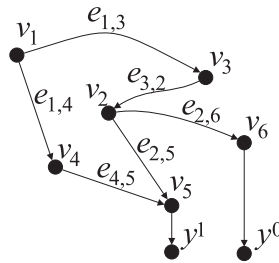
**Lema 4.2** Propagacija greške  $\xi$  je tranzitivna relacija [14]:

$$v_i, v_k, v_j \in V, \quad \xi_{i,k} \wedge \xi_{k,j} \Rightarrow \xi_{i,j}.$$

**Dokaz.** Ako  $\xi_{i,k}$  i  $\xi_{k,j}$  postoje, tada će greška u čvoru  $v_k$ , izazvana greškom u čvoru  $v_i$ , uzrokovati grešku u čvoru  $v_j$ , što dokazuje da su čvorovi  $v_i$  i  $v_j$  u relaciji  $\xi$ .  $\square$

Lemom 4.2 pokazano je da greška nastala u čvoru  $v_i$  može izazvati grešku u čvoru  $v_j$  pod uslovom da postoji put u grafu  $\mathbf{G}$  koji vodi od čvora  $v_i$  do čvora  $v_j$  preko čvora  $v_k$ , ili u opštem slučaju preko konačno mnogo čvorova  $v_{k_1}, v_{k_2}, \dots, v_{k_N}$ . Po pitanju propagacije greške kroz arhitekturu od konačnog interesa su izlazni čvorovi arhitekture.

**Primer** Na slici 4.2 prikazan je graf sa slike 4.1 sa pridruženim izlaznim čvorovima  $y^1$  i  $y^0$ . Greška nastala u čvoru  $v_1$  može se propagirati preko čvorova  $v_3, v_2$  i  $v_6$  do izlaza  $y^0$ , a preko  $v_3, v_2$  i  $v_5$  do izlaza  $y^1$ .



Slika 4.2: Graf sa pridruženim izlaznim čvorovima

$\triangle$

**Definicija 4.6 (Skup značajnih čvorova [14])** Neka je  $Y = \{y^0, y^1, \dots, y^{L-1}\}$ ,  $Y \subseteq V$  skup izlaznih čvorova. Skup  $\mathbf{M}_\eta \subseteq V$  nazivano skup značajnih čvorova za izlazni bit  $y^\eta$  po pitanju propagacije grešaka, akko

$$v_i \in \mathbf{M}_\eta \Leftrightarrow (v_i, y^\eta) \in \xi.$$

Skup značajnih čvorova (def. 4.6) izdvaja deo arhitekture u kome će pojava greške po izabranoj metrici prouzrokovati grešku na izlazu  $y^\eta$ . Ovo je deo arhitekture nad kojim treba primeniti neki od metoda visoke pouzdanosti kako bi se smanjila verovatnoća pojave greške na izlazu  $y^\eta$ .

**Primer** Po definiciji 4.5 za primer sa slike 4.2 važi

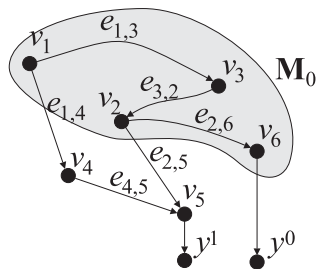
$$\{(v_1, y^0), (v_2, y^0), (v_3, y^0), (v_6, y^0)\} \in \xi.$$

Na osnovu definicije 4.6, skup značajnih čvorova izlaznog čvora  $y^0$  grafa sa slike 4.2 je

$$\mathbf{M}_0 = \{v_1, v_2, v_3, v_6\}.$$

Ovaj skup je označen je na slici 4.3.

$\triangle$



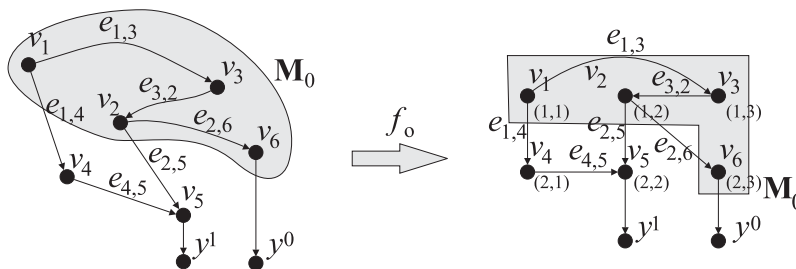
Slika 4.3: Primer skupa značajnih čvorova  $M_0$  izlaznog čvora  $y^0$

U cilju pojednostavljenja operacija za manipulaciju putevima u grafu, uvedimo funkciju za prostorno uređenje čvorova  $f_o : V \rightarrow \mathbb{N}^2$ , gde je  $\mathbb{N}$  skup prirodnih brojeva [14]. Funkciju  $f_o$  nazivamo funkcijom prostornog uređenja, koja preslikava skup čvorova arhitekture  $V$  na dvodimenzionalni prostor, indeksiran uređenim parovima  $(p, q) \in \mathbb{N}^2$ .

**Primer** Na slici 4.4 prikazan je jedan mogući način dvodimenzionalnog uređenja grafa  $G$  sa slike 4.3 funkcijom  $f_o$ . Parovi  $(p, q) \in \mathbb{N}^2$  naznačeni su u donjem desnom uglu svakog čvora. △

Funkcija uređenja  $f_o$  prostorno uređuje čvorove grafa tako da je grafu moguće dodeliti matricu čiji elementi poziciono odgovaraju čvorovima uređenog grafa. Drugim rečima, informacije o uticaju čvorova moguće je mapirati na elemente matrice. Ovakva matrica opisuje pripadnost čvora skupu značajnih čvorova određenog izlaza (def. 4.6), a formalno je moguće definisati na sledeći način.

**Definicija 4.7 (Mapa uticaja greške [14])** Mapu uticaja greške,  $M_\eta =$



Slika 4.4: Primer funkcije uređenja skupa

$(m_{p,q}^\eta)$ , za izlazni čvor  $y^\eta$ , definišemo kao matricu sa elementima

$$m_{p,q}^\eta = \begin{cases} 1, & \exists v_i \in \mathbf{M}_\eta, f_o(v_i) = (p, q) \\ 0, & \forall v_i \in \mathbf{M}_\eta, f_o(v_i) \neq (p, q) \end{cases} .$$

Mapa uticaja greške (def. 4.7) je matrica čiji elementi poziciono odgovaraju čvorovima grafa, prostorno uređenih funkcijom  $f_0$ . Vrednost elementa  $m_{p,q}^\eta$  jednaka je 1 ukoliko se greška iz čvora  $v_i$  može preneti putevima koji postoje u grafu do izlaza  $y^\eta$ , koji odgovara elementu  $m_{p,q}^\eta$ . Vrednost 0 element ima ukoliko greška nastala u čvoru ne utiče na posmatrani izlaz.

**Primer** Po definiciji 4.7, uz funkciju uređenja prikazanu na slici 4.4, mapa uticaja greške  $M_0$ , koja odgovara izlaznom čvoru  $y^0$  je

$$M_0 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \triangle$$

Imajući prethodno u vidu, parcijalnu otpornost na defekte možemo definisati na sledeći način:

**Definicija 4.8 (Parcijalna otpornost na defekte [14])** *Parcijalnu otpornosti na defekte (PDT) arhitekture definišemo kao funkciju iz skupa  $\{0, 1, \dots, L-1\}$  u podskup  $V$ , tako da*

$$\mathbf{P}_{DT}(\alpha) = \bigcup_{\eta=L-\alpha}^{L-1} \mathbf{M}_\eta, \quad (4.3)$$

gde je  $\alpha$  mera parcijalne otpornosti na greške, za koju važi  $\Delta_H = 0$  ili  $\Delta_E \leq 2^{L-\alpha} - 1$ .

**Lema 4.3** *Skup  $\mathbf{P}_{DT}(\alpha)$  određuje netolerantnu oblast arhitekture.*

**Dokaz** Iz definicija 4.4 i 4.8 direktno sledi tvrđenje leme.  $\square$

Po definiciji 4.8, parametar  $\alpha$  predstavlja broj izlaznih čvorova arhitekture koji se karakterišu kao visokopouzdana, i čiji se skupovi značajnih čvorova ujedinjuju u jedinstveni skup  $\mathbf{P}_{DT}$ . Ovaj skup predstavlja skup značajnih čvorova grafa, odnosno arhitekture u celini. Očigledno, za  $\alpha = 0$  unija iz def. 4.8 je prazan skup, pa se u ovom slučaju metodi visoke pouzdanosti ne primenjuju ni nad jednim čvorom arhitekture. Tako, parcijalno visokopouzdana arhitektura  $\mathbf{P}_{DT}(0)$  jednaka je osnovnoj arhitekturi. U skladu sa definicijom 4.8,  $\mathbf{P}_{DT}(L)$  je arhitektura kod koje su svim čvorovi projektovani kao visokopouzdana.

**Lema 4.4** Skup čvorova  $v_i$  za koje važi  $v_i \in \mathbf{P}_{DT}(\alpha)$ , ukoliko su poznate mape uticaja  $M_{L-\alpha}, M_{L-\alpha+1}, M_{L-1}$ , moguće je odrediti kao

$$M = \bigvee_{\eta=L-\alpha}^{L-1} \mathbf{M}_\eta. \quad (4.4)$$

**Dokaz.** Na osnovu definicije 4.7, vrednost elementa  $m_{p,q}^\eta$  je jednaka 1 ukoliko čvor  $v_i$  preslikan funkcijom  $f_0$  na poziciju  $(p, q)$  u skupu  $\mathbf{M}_\eta$ . Vrednost elementa  $m_{p,q}$  u matrici  $M$  iz izraza (4.4) biće 1 ukoliko bar jedan od elemenata

$$m_{p,q}^\eta, \quad \eta = L - \alpha, L - \alpha + 1, \dots, L - 1$$

ima vrednost 1. Ovo znači da će važiti  $m_{p,q} = 1$  samo ukoliko je čvor  $v_i$  bar u jednom od skupova

$$\mathbf{M}_\eta, \quad \eta = L - \alpha, L - \alpha + 1, \dots, L - 1,$$

što odgovara uniji skupova iz izraza (4.3).  $\square$

Na osnovu prethodno izloženog, čvorove koje imaju uticaj na posmatrani skup izlaznih čvorova grafa moguće je odrediti iz mapa uticaja greške pojedinih izlaznih čvorova. Imajući u vidu da je propagacija greške tranzitivna relacija (lema 4.2), mape uticaja greške svih izlaznih čvorova mogu se odrediti iz tranzitivnog zatvaranja grafa, koje daje informaciju o svim putevima u grafu. U narednom poglavlju data je matematička osnova tranzitivnog zatvaranja, a operacija tranzitivnog zatvaranja u odnosu na Hamingovo rastojanje ilustrovana je na primeru *bit-plane* filtra datog u poglavlju 2.3.4.

### 4.3 Tranzitivna zatvaranja

Tranzitivno zatvaranje je operacija koja daje informaciju o svim mogućim putevima u grafu. Kao takva, ova operacija je pogodna za određivanje mapa uticaja greške. U cilju ilustracije matematičkog aparata za određivanje mapa uticaja greške arhitekture (def. 4.7), u ovom poglavlju date su osnovne definicije relacija i njihovog tranzitivnog zatvaranja [92, 93]. U prvom delu poglavlja data je formalna definicija tranzitivnog zatvaranja grafa i algoritam za određivanje tranzitivnog zatvaranja, dok je u drugom delu tranzitivno zatvaranje grafa ilustrovano na primeru arhitekture *bit-plane* FIR filtra.

### 4.3.1 Definicija tranzitivnog zatvaranja

**Definicija 4.9 (Relacija)** *Relacija  $R$  na konačnom skupu  $S$  je podskup Dekartovog proizvoda  $R \subset S \times S = \{(x, y) | x, y \in S\}$  [93].*

**Definicija 4.10 (Reprezentacija relacija)** *Digraf (usmereni graf) je reprezentacija relacije  $R$  na konačnom skupu  $S$  čiji čvorovi odgovaraju elementima skupa  $S$  i čije grane odgovaraju uređenim parovima u relaciji [93].*

Po definiciji 4.10 grana od čvora  $x$  do čvora  $y$  postoji samo ukoliko važi  $(x, y) \in R$ . Posledica ovoga je i obrnuto tvrđenje. Naime,  $(x, y) \in R$  samo ukoliko u digrafu  $G$  postoji odgovarajuća grana. Ukoliko je relacija  $R$  tranzitivna relacija, tada se digraf  $G$  naziva tranzitivni digraf [93].

**Definicija 4.11 (Tranzitivno zatvaranje)** *Tranzitivno zatvaranje  $R^*$  binarne relacije  $R$  je relacija  $R^*$  definisana kao*

$$(x, y) \in R^*$$

*ako, i samo ako postoji sekvenca*

$$x = v_0, v_1, v_2, \dots, v_k = y$$

*takva da je  $k \geq 1$  i  $(v_i, v_{i+1}) \in R$ , za  $i = 0, 1, \dots, k - 1$ . Ekvivalentno, tranzitivno zatvaranje  $R^*$  relacije  $R$  je najmanja tranzitivna relacija koja sadrži  $R$  [93].*

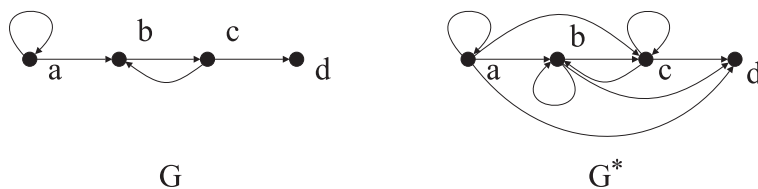
**Definicija 4.12 (Tranzitivno zatvaranje grafa)** *Neka je  $G$  digraf reprezentacija relacije  $R$ . Tada digraf  $G^*$ , koji predstavlja tranzitivno zatvaranje  $R^*$  relacije  $R$ , naziva se tranzitivno zatvaranje digrafa  $G$  [93].*

Po definiciji 4.12, grana  $e = (x, y)$ ,  $x \neq y$  pripada grafu  $G^*$  akko u grafu  $G$  postoji usmereni put koji vodi od čvora  $x$  do čvora  $y$ .

**Primer** Pretpostavimo da je relacija  $R$  na skupu  $S = \{a, b, c, d\}$  data sa

$$\{(a, a), (a, b), (b, c), (c, b), (c, d)\}$$

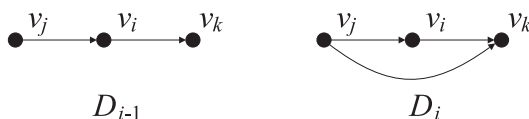
Tada, digraf reprezentacija  $G$ , relacije  $R$ , i tranzitivno zatvaranje  $G^*$  imaju oblik prikazan na slici 4.5 [93]. △

Slika 4.5: Digraf  $G$  i njegovo tranzitivno zatvaranje  $G^*$ 

### 4.3.2 Konstrukcija tranzitivnog zatvaranja digrafa: Warshallov algoritam

Neka digraf  $G$  ima  $\nu$  čvorova  $v_1, v_2, \dots, v_\nu$ . Algoritam efikasan u pogledu izračunavanja, po Warshallu [94], konstruiše sekvencu digrafa u  $n$  iteracija  $D_0, D_1, \dots, D_n$ , takvu da je  $D_0 = G$ , gde je digraf  $D_{i-1}$  podskup  $D_i, i = 1, 2, \dots, n$ , a  $D_n$  tranzitivno zatvaranje grafa  $G$ .

Digraf  $D_i$  dobija se iz digrafa  $D_{i-1}$  dodavanjem grane  $(v_j, v_k)$  digrafu  $D_{i-1}$  (ukoliko već nije u  $D_{i-1}$ ) uvek kada postoji usmereni put dužine 2 u  $D_{i-1}$  koji ide od čvora  $v_j$  do čvora  $v_k$ , preko čvora  $v_i$  (slika 4.6).

Slika 4.6: Dodavanje grane  $(v_j, v_k)$  digrafu  $D_{i-1}$  po Warshallovom algoritmu

Na slici 4.7 je u pseudo-kodu prikazan Warshallov algoritam za konstrukciju tranzitivnog zatvaranja digrafa  $D$ .

Graf se može predstaviti i u matricnom obliku. Matricna reprezentacija grafa, za razliku od grafičke reprezentacije, pogodnija je za manipulaciju i transformaciju. Matricnu reprezentaciju koristićemo u narednim poglavljima za određivanje tranzitivnog zatvaranja grafa toka podataka arhitekture.

Svakom grafu  $\mathbf{G} = (V, E)$ , gde je skup  $V = \{v_1, v_2, \dots, v_\nu\}$  konačni skup čvorova, a  $E = \{e_1, e_2, \dots, e_\varepsilon\}$  konačni skup grana, odgovara matrica dimenzija  $\nu \times \varepsilon$  koja se naziva matrica incidencije i matrica dimenzija  $\nu \times \nu$  koja se naziva matrica susedstva [92].

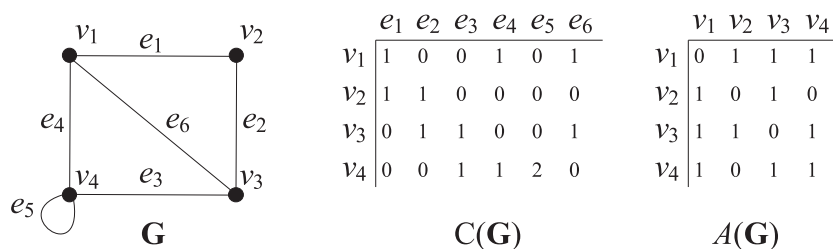
**Definicija 4.13 (Matrica incidencije)** *Matrica incidencije  $C(\mathbf{G}) = [c_{i,j}]$ , koja odgovara grafu  $\mathbf{G}$ , je matrica dimenzija  $\nu \times \varepsilon$ , čiji je element  $c_{i,j}$  jednak 1, ako je  $v_i \in e_j$ . Ukoliko važi  $e_j = (v_i, v_i)$ , odgovarajući element matrice incidencije je  $c_{i,j} = 2$ . [92].*



**Warshallov algoritam:****Ulaz:** digraf  $D$  sa čvorovima  $v_1, v_2, \dots, v_\nu$ .**Izlaz:** tranzitivno zatvaranje digraaf  $D$ .Inicijalizuj digraf  $D_0$  na  $G$ .**For**  $i = 1$  to  $\nu$     **For**  $j = 1$  to  $\nu$         **If**  $e = (v_j, v_i) \in D_{i-1}$             **For**  $k = 1$  to  $\nu$                 **If**  $e = (v_i, v_k) \in D_{i-1}$                     Dodaj granu  $e = (v_j, v_k)$  digrafu  $D_{i-1}$ .**Return** digraf  $D_n$ .Slika 4.7: Warshallov algoritam za konstrukciju tranzitivnog zatvaranja digrafa  $D$ 

**Definicija 4.14 (Matrica susedstva)** *Matrica susedstva*  $A(\mathbf{G}) = [a_{i,j}]$ , koja odgovara grafu  $\mathbf{G}$ , je matrica dimenzija  $\nu \times \nu$ , čiji je element  $a_{i,j}$  jednak 1 ukoliko postoji grana od čvora  $v_i$  do čvora  $v_j$ , i 0 u suprotnom [92].

U slučaju grafa čijim su granama dodeljene težine, element matrice susedstva jednak je težini odgovarajuće grane. Na slici 4.8 dat je primer grafa  $\mathbf{G}$  sa odgovarajućim matricnim reprezentacijama  $C(\mathbf{G})$  i  $A(\mathbf{G})$ .



Slika 4.8: Primer matricne reprezentacije grafa: a) grafička reprezentacija, b) odgovarajuća reprezentacija matricom incidencije, c) odgovarajuća reprezentacija matricom susedstva

Jedan od načina kako je moguće odrediti sve najkraće puteve u grafu korišćenjem matrice susedstva grafa je Floyd-Warhallov algoritam [95]. Ovaj algoritam određuje matricu težina najkraćih puteva svih parova čvorova u grafu  $\mathbf{G} = (V, E)$ .

**Definicija 4.15 (Matrica najkraćih puteva)** *Matricu*  $S(\mathbf{G}) = [s_{i,j}]$ , čiji

je element  $s_{i,j}$  jednak dužini najkraćeg puta između čvorova  $v_i$  i  $v_j$ , nazivamo matricom najkraćih puteva grafa  $\mathbf{G}$ .

Floyd-Warshallov algoritam prikazan je na slici 4.9. Da bi Floyd-Warshallov algoritam mogao da odredi najkraće puteve potrebno je matricu susedstva  $A = [a_{i,j}]$  formirati tako da se nepostojanje grane od čvora  $v_i$  do čvora  $v_j$  predstavi sa  $a_{i,j} = \infty$ .

**Floyd-Warshallov algoritam:**

**Ulaz:** Matrica susedstva  $A(\mathbf{G})$  digrafa  $\mathbf{G}$  sa čvorovima  $V = \{v_1, v_2, \dots, v_\nu\}$ .

**Izlaz:** Matrica cena svih najkraćih puteva u grafu  $S(\mathbf{G})$ .

```

For  $i = 1$  to  $\nu$ 
  For  $j = 1$  to  $\nu$ 
     $S[i, j] = A[i, j]$ 
For  $k = 1$  to  $\nu$ 
  For  $i = 1$  to  $\nu$ 
    For  $j = 1$  to  $\nu$ 
      If  $S[i, k] + S[k, j] < S[i, j]$ 
         $S[i, j] = S[i, k] + S[k, j]$ 

```

Slika 4.9: Floyd-Warshallov algoritam

**Primer** Uzmimo primer grafa  $\mathbf{G}$  sa slike 4.8, čija je matrica susedstva  $A(\mathbf{G})$  prikazana na istoj slici. Matrica susedstva data je sa

$$A(\mathbf{G}) = \begin{bmatrix} \infty & 1 & 1 & 1 \\ 1 & \infty & 1 & \infty \\ 1 & 1 & \infty & 1 \\ 1 & \infty & 1 & 1 \end{bmatrix}.$$

Izvršenjem Floyd-Warshallovog algoritma sa slike 4.9 nad matricom susedstva dobija se matrica

$$S(\mathbf{G}) = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix}.$$

Element  $s_{2,4} = 2$  označava da postoji bar jedan put između čvorova  $v_2$  i  $v_3$ , minimalne dužine 2. Ovo se može proveriti na slici 4.8, gde postoje tačno dva puta između pomenutih čvorova: put  $v_2 \rightarrow v_1 \rightarrow v_4$ , i put  $v_2 \rightarrow v_3 \rightarrow v_4$ .

△

Problem određivanja tranzitivnog zatvaranja  $\mathbf{G}^*$  grafa  $\mathbf{G} = (V, E)$  u matricnoj reprezentaciji svodi se na problem određivanja postojanja puta od čvora  $v_i$  do čvora  $v_j$  u grafu  $\mathbf{G}$ . Neka je matrica susedstva  $A(\mathbf{G}) = [a_{i,j}]$ , i neka je matrica  $A^* = [a_{i,j}^*]$  tranzitivno zatvaranje, čiji je element  $a_{i,j}^*$  jednak 1 ukoliko postoji put od čvorova  $v_i$  do čvora  $v_j$ , i 0 obrnuto. Modifikacija Warshallovog algoritma sa slike 4.7 za određivanje tranzitivnog zatvaranja grafa predstavljenog matricom susedstva  $A$  prikazan je na slici 4.10 [95].

**Tranzitivno zatvaranje:**

**Ulaz:** Matrica susedstva  $A(\mathbf{G})$  digrafa  $\mathbf{G}$  sa čvorovima  $V = \{v_1, v_2, \dots, v_\nu\}$ .

**Izlaz:** Tranzitivno zatvaranje  $A^*(\mathbf{G})$ .

**For**  $i = 1$  to  $\nu$

**For**  $j = 1$  to  $\nu$

$A^*[i, j] = A[i, j]$

**For**  $i = 1$  to  $\nu$

**For**  $j = 1$  to  $\nu$

**For**  $k = 1$  to  $\nu$

**If**  $A^*[i, k] = 1$

$A^*[i, j] = A^*[i, k] \wedge A^*[k, j]$ .

Slika 4.10: Warshallov algoritam za određivanje tranzitivnog zatvaranja na osnovu matrice susedstva

Može se pokazati da važi

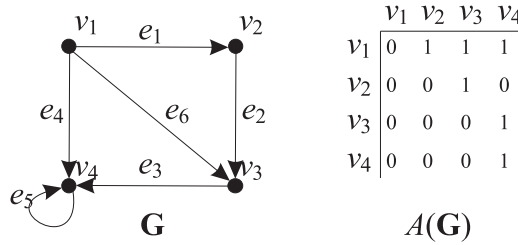
$$A(\mathbf{G}^*) = A^*(\mathbf{G}),$$

odnosno, da je matrica susedstva grafa  $\mathbf{G}^*$  jednaka matrici  $A^*$  koja se dobija izvršenjem algoritma sa slike 4.10 za osnovni graf  $\mathbf{G}$  [95].

**Primer** Uzmimo primer usmerenog grafa  $\mathbf{G}$  sa slike 4.11, čija je matrica susedstva  $A(\mathbf{G})$  prikazana na istoj slici. Izvršenjem algoritma sa slike 4.10 nad matricom susedstva dobija se matrica

$$A^*(\mathbf{G}) = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.5)$$

Matrica (4.5), za razliku od matrice susedstva, ima vrednost 1 na mestu elementa  $a_{2,4}^*$ . Ovo znači da bez obzira što ne postoji direktni put između čvorova  $v_2$  i  $v_4$  postoji/postoje čvorovi preko kojih je iz čvora  $v_2$  moguće doći u čvor  $v_4$ , što se može potvrditi iz digrafa sa slike 4.11.  $\triangle$



Slika 4.11: Primer tranzitivnog zatvaranja: digraf  $\mathbf{G}$  sa odgovarajućom matricom susedstva  $A(\mathbf{G})$

Drugi način određivanja tranzitivnog zatvaranja izvodi se iz tvrđenja sledeće leme.

**Lema 4.5** *Matrica  $A^{(k)} = [a_{i,j}^{(k)}]$ , kod koje je vrednost elementa  $a_{i,j}^{(k)}$  jednaka broju puteva dužine  $k$  grafa  $\mathbf{G}$ , jednaka je*

$$A^{(k)}(\mathbf{G}) = A(\mathbf{G})^k,$$

gde je sa  $A^k$  označen  $k$ -ti stepen matrice  $A$  [96, str. 229].

**Dokaz** Dokaz je dat u [96].  $\square$

Neka je  $T$  dužina najdužeg puta u grafu.

**Lema 4.6** *Elemente matrice tranzitivnog zatvaranja moguće je odrediti iz matrice*

$$B = \sum_{i=1}^T A^i(\mathbf{G}), \quad (4.6)$$

kao

$$a_{i,j}^* = \begin{cases} 1, & b_{i,j} > 0 \\ 0, & \text{inače} \end{cases}. \quad (4.7)$$

**Dokaz** Dokaz leme sledi iz tvrđenja leme 4.5.  $\square$

### 4.3.3 Tranzitivno zatvaranje *bit-plane* FIR filtra

U ovom poglavlju dat je primer određivanja tranzitivnog zatvaranja semi-sistoličkog polja *bit-plane* FIR filtra prikazanog u poglavlju 2.3.4 (slika 2.25), u odnosu na Hamingovu metriku otkaza (def. 4.3).

Na slici 2.25 prikazan je funkcionalni blok dijagram semi-sistoličkog *bit-plane* polja za FIR filtriranje sa  $k = 3$  koeficijenata, dužine  $m = 4$  bita i širinom ulaznih podataka  $n = 5$ , koji izvršava algoritam (2.13).

U ovom primeru, radi jednostavnosti, nećemo razmatrati greške nastale na linijama za paralelno uvođenje ulaznih reči  $\{x_i\}$  u arhitekturu, kao ni emisione linije za uvođenje bitova koeficijenata  $c_i^j$  (slika 2.25). Kao potencijalni uzrok greške uzećemo kombinacionu mrežu ćelija filtra, a propagaciju greške i otkaz sistema razmatraćemo u odnosu na Hamingovu metriku (def. 4.3). Tako, greška nastala u ćeliji filtra može biti na izlazu ćelije  $s$  (suma), ili na prenosu  $c$  (slika 2.25). Bez obzira na tip greške, sve greške ćemo uzeti sa istom težinom, odnosno, greškom ćemo smatrati svako odstupanje od tačnog rezultata (def. 4.3). Težina greške će biti razmatrana u glavi 6.

Imajući u vidu prethodno navedena ograničenja, na osnovu polja sa slike 2.25, formiran je graf  $\mathbf{G}$  i prikazan na slici 4.12. Grane grafa pokazuju moguće puteve propagacije greške (def. 4.3 i 4.5). Graf u primeru sa slike 4.12 sadrži  $m \cdot k = 12$  vrsta, a na osnovu izraza (2.14) broj kolona je  $l_0 = 9$ .

Veze između ćelija u okviru jednog BP elementa su identične, a na prelazu od jednog BP elementa do drugog (svaka  $k$ -ta vrsta) vrši se pomeranje za jednu bit poziciju udesno, tako da su grane koje odgovaraju tim ćelijama polja usmerene u desno [14, 97]. Da bi se izbegla neregularnost veza između ćelija BP FIR filtra i omogućilo jednostavnije nalaženje opšteg oblika tranzitivnog zatvaranja grafa sa slike 4.12, potrebno je fiktivno "pomeriti" sve ćelije sledećeg BP elementa za jedno mesto ulevo u odnosu na prethodni i zadržati sve veze između ćelija. Kako arhitekturu čine  $m$  BP elemenata (slika 2.24), ukupan broj kolona proširenog grafa je

$$L = m + l_0. \quad (4.8)$$

Prošireni graf *bit-plane* FIR filtra prikazan je na slici 4.13 [14]. Čvorovi grafa sa slike 4.13 povezani su sa susednim čvorovima na identičan način. Neosenčanim krugovima su prikazani fiktivni čvorovi, dodati u cilju postizanja regularnosti grafa. Ovi čvorovi ne vrše izračunavanja, već samo prosleđuju ulazne podatke. Predloženom transformacijom dobijen je graf sa strogo regularnom topologijom tipa  $4^4$  (poglavlje 2.3.2, def. 2.14).

**Teorema 4.1** *Opšti oblik matrice tranzitivnog zatvaranja grafa sa slike 4.13 je*

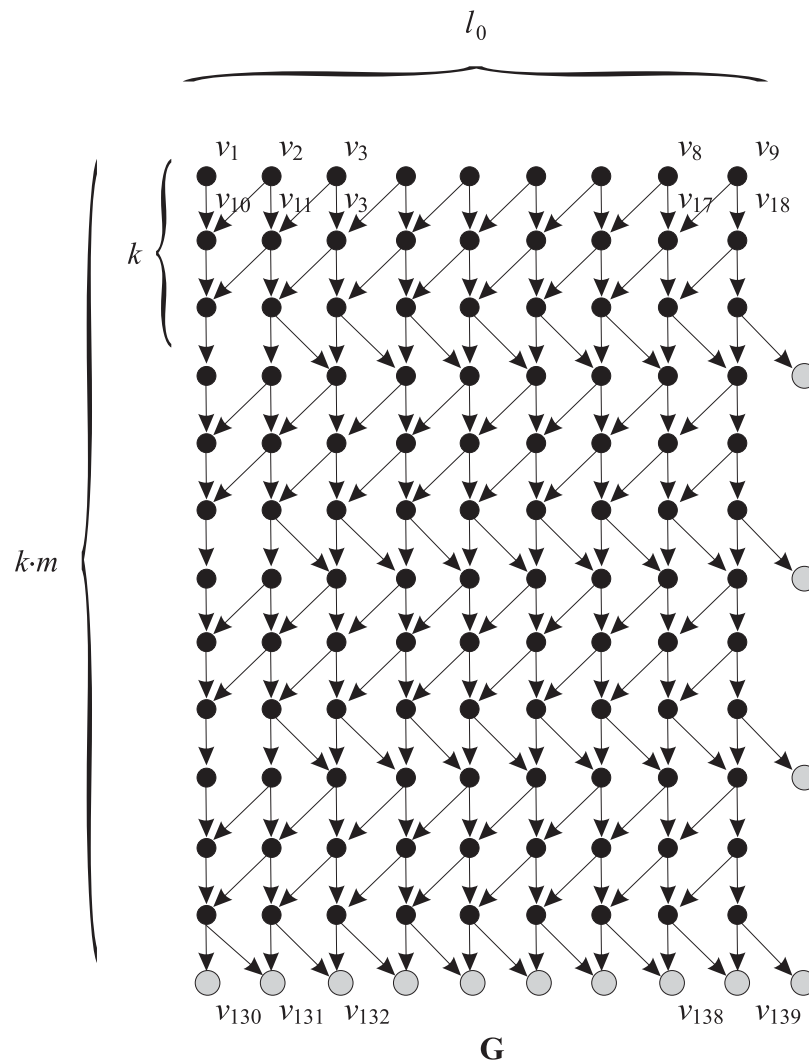
$$A^*(\mathbf{G}) = \begin{bmatrix} 0 & A_C & A_C^2 & \dots & A_C^{m-k} \\ 0 & 0 & A_C & \dots & A_C^{m-k-1} \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & A_C \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{(m-k+1) \cdot L \times (m-k+1) \cdot L}, \quad (4.9)$$

gde je  $A_C^n = [a_{i,j}^C]^n$   $n$ -ti stepen matrice susjedstva kolona  $A_C$ , čiji su elementi

$$(a_{i,j}^C)^n = \begin{cases} 1, & j + n \geq i \geq j \\ 0, & \text{inače} \end{cases} \quad (4.10)$$

(Ćirić et al, 2009, [14]).

**Dokaz.** Sa slike 4.13 može se utvrditi zavisnost izmedju čvorova. Čvor  $v_1$  u prvoj vrsti prve kolone sa slike 4.13 povezan je samo sa čvorom koji se nalazi



Slika 4.12: Graf  $G$  toka podataka *bit-plane* FIR filtra sa  $k = 3$  koeficijenata, dužine  $m = 4$  bita i širinom ulaznih podataka  $n = 5$

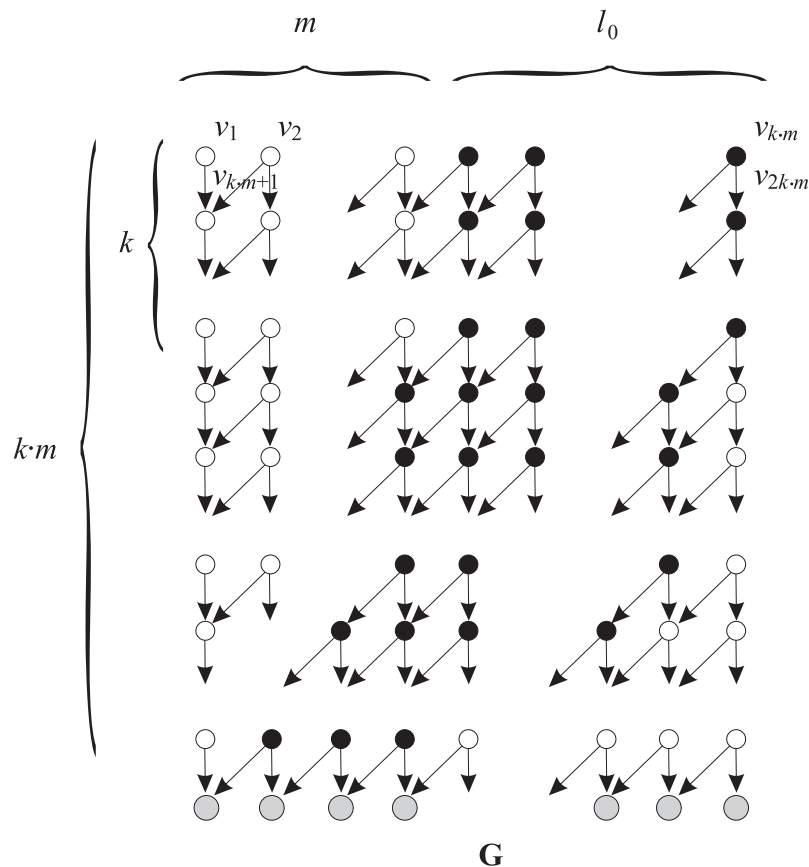
u drugoj vrsti i prvoj koloni. Čvor  $v_2$  iz prve vrste i druge kolone povezan je sa ćelijama iz prve i druge kolone druge vrste i tako redom.

Ako se svaka kolona predstavi sa po jednim čvorom novog grafa  $\mathbf{G}_C$ , dobija se graf prikazan na slici 4.14. Svaki čvor je povezan sa samim sobom i sa prethodnim čvorom, izuzev prvog koji nema prethodnika.

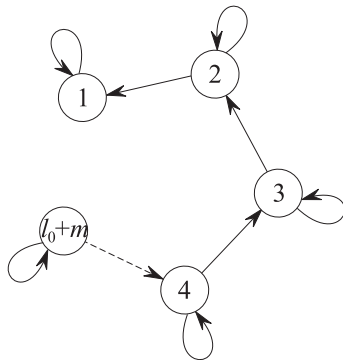
Graf  $\mathbf{G}_C$  sa slike 4.14 moguće je predstaviti matricom susedstva

$$A_C = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & \dots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}_{L \times L} . \quad (4.11)$$

Matrica (4.11) je dimenzije  $L \times L$ , gde je  $L$  broj čvorova grafa  $\mathbf{G}_C$  sa slike 4.14, odnosno broj kolona grafa  $\mathbf{G}$  sa slike 4.13 datog izrazom (4.8). Iz (4.11) može



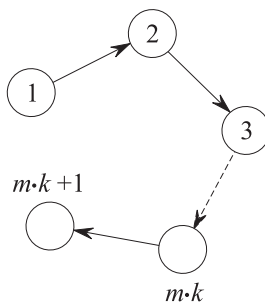
Slika 4.13: Modifikovani graf toka podataka *bit-plane* FIR filtra

Slika 4.14: Graf povezanosti kolona *bit-plane* FIR filtra

se uočiti da samo elementi na glavnoj dijagonali i dijagonali ispod nje imaju vrednost 1. Vrednost elemenata matrice  $A_C = [a_{i,j}^C]$  je

$$a_{i,j}^C = \begin{cases} 1, & j+1 \geq i \geq j \\ 0, & \text{inače} \end{cases} . \quad (4.12)$$

Predstavimo čvorovima novog grafa  $\mathbf{G}_R = [g_{i,j}^R]$  vrste čvorova grafa sa slike 4.13. Graf sa slike 4.13 ima  $m \cdot k + 1$  vrstu, koje su povezane međusobno tako da je  $i$ -ta vrsta povezana samo sa vrstom  $i+1$ . Graf  $\mathbf{G}_R$  prikazan je na slici 4.15. Izuzetak je poslednja vrsta koja nije povezana dalje ni sa jednom

Slika 4.15: Graf povezanosti vrsta *bit-plane* FIR filtra

drugom vrstom. Matrica susedstva imaće elemente različite od nule samo na prvoj dijagonali iznad glavne dijagonale. Neophodno je imati u vidu da u ovom slučaju elementi nisu jedinice, jer postoji i povezanost čvorova po kolonama, već su elementi matrice susedstva  $A_C$ . Matrica susedstva, kojom



se može opisati graf  $\mathbf{G}_R$  sa slike 4.15, ima oblik

$$A_R = \begin{bmatrix} 0 & A_C & 0 & \dots & 0 \\ 0 & 0 & A_C & \dots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & A_C \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{(m \cdot k + 1) \times (m \cdot k + 1)}, \quad (4.13)$$

što je ujedno i matrica susedstva  $A$  proširenog grafa arhitekture *bit-plane* FIR filtra sa slike 4.13, odnosno važi

$$A(\mathbf{G}) = A_R.$$

Matrica (4.13) je kvadratna matrica dimenzije  $(m \cdot k + 1) \cdot L$ . Po definiciji 4.14, ova matrica opisuje sve puteve dužine 1 u grafu sa slike 4.13. Sve puteve dužine dva, na osnovu leme 4.5, dobijamo kvadriranjem matrice susedstva. Matrica  $A^2$ , koja sadži sve puteve dužine 2, je

$$A^2 = \begin{bmatrix} 0 & A_C & A_C^2 & 0 & \dots & 0 & 0 \\ 0 & 0 & A_C & A_C^2 & \dots & 0 & 0 \\ & & & & \vdots & & \\ 0 & 0 & 0 & 0 & \dots & A_C & A_C^2 \\ 0 & 0 & 0 & 0 & \dots & 0 & A_C \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}_{(m \cdot k + 1) \times (m \cdot k + 1)}.$$

Daljim stepenovanjem dobija se

$$A^{k \cdot m} = \begin{bmatrix} 0 & A_C & A_C^2 & \dots & A_C^{m \cdot k} \\ 0 & 0 & A_C & \dots & A_C^{m \cdot k - 1} \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & A_C \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}.$$

Na osnovu tvrđenja leme 4.6, tranzitivno zatvaranje  $A^*$  može se odrediti iz matrice  $B$  date sa (4.6). Na osnovu oblika stepena matrice susedstva  $A^i$  važi  $A^* = A^{km}$ , što dokazuje tvrđenje teoreme.

Da elementi matrice  $A_C^n = [a_{i,j}^C]^n$  imaju oblik dat jednačinom (4.10) pokazaćemo matematičkom indukcijom. Za  $n = 1$  izraz (4.10) postaje

$$(a_{i,j}^C) = \begin{cases} 1, & j + 1 \geq i \geq j \\ 0, & \text{inače} \end{cases},$$

što je identično izrazu (4.12).

Pretpostavimo da važi tvrđenje leme da je

$$(a_{i,j}^C)^n = \begin{cases} 1, & j + n \geq i \geq j \\ 0, & \text{inače} \end{cases} .$$

Tada, za elemente  $(a_{i,j}^C)^{n+1}$  važi

$$\begin{aligned} A_C^{n+1} = A_C^n \times A_C &= \sum_{p=0}^{k \cdot m + 1} (a_{i,p}^C)^n \cdot a_{p,j}^C = \\ &= \begin{cases} 1, & p + n \geq i \geq p \wedge j + 1 \geq p \geq j \\ 0, & \text{inače} \end{cases} = \\ &= \begin{cases} 1, & j + n + 1 \geq i \geq j \\ 0, & \text{inače} \end{cases} , \end{aligned}$$

čime je dokazana jednakost (4.10). □

## 4.4 Primer projektovanja parcijalno visokopouzdanog bit-plane FIR filtra

Opšti oblik matrice tranzitivnog zatvaranja arhitekture *bit-plane* FIR filtra, koja je potrebna za određivanje mapa uticaja grešaka *bit-plane* arhitekture dat je jednačinom (4.9) u teoremi 4.1. Za slučaj polja sa slike 2.25 dimenzije matrice  $A^*$  su 195x195. Zbog veličine problema, u ovom poglavlju biće razmatran primer *bit-plane* polja dimenzija  $k = 2$ ,  $m = 2$  i  $l_0 = 4$ . Semi-sistoličko polje *bit-plane* FIR filtra, za zadate parametre, prikazano je na slici 4.16(a).

Dijagram toka algoritma za projektovanje PDT sistema dat je na slici 4.17.

### 1. i 2. korak – izbor PDT parametara i formiranje grafa G

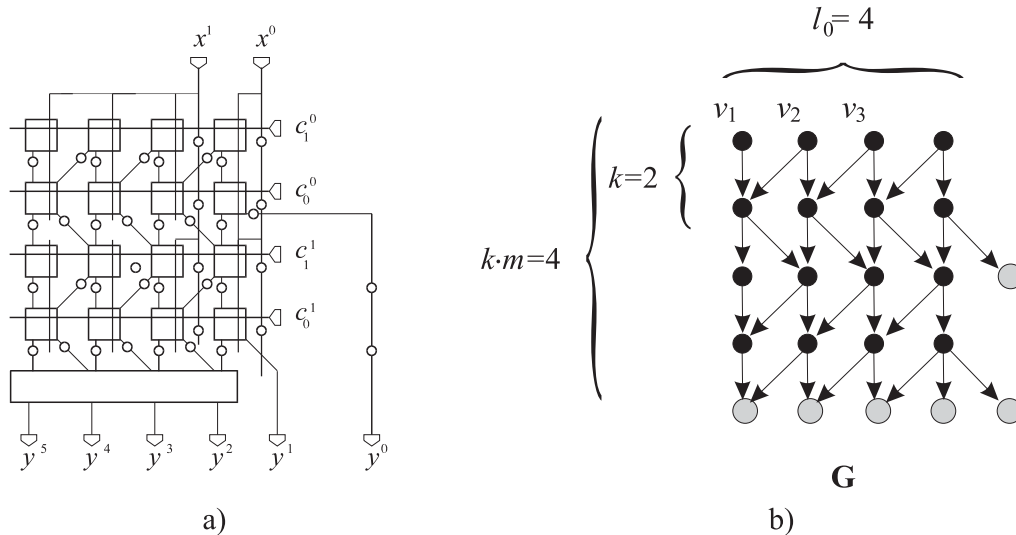
Neka je minimalna jedinica zamene jedna ćelija arhitekture sa slike 2.25. Radi jednostavnosti neće biti razmatrane greške nastale na linijama za paralelno uvođenje ulaznih reči  $\{x_i\}$  u arhitekturu, kao ni na emisionim linijama za uviđenje bitova koeficijenata  $c_i^j$  (slika 4.16a). Kao potencijalni uzrok greške uzeta je jedino kombinaciona mreža ćelija filtra. Tako, greška nastala u ćeliji filtra može biti na izlazu ćelije  $s$  (suma), ili na prenosu  $c$  (slika 4.16a). Za metriku otkaza u ovom primeru izabrana je Hamingova metrika (def. 4.3). U ovom primeru greškom smatramo svako odstupanje od tačnog rezultata, bez

uzimanja u obzir težine greške (def. 4.3). Po ovome, graf  $\mathbf{G}$  koji odgovara DFG-u sa slike 4.16a, za izabranu metriku, ima topologiju  $4^4$  i prikazan je na slici 4.16(b). Modifikacijom grafa  $\mathbf{G}$  sa slike 4.16b u cilju postizanja regularnosti na način predložen na slici 4.13, dobija se graf sa slike 4.18.

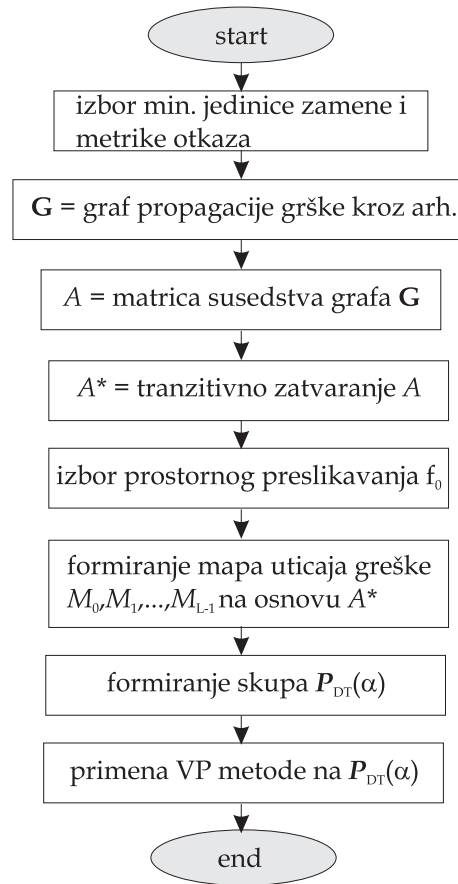
### 3. i 4. korak – matrica susjedstva i tranzitivno zatvaranje

Po definiciji 4.14 i na osnovu izraza (4.11), matrica susjedstva kolona  $A_C(\mathbf{G})$ , grafa  $\mathbf{G}$  sa slike 4.18 je

$$A_C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \quad (4.14)$$



Slika 4.16: Osnovno polje *bit-plane* FIR filtra za projektovanje PDT *bit-plane* polja: a) DFG *bit-plane* FIR filtra za  $k = 2$ ,  $m = 2$  i  $l_0 = 4$ ; b) graf uticaja grešaka  $\mathbf{G}$ .



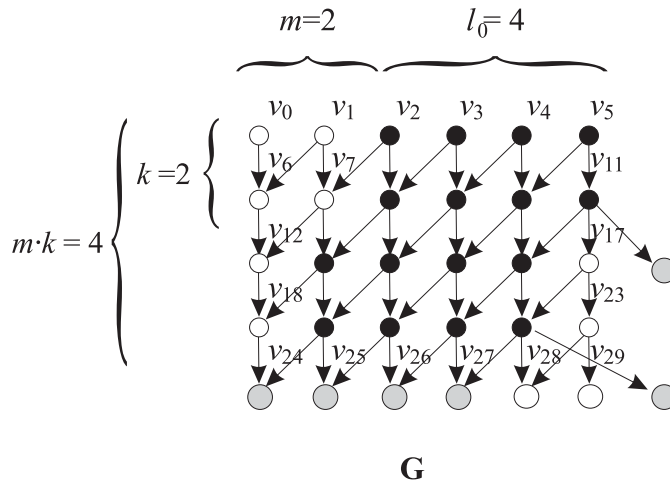
Slika 4.17: Dijagram toka projektovanja PDT sistema

Tranzitivno zatvaranje, na osnovu izraza (4.9), je

$$A(\mathbf{G})^* = \begin{bmatrix} 0 & A_C & A_C^2 & A_C^3 & A_C^4 \\ 0 & 0 & A_C & A_C^2 & A_C^3 \\ 0 & 0 & 0 & A_C & A_C^2 \\ 0 & 0 & 0 & 0 & A_C \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4.15)$$

Ukupan broj kolona proširenog grafa sa slike 4.18 je  $L = m + l_0 = 6$ . Matrica tranzitivnog zatvaranja u razvijenoj formi može se dobiti zamenom (4.14) u (4.15). Tranzitivno zatvaranje grafa sa slike 4.18 prikazano je na slici 4.19 [14]. Osenčanim poljima na slici 4.19 prikazani su delovi matrice oblika  $(A_C)^n$  iz izraza (4.15). Ostali elementi matrice su po (4.15) jednaki 0.

Na osnovu matrice sa slike 4.19 moguće je odrediti koji čvorovi grafa mogu propagirati greške do kojih bitova rezultata. Od značaja u matrici su samo



Slika 4.18: Modifikovani graf propagacije greške *bit-plane* FIR filtra sa slike 4.16 za  $k = 2$ ,  $m = 2$  i  $l_0 = 4$

poslednjih  $L$  kolona, kolone 24 – 29 u primeru sa slike 4.19, koje prikazuju postojanje uticaja svih čvorova grafa na izlazne čvorove  $v_{24}, v_{25}, \dots, v_{29}$  (slika 4.18). S obzirom na strukturu BP (slika 4.18), uticaji na izlaze arhitekture  $\{y_i\}$  opisani su kolonama matrice  $A^*$

$$(m \cdot k + 1) \cdot L - 1 - i,$$

gde je  $i$  indeks posmatranog izlaza. Ove kolone ukazuju na čvorove koji utiču na izlazne bitove rezultata, i to tako da kolona 24 ima jedinice tamo gde postoje putevi od odgovarajućeg čvora do bita  $y_5$ , 25 do bita  $y_4$  itd. Dakle, povećanje pouzdanosti bita rezultata  $y_5$ , moguće je postići povećanjem pouzdanosti čvorova na koje ukazuju jedinice u 24. koloni. Informacija o vrsti u kojoj se nalazi posmatrani element ukazuje na traženi čvor.

## 5. korak – prostorno uređenje

Funkcija prostornog uređenja čvorova grafa  $f_o$ , kako je opisano u poglavlju 4.2, uređuje čvorove  $v_i$  tako da ih je moguće mapirati na elemente matrice, čime se dobija mapa uticaja greške (definicija 4.7). Uzmimo za graf *bit-plane* arhitekture sa slike 4.18 funkciju uređenja

$$f_o : V \rightarrow \mathbf{N}^2, \quad (p, q) = f_o(v_i),$$

gde su  $p$  i  $q$  indeksi kolone, odnosno vrste matrice mape uticaja greške, takvu da je

$$p = \left\lfloor \frac{i_v}{L} \right\rfloor, \quad q = i_v \bmod L,$$

gde je  $i_v = 0, 1, \dots, (m \cdot k + 1) \cdot L - 1$ , indeks vrste matrice tranzitivnog zatvaranja (slika 4.19). Potrebno je napomenuti da se mapa uticaja formira na osnovu elemenata jedne kolone matrice tranzitivnog zatvaranja ( $i_k = c^{ta}$ ), tako da indeks kolone ne figuriše u izrazu (6.42).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0
2	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0
3	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	1	1	1	0	0
4	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	1	0
5	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	1	1	1	1	0	1	1	1	1
6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Slika 4.19: Tranzitivno zatvaranje bit-plane FIR filtra za  $k = 2$ ,  $m = 2$ ,  $l_0 = 4$  i  $L = 6$

## 6. korak – mape uticaja greške

Na osnovu tranzitivnog zatvaranja  $A^*$ , datog na slici 4.19, i funkcije uređenja (6.42), možemo odrediti sve mape uticaja greške *bit-plane* arhitekture. Postupak formiranja mape uticaja greške na izlazni čvor  $y_5$  ( $v_{24}$  na slici 4.18), funkcijom uređenja (6.42), prikazan na slici 4.20. Odgovarajuća mapa uticaja greške za primer sa slike 4.20 je

$$M_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Na sličan način možemo odrediti i mape uticaja greške na ostale izlazne bitove arhitekture. Mape uticaja greške  $M_4$ ,  $M_3$ ,  $M_2$ ,  $M_1$  i  $M_0$  su

	21	22	23	24	25	26	27	28	29	
0	0	0	0	1	0	0	0	0	0	
0	0	0	0	1	1	0	0	0	0	→ 1 1 1 1 1 0
0	0	0	0	1	1	1	0	0	0	
1	1	0	0	1	1	1	1	0	0	
1	1	1	0	1	1	1	1	1	0	
1	1	1	1	0	1	1	1	1	1	
0	0	0	0	1	0	0	0	0	0	1 1 1 1 0 0
0	0	0	0	1	1	0	0	0	0	
0	0	0	0	1	1	1	0	0	0	
1	1	0	0	1	1	1	1	0	0	
1	1	1	0	0	1	1	1	1	0	
1	1	1	1	0	0	1	1	1	1	
0	0	0	0	1	0	0	0	0	0	1 1 1 0 0 0
0	0	0	0	1	1	0	0	0	0	
0	0	0	0	1	1	1	0	0	0	
1	1	0	0	0	1	1	1	0	0	
1	1	1	0	0	0	1	1	1	0	
0	0	1	1	0	0	0	1	1	1	
0	0	0	0	1	0	0	0	0	0	1 1 0 0 0 0
0	0	0	0	1	1	0	0	0	0	
0	0	0	0	0	1	1	0	0	0	
0	0	0	0	0	0	1	1	0	0	
0	0	0	0	0	0	0	1	1	0	
0	0	0	0	0	0	0	0	1	1	
0	0	0	0	0	0	0	0	0	0	

Slika 4.20: Primer mapiranja elemenata tranzitivnog zatvaranja  $A^*$  na elemente mape uticaja greške  $M_\eta$  funkcijom uređenja  $f_o$

$$\begin{aligned}
 M_4 &= \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}, \\
 M_3 &= \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}, \\
 M_2 &= \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \\
 M_1 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \\
 M_0 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.
 \end{aligned}$$

**7. i 8. korak - formiranje skupa  $P_{DT}$  i primena metoda za postizanje visokepouzdanosti**

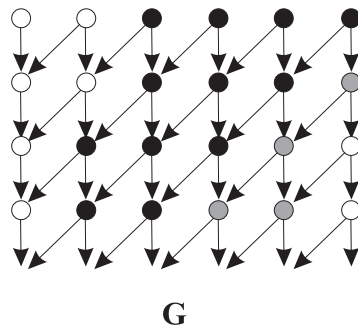
Na osnovu definicije 4.8 i postupka pokazanog lemom 4.4, ćelije arhitekture sa slike 4.16 koje je potrebno projektovati kao visokopuzdane, kako bi dobili arhitekturu parcijalno visokopouzdanog *bit-plane* FIR filtra za primer  $\alpha = 2$  su

$$M(\alpha = 2, L = 6) = \bigvee_{\eta=L-\alpha}^{L-1} M_{\eta} = M_4 \bigvee M_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (4.16)$$

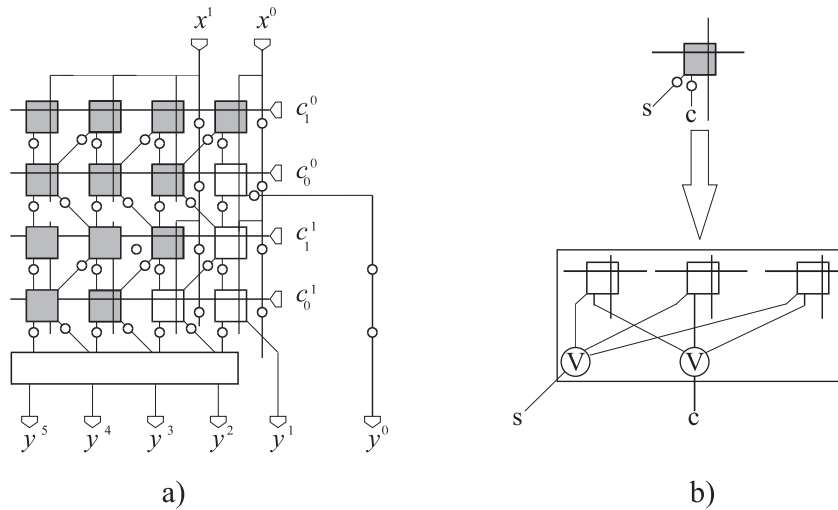
Graf parcijalno visokopouzdanе arhitekture opisane mapom uticaja greške (4.16) prikazan je na slici 4.21. Čvorovi na slici 4.21 koje je potrebno projektovati kao visokopouzdanе prikazani su crnom bojom. Čvorovi koji se ne menjaju su dati sa sivom ispunom, dok beli čvorovi predstavljaju ćelije dodate radi postizanja regularnosti grafa.

Arhitektura parcijalno visokopouzdanog *bit-plane* FIR filtra za  $k = 2$ ,  $m = 2$  i  $n = 2$  prikazana je na slici 4.22.





Slika 4.21: Graf parcijalno visokopouzdanog arhitekture *bit-plane* FIR filtra za  $k = 2$ ,  $m = 2$  i  $n = 2$



Slika 4.22: Arhitektura parcijalno visokopouzdanog *bit-plane* FIR filtra za  $k = 2$ ,  $m = 2$  i  $n = 2$  i  $\alpha = 2$ : a) arhitektura filtra, osenčane ćelije je potrebno projektovati sa povećanom pouzdanošću, b) struktura ćelije sa povećanom pouzdanošću projektovana TMR tehnikom

## 4.5 Implementacija parcijalno visokopouzdanog *bit-plane* FIR filtra

U cilju ilustracije kompromisa između ET i PDT metoda u pogledu zauzeća resursa, u ovom poglavlju dati su rezultati FPGA implementaciju ET i PDT BP FIR filtra [14].

Arhitektura prikazana na slici 2.25 je korišćenjem Xilinx ISE 8.2 WebPack

razvojnog okruženja opisana u VHDL-u i implementirana na Xilinx Spartan2 FPGA čipu [32]. Konstantama su u VHDL-u definisani parametri arhitekture ( $n, k, m$  i  $\alpha$ ), tako da je sintezu moguće izvršiti za proizvoljne dimenzije polja.

Parcijalna visoka pouzdanost je ugrađena u VHDL opis kroz implementaciju funkcije pridružene arhitekturi, koja u zavisnosti od zadatog parametra  $\alpha$  određuje i vraća mapu uticaja greške arhitekture. Funkcija je zasnovana na tvrđenju teoreme 4.1 i leme 4.4, koje opisuju postupak za određivanje tranzitivnog zatvaranja BP arhitekture i ćelija polja koje treba projektovati kao visokopouzdanost za poznate mape uticaja greške.

Formirana VHDL funkcija vraća matricu kod koje svaki element odgovara jednoj ćeliji polja. Ukoliko je vrednost elementa jednaka 1, na mestu te ćelije instancira se komponenta opisana u VHDL-u kao visokopouzdanost sistem (slika 4.22). Ukoliko je odgovarajući element matrice 0, na mesto ćelije vrši se instanciranje komponente osnovne ćelije filtra opisane takođe u VHDL-u. Funkcija je data u dodatku A.

U ovom poglavlju dati su rezultati implementacije tri *bit-plane* polja različitih dimenzija. Dimenzije implementiranih polja prikazani su u tabeli 4.1, a polja su nazvana redom Arr1, Arr2 i Arr3.

	$k$	$m$	$n$	$l_0$	$m \cdot k$	$\mathbf{P}_{DT}(\alpha)$
Arr1	4	8	8	<b>16</b>	<b>32</b>	$0 \leq \alpha \leq 16$
Arr2	4	8	16	<b>24</b>	<b>32</b>	$0 \leq \alpha \leq 24$
Arr3	4	8	24	<b>32</b>	<b>32</b>	$0 \leq \alpha \leq 32$

Tabela 4.1: Dimenzije implementiranih *bit-plane* polja

Vrednosti širine ( $l_0$ ) i visine polja ( $m \cdot k$ ) sa slike 4.13 su u tabeli 4.1 zadebljane. Polja data u tabeli se razlikuju u širini ulazne reči ( $n$ ), koja uzrokuje različite širine polja ( $l_0$ ). Parametar  $m \cdot k$  ima istu vrednost kod svih polja datih u tabeli 4.1. Promenom odgovarajućeg parametra pomenute VHDL funkcije, sva tri polja implementirana su za  $\alpha$  u opsegu  $0 \leq \alpha \leq l_0$ , uz napomenu da je  $\mathbf{P}_{DT}(0)$  osnovna arhitektura sa slike 2.25, a  $\mathbf{P}_{DT}(l_0)$  arhitektura kod koje su sve ćelije projektovani kao visokopouzdanost sistemi.

Rezultati implementacije dati su u tabeli 4.2 [14]. U tabeli 4.2 uporedno su dati rezultati implementacije PDT polja sa rezultatima implementacije osnovne BP arhitekture (poglavlje 2, tabela 2.6). Rezultati implementacije osnovne arhitekture (tabela 2.6) prikazani su u vrsti  $\alpha = 0$  tabele 4.2. Za svaku vrstu iz tabele 4.1 tabela 4.2 sadrži dve kolone. U levoj koloni dat je ukupan broj potrebnih osnovnih ćelija, određen analitički na osnovu dimenzija polja i zadatog parametra  $\alpha$ , dok su u desnoj koloni dati rezultati FPGA implementacije izraženi u ekvivalentnom broju gejtova (kilo gejtova - kG) [14, 32].

$\alpha$	Arr1		Arr2		Arr3	
	Broj ćelija	Impl. [kG]	Broj ćelija	Impl. [kG]	Broj ćelija	Impl. [kG]
32	/	/	/	/	<b>3072</b>	<b>82.0</b>
24	/	/	<b>2304</b>	<b>61.6</b>	2986	80.2
16	<b>1536</b>	<b>41,4</b>	2218	59.8	2730	74.7
8	1450	39.6	1962	54.3	2304	65.4
4	1344	37.3	1770	50.1	2048	59.9
2	1274	35.8	1658	47.1	1920	57.1
1	1236	34.9	1598	46.4	1856	55.7
0	512	19.4	768	28.6	1024	37.8

Tabela 4.2: Broj osnovnih ćelija potrebnih za implementaciju arhitekture, i ekvivalentni broj gejtova FPGA implementacije za različite parametre implementacije *bit-plane* FIR filtra

Analitički određene vrednosti potrebnog broja ćelija objasnićemo na primeru arhitekture Arr1 (tabela 4.2). Naime, ukoliko se arhitektura projektuje tako da se ne povećava pouzdanost ni jednog izlaznog bita ( $\alpha = 0$ ), tada je broj potrebnih osnovnih ćelija jednak proizvodu visine i širine polja, odnosno

$$(m \cdot k) \cdot l_0 = 8 \cdot 4 \cdot 16 = 512.$$

Ukoliko se svi bitovi izlaznog rezultata projektuju sa povećanom pouzdanošću ( $\alpha = l_0=16$ ), tada se sve ćelije tripliciraju, pa je ukupan broj ćelija  $512 \cdot 3 = 1536$ , što je prokazano u tabeli 4.2.

Za ( $\alpha = 1$ ), na osnovu teoreme 4.1 i jednakosti (6.42), dobijamo da je 362 ćelije (od ukupno 512) potrebno projektovati kao visokopouzdanu. Tako dobijamo da je ukupan broj ćelija potreban za implementaciju sistema  $\mathbf{P}_{DT}(\alpha = 1)$  jednak

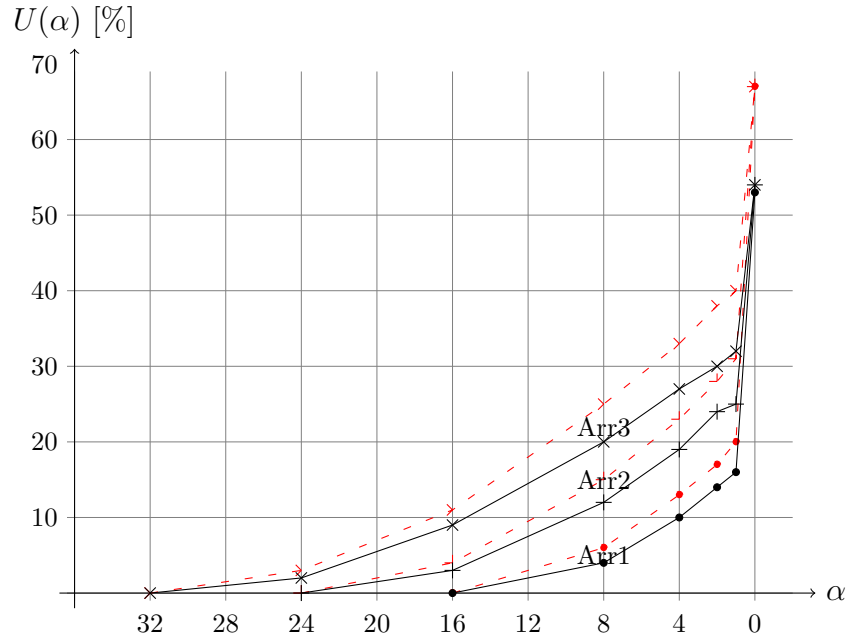
$$362 \cdot 3 + (512 - 362) \cdot 1 = 1236,$$

što je prikazano u tabeli 4.2 na mestu preseka kolone Arr1 i vrste  $\alpha = 1$ .

Broj osnovnih ćelija potrebnih za implementaciju arhitektura kod kojih je stepen parcijalne visoke pouzdanosti  $\alpha = l_0$ , kao i potreban ekvivalentni broj gejtova za implementaciju ovih arhitektura prikazan je zadebljano u tabeli 4.2. Ove vrednosti uzete su za referentne vrednosti prilikom određivanja uštede kod implementacija *bit-plane* polja za  $\alpha < l_0$ . Ušteta u pogledu potrebnih osnovnih ćelija za implementaciju *bit-plane* polja prikazana je na slici 4.23 [14]. Za analitički dobijene rezultate, kao i za rezultate FPGA

implementacije, ušteda je određena kao relativni odnos potrebnih resursa

$$U(\alpha) = \frac{\mathbf{P}_{DT}(l_0) - \mathbf{P}_{DT}(\alpha)}{\mathbf{P}_{DT}(l_0)} \cdot 100\%. \quad (4.17)$$



Slika 4.23: Relativni odnos resursa potrebnih za implementaciju arhitektura u funkciji od stepena parcijalne visoke pouzdanosti  $\alpha$

Analički određeni rezultati prikazani su isprekidanim linijama na slici 4.23, a rezultati FPGA implementacije punim linijama. Sa slike 4.23 može se uočiti da su rezultati FPGA implementacije neznatno pomereni naniže u odnosu na analitičke. Ovo se javlja zbog optimizacije koju vrši alat za FPGA sintezu, mada se sa slike 4.14 može uočiti da trend promene rezultata FPGA implementacije prati promenu analitički dobijenih rezultata. Takođe se sa slike 4.23 može uočiti da je ušteda kod svih analitičkih rezultata za  $\alpha = 0$  jednaka  $U(0) = 66.7$ . Ovo je direktna posledica izraza (4.17). Naime, imajući u vidu da je  $\mathbf{P}_{DT}(0) = \frac{\mathbf{P}_{DT}(l_0)}{3}$ , imamo da je

$$\frac{\mathbf{P}_{DT}(l_0) - \frac{\mathbf{P}_{DT}(l_0)}{3}}{\mathbf{P}_{DT}(l_0)} \cdot 100\% = 66.7\%.$$

Za parcijalno visokopouzdana *bit-plane* polja iz tabele 4.1 rezultati FPGA implementacije imaju vrednost  $\mathbf{P}_{DT}(l_0) \approx 56\%$  [14].

Sa slike 4.23 može se uočiti da se ušteda u odnosu na polje u celosti projektovano kao visokopouzdanost postiže za svako  $\alpha$ , za koje važi  $\alpha < l_0$  [14]. Ove je očekivano, imajući u vidu da je za ovu uštedu u površini plaćena cena u smanjenju pouzdanosti u odnosu na potpuno visokopouzdanost polje. Međutim, postavlja se pitanje opravdanosti uvođenja dodatnog hardvera u bilo kom slučaju. Naime, ukoliko je verovatnoća pojave defekta jednaka  $p = 0$ , ni najmanje povećanje obima hardvera u cilju postizanja pouzdanosti nije ekonomski opravdano.

U cilju nalaženja stepena parcijalne visokopouzdanosti  $\alpha$ , koji je ekonomski opravdan za zadatu verovatnoću pojave defekta ( $p$ ), u narednom poglavlju biće razmotren prinos fabrikacije arhitektura projektovanih predloženom tehnikom.

# Glava 5

## Prinos parcijalno visokopouzdanih sistema

U primeru datom u poglavlju 3.4.2 razmotren je slučaj FPGA implementacije arhitekture, koju čine 10 LUT tabela, sa 2 LUT tabele dodate u cilju postizanja veće pouzdanosti, koje mogu zameniti bilo koju od osnovnih 10 u slučaju njihovog otkaza. U primeru je ilustrovan zaključak dobijen iz jednačine (3.6), koji kaže da je pouzdanost sistema veća ukoliko se sistemu dodaju podsistemi koji mogu zameniti neispravne komponente. Bez obzira na povećanje pouzdanosti, postavlja se pitanje isplativosti ovakvog "ulaganja". Na početku ovog poglavlja razmotrićemo opravdanost dodavanja rezervnih komponenti za ovaj slučaj.

Verovatnoća otkaza komponente, data u primeru u poglavlju 3.4.2, je  $p = 10^{-4}$ . Verovatnoća da je sistem bez rezervnih komponenti ispravan na osnovu (3.6) je

$$R_G(10, 10) = (10^{-4})^{10} (1 - 10^{-4})^0 \approx 0.9990005. \quad (5.1)$$

Verovatnoća ispravnosti sistema sa rezervnim komponentama na osnovu (3.6) je

$$R_G(10, 12) \approx 0.9999999998 > 1 - 10^{-9}. \quad (5.2)$$

Iz (5.1) zaključujemo da bez dodavanja rezervnih komponenti na svakih 10.000.000 proizvedenih sistema imamo 9.990.005 ispravnih, a ostatak od 9.995 predstavlja škart procesa proizvodnje.

Cena povećanja pouzdanosti sa (5.1) na (5.2) je po 2 LUT-a po sistemu, ne računajući dodatnu upravljačku logiku koja vrši zamenu, što na 10 miliona sistema iznosi 20 miliona LUT-ova. Za ovu cenu, na 10 miliona sistema samo su 2 neispravna, što ćemo, radi jednostavnosti analize, zanemariti.

Dakle, da bi (skoro) svi sistemi bili ispravni, plaćena je cena od 20.000.000 LUT tabela. U slučaju sistema bez povećane pouzdanosti imali smo 9.995 neispravnih sistema, odnosno 99.995 LUT tabela je odbačeno kao neispravno. To znači, da ne bi odbacili skoro 100.000 LUT tabela, platili smo cenu od dodatnih 20 miliona LUT-ova za povećanje pouzdanosti. Grubo rečeno, u ovom slučaju se preko 200 puta više gubi na proizvodnji visokopouzdanog sistema nego na originalnom.

Kao što je pokazano prethodnom analizom na jednostavnom primeru, uvođenje visoke pouzdanosti za otklanjanje permanentnih defekata nastalih u procesu proizvodnje nije uvek opravdano [11, 14]. U ovom poglavlju biće data analiza prinosa primene predložene tehnike.

## 5.1 Prinos ET i PDT sistema

Neka je  $C$  cena fabrikacije jednog čipa, i neka je  $p$  verovatnoća pojave defekta u jedinici zamene (def. 3.7).

**Definicija 5.1 (Upotrebljivi čip)** *Za čip kažemo da je upotrebljiv ukoliko ne sadrži defekte, ili sadrži takve defekte da po definiciji 4.1 nije došlo do otkaza sistema. Za ostale čipove kažemo da su neupotrebljivi.*

U primeru FPGA implementacije u poglavlju 4.5 pokazano je da je PDT dizajn uvek isplativiji po pogledu zauzeća resursa na čipu od potpuno visokopouzdanog sistema, za aplikacije koje mogu tolerisati određene defekte [14]. Međutim, postavlja se pitanje isplativosti PDT dizajna u odnosu na ET dizajn. U cilji poređenja ET i PDT pristupa projektovanju definisaćemo cenu po upotrebljivom čipu.

**Definicija 5.2 (Cena upotrebljivog čipa)** *Cena upotrebljivog čipa je funkcija  $U(p, \alpha)$ , iz skupa  $\{(p, \alpha) \mid p \in [0, 1] \wedge \alpha \in \{0, 1, \dots, L\}\}$  u skup pozitivnih realnih brojeva, takvih da je*

$$U(p, \alpha) = u \cdot C,$$

*gde je  $\alpha$  broj bitova rezultata na kojima u konkretnoj primeni nije moguće tolerisati greške, a  $u$  ( $u \geq 1$ ) faktor koji zavisi od verovatnoće pojave defekta  $p$ , kao i od geometrije sistema [20].*

Ukoliko je verovatnoća pojave defekta  $p = 0$ , cena jednog upotrebljivog čipa jednaka je ceni jednog čipa, tj. za  $p = 0$  faktor  $u$  je jednak  $u = 1$ , bez obzira na geometriju veza u sistemu. Međutim, ukoliko je verovatnoća pojave defekta  $p \neq 0$ , cena upotrebljivog čipa je veća od  $C$ . Faktor  $u$  iz definicije 5.2 nazivamo prinos (eng. *yield*) [11, 19, 20].

**Definicija 5.3 (Prinos)** *Prinos dizajna je funkcija  $Y(p, \alpha)$  definisana kao [20]*

$$Y(p, \alpha) = \frac{C}{U(p, \alpha)}.$$

Prema definiciji 5.2, prinos je faktor, skaliran na interval  $Y \in [0, 1]$ , gde  $Y = 1$  znači da u procesu nema neupotrebljivih čipova, i  $Y = 0$  pokazuje da su svi čipovi neupotrebljivi. Vrednost  $Y = 1/2$  označava proces u kome treba proizvesti 2 čipa da bi dobili 1 upotrebljivi čip [20].

Neka je  $\Gamma(\alpha)$  verovatnoća da dati podsistem pripada netolerantnom delu arhitekture (def. 4.4), gde je  $\alpha$  stepen parcijalne visokopouzdanosti arhitekture (def. 4.8).

**Definicija 5.4 (Mera netolerantne oblasti)** *Mera veličine netolerantnog dela sistema je funkcija  $\Gamma(\alpha)$  iz skupa  $\{\alpha \mid \alpha = 0, 1, \dots, L\}$  u skup realnih brojeva iz intervala  $[0, 1]$ , takva da ima vrednost 0 ukoliko sistem nema netolerantni deo, i 1 ukoliko je ceo sistem netolerantan.*

Veličinu netolerantnog dela  $\Gamma(\alpha)$  moguće je odrediti iz skupa  $\mathbf{P}_{DT}(\alpha)$  (def. 4.8) i ova mera zavisi od posmatranog tipa greške, izabrane metrike i same topologije polja. Na kraju ovog poglavlja će na primeru ortogonalnog BP polja za FIR filtriranje biti ilustrovan način formiranja funkcije  $\Gamma(\alpha)$ .

Ukoliko sistem ima ukupno  $T$  podсистema, verovatnoća da će sistem biti upotrebljiv je

$$R(\alpha, p, T) = (1 - p)^{\Gamma(\alpha) \cdot T}.$$

Iz ovoga sledi tvrđenje leme:

**Lema 5.1** *Neka je  $T$  ukupan broj podсистema i neka je  $p$  verovatnoća da neki podсистem sadrži defekte, cena jednog upotrebljivog čipa je [20]*

$$U^{ET}(p, \alpha) = \frac{C}{(1 - p)^{\Gamma(\alpha) \cdot T}}. \quad (5.3)$$

**Lema 5.2** *Prinos ET sistema je [20]*

$$Y_{ET}(p, \alpha) = \frac{C}{U^{ET}(p, \alpha)} = (1 - p)^{\Gamma(\alpha) \cdot T}. \quad (5.4)$$

**Dokaz** Izraz (5.4) dobija se zamenom izraza (5.3) u izrazu iz definicije 5.3.  $\square$

Kako bi povećali broj upotrebljivih čipova sa što manje redundantnih resursa, u ovom radu autor predlaže primenu metoda za postizanje visoke



pouzdanosti samo na podsistemima u netolerantnom delu, definisanom funkcijom  $\Gamma(\alpha)$  [20].

Cena sistema sa dodatim redundantnim podsistemima se menja u odnosu na cenu u izrazu (5.3). Neka je cena sistema sa redundantnim podsistemima  $C'$ . Takođe, izraz za verovatnoću da je podsistem ispravan je kompleksniji od izraza  $(1 - p)$ , datog jednačinom (5.3), i zavisi od verovatnoće otkaza visokopouzdanog sistema. Označimo ovu verovatnoću sa  $R$ .

**Lema 5.3** *Cena upotrebljivog čipa na kome je implementiran PDT sistem je [20]*

$$U^{PDT}(p, \alpha) = \frac{C'}{R^{\Gamma(\alpha) \cdot T}}. \quad (5.5)$$

**Lema 5.4** *Prinos PDT sistema je [20]*

$$Y_{PDT}(p, \alpha) = \frac{C}{U^{PDT}(p, \alpha)} = \frac{C}{C'} \cdot R^{\Gamma(\alpha) \cdot T}. \quad (5.6)$$

**Dokaz** Jednačina (5.6) izvedena je iz (5.5) i definicije 5.3.  $\square$

Prethodno tvrđenje ilustrovaćemo na primeru.

**Primer** Neka je izabrani visokopouzdan metod SC3 (poglavlje 3.3.1). Verovatnoća da je SC3 sistem ispravan data je izrazom (3.3) i iznosi

$$R = (1 - p)^3 + 3(1 - p)^2p + 3(1 - p)p^2.$$

U ovom slučaju, umesto svakog podsistema u netolerantnom delu arhitekture nalaze se 3 podsistema. Kako je veličina netolerantnog dela  $\Gamma(\alpha)$ , nova cena PDT sistema je

$$C' = \Gamma(\alpha) \cdot 3 \cdot C + (1 - \Gamma(\alpha)) \cdot C = (1 + 2\Gamma(\alpha))C, \quad (5.7)$$

Prinos PDT sistema dobijamo zamenom (5.7) i (3.3) u (5.6). Uporedni prikaz prinosa ET sistema (5.4) i PDT sistema opisanog u ovom primeru dat je na slici 5.1. U cilju dobijanja numeričkih vrednosti pretpostavljeno je da je ukupan broj podsistema  $T = 168$ , koliko ima polje sa slike 2.25, i da je veličina netolerantnog dela  $\Gamma(\alpha) = 0.5$ .

Sa slike 5.1 se može uočiti da oba prinosa opadaju sa porastom verovatnoće pojave defekta. Za vrednost  $p = 0$  vrednost prinosa PDT sistema je  $Y_{PDT} = 0.5$  (slika 5.1). Ovo proizilazi iz činjenice da su u cenu PDT čipa uključeni redundantni SC3 sistemi (5.7), tako da je cena po čipu, za date parametre duplirana. Istovremeno, prinos ET sistema je  $Y_{ET} = 1$ , što znači da je cena proizvodnje jednog čipa jednaka ceni proizvodnje jednog upotrebljivog

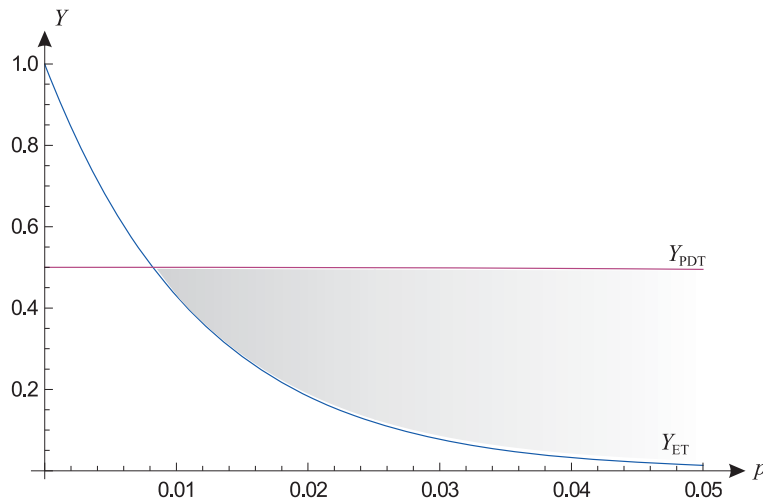
p	0	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
$U^{ET}$	1.	1.087	1.183	1.287	1.400	1.523	1.657	1.804	1.963	2.137
$U^{PDT}$	2.	2.000	2.002	2.004	2.008	2.012	2.018	2.024	2.032	2.041

Tabela 5.1: Cene upotrebljivog čipa za različite verovatnoće otkaza  $p$ 

čipa. Međutim, sa slike se može uočiti da postoji  $p$  počev od kog prinos PDT sistema postaje veći od prinosa ET sistema. Ova verovatnoća zavisi od funkcije  $\Gamma(\alpha)$ , koju oblikuje topologija sistema, kao i stepena parcijalne visoke pouzdanosti  $\alpha$ .

U cilju ilustracije, u tabeli 5.1 date su numeričke vrednosti cena upotrebljivog čipa za prinose sa slike 5.1. Za slučaj kada je verovatnoća pojave defekta  $p = 0.003$ , kod ET sistema potrebno je proizvesti 1.287 čipova da bi se dobio jedan upotrebljivi čip. Za istu verovatnoću, u slučaju PDT sistema potrebno je proizvesti 2.004, što čini ET sistem isplativijim. Za verovatnoću  $p$  veću od 0.008, PDT je isplativiji. Osenčana oblast na slici 5.1 pokazuje verovatnoće za koje je PDT sistem, za parametre iz ovog primera, isplativiji od ET sistema.  $\triangle$

**Primer** Razmotrimo primer projektovanja PDT polja kod koga je u cilju povećanja pouzdanosti ćelija izabrana TMR metoda. Verovatnoća da je TMR



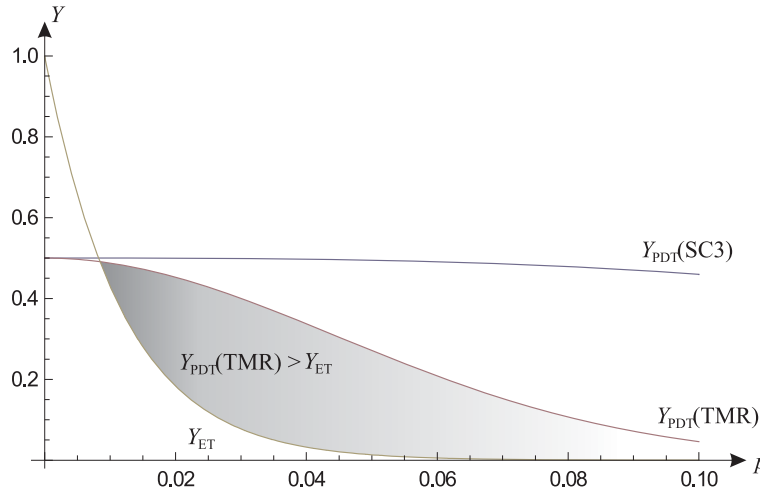
Slika 5.1: Uporedni prikaz prinosa za slučaj projektovanja sistema ET i PDT metodom sa SC3 ćelijama u netolerantnoj oblasti, za  $\Gamma(\alpha) = 0.5$  i ukupan broj podsistema  $T = 168$  [20]

sistem ispravan data je izrazom (3.2) i iznosi

$$R_{TMR} = (1 - p)^3 + 3 \cdot (1 - p)^2 \cdot p.$$

Cena PDT sistema za slučaj upotrebe TMR tehnike jednaka je ceni iz prethodnog primera kod koga je primenjena SC3 tehnika, i opisana je izrazom (5.7).

Prinos PDT sistema dobijamo zamenom (3.2) i (5.7) u (5.6). Funkcija  $Y_{PDT}(p, \alpha)$  je za slučaj primene TMR tehnike prikazana na slici 5.2. Na slici 5.2 su u cilju ilustracije međusobnog odnosa prinosa za slučaj primene TMR i SC3 tehnike prikazane i funkcije prinosa sa slike 5.1. Oblast u kojoj PDT sistem daje veći prinos od ET sistema je osenčana (slika 5.2).



Slika 5.2: Uporedni prikaz prinosa za slučaj projektovanja sistema ET i PDT metodom sa TRM ćelijama u netolerantnoj oblasti, za  $\Gamma(\alpha) = 0.5$  i ukupan broj podsistema  $T = 168$  [20]

△

## 5.2 Poređenje prinosa ET i PDT sistema

U cilju određivanja međusobnog odnosa prinosa ET i PDT metoda, u ovom poglavlju dat je postupak nalaženja tačke preseka funkcija (5.4) i (5.6) za zadanu geometriju polja. Funkcije (5.4) i (5.6) seku se za vrednost verovatnoće pojave defekta  $p$  za koju obe tehnike projektovanja imaju podjednak prinos. Određivanjem tačke preseka, možemo naći vrednost verovatnoće  $p$  počev od koje PDT metod ima veći prinos od ET, ili obrnuto, zavisno da li se posmatra smanjenje, ili povećanje verovatnoće.

Verovatnoća  $p$  počev od koje PDT pristup daje veći prinos u odnosu na ET može se odrediti iz preseka funkcija (5.4) i (5.6), ilustrovanih u primerima na slikama 5.1 i 5.2. Označimo sa  $\gamma = \Gamma(\alpha)$  funkciju koja u zavisnosti od topologije definiše veličinu netolerantnog dela arhitekture, tada iz (5.4) i (5.6) sledi [20]

$$(1 - p)^{\gamma T} = \frac{C}{C'} R^{\gamma T}. \quad (5.8)$$

Neka je u cilju povećanja pouzdanosti ćelija izabrana SC3 tehnika (poglavlje 3.3.1).

**Teorema 5.1** *Verovatnoća počev od koje je prinos PDT polja sa SC3 tehnikom za povećanje pouzdanosti podsistema veći od prinosa ET polja je [20]*

$$p = \frac{1}{2} \left( -1 + \sqrt{-3 + 4(1 + 2\Gamma(\alpha))^{\frac{1}{\Gamma(\alpha)T}}} \right). \quad (5.9)$$

**Dokaz** Jednakost (5.8) je potrebno rešiti po  $p$ . Zamenimo  $R$  i  $C'$  iz izraza (3.3) i (5.7) respektivno u (5.8). Dobijamo

$$(1 - p)^{\gamma T} = \frac{C}{(1 + 2\gamma) \cdot C} \cdot ((1 - p)^3 + 3 \cdot (1 - p)^2 \cdot p + 3 \cdot (1 - p) \cdot p^2)^{\gamma T},$$

odnosno

$$(1 + 2\gamma) = ((1 - p)^2 + 3(1 - p)p + 3p^2)^{\gamma T}. \quad (5.10)$$

Jednačina (5.10) je nelinearna. U cilju linerizacije logaritmoaćemo obe strane jednakosti i označiti izraz sa  $A$

$$A = \frac{\log(1 + 2\gamma)}{\gamma \cdot T} = \log[(1 - p)^2 + 3(1 - p)p + 3p^2]. \quad (5.11)$$

Neka je  $B = e^A$ , tada je

$$\begin{aligned} B = e^A &= (1 - p)^2 + 3(1 - p)p + 3p^2 = \\ &= p^2 + p + 1. \end{aligned} \quad (5.12)$$

Rešavanjem kvadratne jednačine (5.12) po  $p$  dobijamo dva rešenja

$$p_{1,2} = \frac{1}{2}(-1 \pm \sqrt{-3 + 4B}). \quad (5.13)$$

Kako je  $A \geq 0$ , sledi da je  $B \geq 1$ . Na osnovu ovoga, rešenje (5.13) sa predznakom minus nije validno i dalo bi kao rezultat negativnu verovatnoću, pa ga kao takvo odbacujemo.

Neka je  $P(\alpha, T)$  funkcija iz  $\{(\alpha, T) \mid \alpha = 0, \dots, L, T \in \mathbb{N}\}$  u  $p \in [0, 1]$  za koju se funkcija sa slike 5.1 seku. Uz korišćenje činjenice da je  $e^{\frac{\log a}{b}} = a^{1/b}$ , imamo da je

$$B = e^{\frac{\log 1+2\gamma}{\gamma T}} = (1 + 2\gamma)^{1/\gamma T}, \quad (5.14)$$

pa se iz (5.13) i (5.14) dobija

$$P(\alpha, T) = \frac{1}{2}(-1 + \sqrt{-3 + 4(1 + 2\Gamma(\alpha))^{\frac{1}{\Gamma(\alpha)T}}}),$$

čime je tvrđenje teoreme dokazano.  $\square$

Razmotrimo slučaj kada je u cilju povećanja pouzdanosti ćelija PDT polja izabrana TMR tehnika (poglavlje 3.3.1).

**Teorema 5.2** *Opseg verovatnoća za koje je prinos PDT polja sa TMR tehnikom za povećanje pouzdanosti podsistema veći od prinosa ET polja je*

$$p \in [p_1, p_2], \quad p_{1,2} = \frac{1}{4} \left( 1 \pm \sqrt{9 - 8(1 + 2\Gamma(\alpha))^{\frac{1}{\Gamma(\alpha)T}} \right), \quad (5.15)$$

gde je  $p_1$  uzeto za znakom "-", a  $p_2$  sa znakom "+".

**Dokaz** Jednakost (5.8) je potrebno rešiti po  $p$ . Zamenimo  $R$  i  $C'$  iz izraza (3.2) i (5.7) respektivno u (5.8). Dobijamo

$$(1 - p)^{\gamma T} = \frac{C}{(1 + 2\gamma) \cdot C} \cdot ((1 - p)^3 + 3 \cdot (1 - p)^2 \cdot p)^{\gamma T},$$

odnosno

$$(1 + 2\gamma) = ((1 - p)^2 + 3(1 - p)p)^{\gamma T}. \quad (5.16)$$

Jednačina (5.16) je nelinearna. Rešavanjem jednačine (5.16) na isti način kao u dokazu teoreme 5.1, dobijamo dva rešenja

$$p_{1,2} = \frac{1}{4}(1 \pm \sqrt{9 - 8B}). \quad (5.17)$$

Zavisno od parametara  $\gamma$  i  $T$ , oba rešenja (5.17) mogu dati validne verovatnoće, što ukazuje da funkcije (5.4) i (5.6) imaju dve tačke preseka za skup verovatnoća  $p \in [0, 1]$ . Za TMR je ovo očekivano, kako je pokazano u poglavlju 3.3.1 i ilustrovano na slici 3.10, verovatnoća da je TMR sistem ispravan za  $p > 0.5$  je manja od verovatnoće da je osnovni sistem ispravan. Na slici 5.2 u primeru iz prethodnog poglavlja takođe je moguće uočiti ovaj trend, a na osnovu (5.17) i zaključiti da sigurno postoji verovatnoća kada se prinosi još jednom seku.

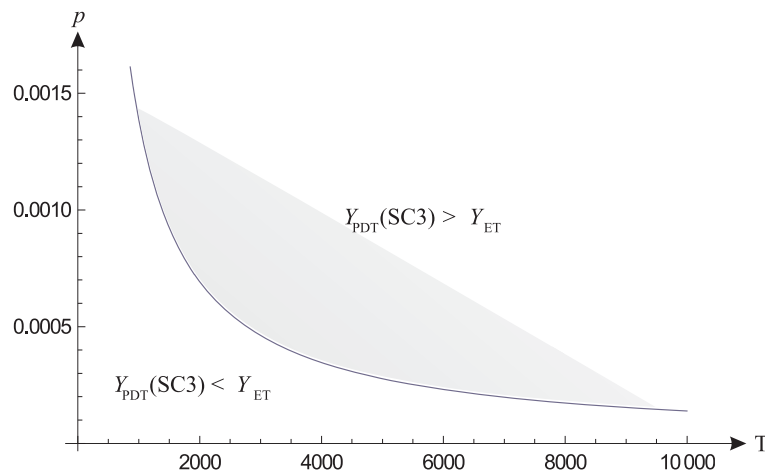
Utvrđivanjem međusobnog odnosa rešenja (5.17) i zamenom (5.14) u (5.17) dobija se tvrđenje teoreme.  $\square$

Po tvrđenju teorema 5.1 i 5.2, verovatnoće za koje PDT tehnika projektovanja daje veći prinos od ET tehnike preko funkcije  $\Gamma(\alpha)$  posredno zavise od tolerantnosti polja na greške  $\alpha$ . Primer određivanja mere netolerantne oblasti ortogonalnog polja i uticaja  $\alpha$  na verovatnoću  $p$  za koju se prinosi seku dat je u narednom poglavlju. Naredna glava posvećena je analizi uticaja topologije i izabrane metrike na oblik funkcije  $\Gamma(\alpha)$ . U daljem tekstu razmotrićemo uticaj broja podsistema  $T$ , kao bitan faktor koji utiče na presek prinosa i verovatnoće date izrazima (5.9) i (5.15).

### 5.2.1 Uticaj broja podsistema na prinos PDT sistema

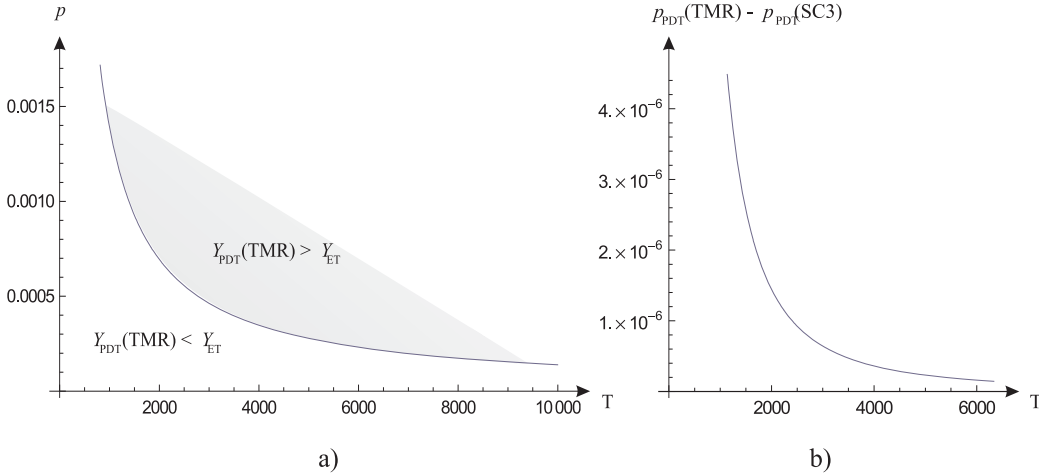
Izrazima (5.9) i (5.15) određena je promena tačke preseka jednačina (5.4) i (5.6) za slučaj primene SC3 i TMR tehnike, respektivno. U oba izraza broja ćelija polja  $T$  nalazi se u eksponentu. Kako  $\gamma$  zavisi od topologije i izabrane metrike, u cilju ilustracije zavisnosti verovatnoće  $p$  od broja ćelija polja uzećemo konstantno  $\gamma$  ( $\gamma = 0.5$ ).

Grafik zavisnosti verovatnoće za koju se prinosi ET i PDT tehnike sa SC3 ćelijama seku prikazan je na slici 5.3. Sa slike 5.3 se može uočiti da verovatnoća za koju se funkcije prinosa seku eksponencijalno opada sa porastom broja ćelija filtra, i to u korist PDT polja, što se vidi i iz jednačine (5.9).



Slika 5.3: Grafik zavisnosti verovatnoće za koju se prinosi ET i PDT tehnike sa SC3 ćelijama seku za  $\gamma = 0.5$ , u funkciji broja ćelija polja  $T$

Na slici 5.4(a) prikazana je zavisnost verovatnoće za koju se prinosi ET i PDT tehnike sa TMR ćelijama seku. Sa slike 5.4(a) može se uočiti isti trend promene tačke preseka (5.15) kao i slučaju funkcije promene tačke preseka (5.9) prikazanog na slici 5.3. Zavisnost tačke preseka od broja podsistema  $T$  takođe je i u ovom slučaju eksponencijalna.



Slika 5.4: a) Grafik zavisnosti verovatnoće za koju se prinosi ET i PDT tehnike sa TMR ćelijama seku za  $\gamma = 0.5$ , u funkciji broja ćelija polja  $T$ , b) grafik razlike funkcija (5.9) i (5.15) za  $\gamma = 0.5$  u funkciji od  $T$

Sa grafika funkcija datih na slici 5.2 može se uočiti da su tačke preseka prinosa  $Y_{ET}$ , i prinosa  $Y_{PDT}$  za SC3 i TMR tehniku relativno blizu. Ovdje ne govorimo o gornjoj granici i rešenju  $p_2$  iz (5.15), jer, kako je u poglavlju 3.3.1 rečeno, da bi proces fabrikacije bio ekonomski isplativ, verovatnoće pojave defekta moraju biti daleko ispod  $p = 0.5$ . Na slici 5.4(b) prikazana je razlika (5.9) i (5.15) za  $\gamma = 0.5$  u funkciji od  $T$ . Sa slike se može videti da je razlika reda veličine  $10^{-6}$  i ispod, zavisno od broja podsistema  $T$ . Ovaj grafik pokazuje kako verovatnoća za koju PDT i ET daju isti prinos zavisi od tehnike za povećanje pouzdanosti podsistema.

Kako je zavisnost verovatnoće za koju se funkcije (5.9) i (5.15) seku eksponencijalna u odnosu na broj podsistema  $T$ , i u slučaju SC3, i u slučaju TMR tehnike, a razlika primene SC3 ili TMR tehnike se značajno smanjuje sa povećanjem broja podsistema (slika 5.4b), za analizu arhitektura u narednim poglavljima razmatraćemo uglavnom ponašanje sistema u odnosu na jednu od ove dve tehnike.

Na osnovu tvrđenja teorema 5.1 i 5.2 datih izrazima (5.9) i (5.15), i ilustrovanih na slikama 5.3 i 5.4, može se zaključiti da primena parcijalne

tolerancije na defekte može imati značajno veći prinos i biti dobro iskorišćena u nanotehnologiji. Iz (3.5) u poglavlju 3.4.2 zaključeno je da "kod današnjih sistema sa  $N > 10^9$  komponenata, verovatnoća otkaza pojedinih komponenti mora biti ispod  $10^{-10}$  kako bi bilo moguće očekivati 90 procenata, ili više ispravnih čipova" [11, p. 833]. Ukoliko se verovatnoća pojave otkaza poveća samo 10 puta i dostigne vrednost od  $10^{-9}$ , ili čak i više, što je slučaj u nanotehnologiji, prinos ET polja pada značajno ispod prinosa PDT polja. Za ovakve sisteme isplativije je primeniti PDT metod projektovanja polja.

**Primer** Da bi numerički ilustrovali ovo tvrđenje, uzmimo za verovatnoću pojave defekta  $p = 10^{-8}$ , i broj podsistema tipičan za nanotehnologiju  $T = 10^9$ . Prinos ET polja, na osnovu (5.4) je  $Y_{ET} = 0.0067$ . Prinos PDT polja za istu verovatnoću je, na osnovu (5.6),  $Y_{PDT} = 0.5$  [20].  $\triangle$

Zavisnost verovatnoće za koju se funkcije prinosa ET i PDT tehnike seku eksponencijalno zavisi od broja podsistema  $T$ . U izrazima (5.9) i (5.15), posredno preko veličine netolerantne oblasti arhitekture  $\Gamma(\alpha)$ , figuriše tolerantnost arhitekture na greške. U cilju ilustracije uticaja topologije polja, izabrane metrike otkaza sistema i tolerantnosti sistema na verovatnoću počev od koje je prinos PDT tehnike veći od prinosa ET tehnike, u narednom poglavlju biće dat postupak određivanja funkcije  $\Gamma(\alpha)$  na relativno složenom primeru ortogonalnog semi-sistoličkog polja *bit-plane* FIR filtra.

### 5.3 Netolerantna oblast ortogonalnog polja

Funkcija  $\Gamma(\alpha)$ , opisana definicijom 5.4, zavisi od topologije polja, a zavisnost je data definicijama 4.5 do 4.8. U ovom poglavlju biće ilustrovan postupak određivanja veličine netolerantne oblasti na primeru ortogonalnog semi-sistoličkog polja *bit-plane* FIR filtra sa slike 2.25, kao predstavnika strogo regularnih topologija tipa  $4^4$ .

Neka je  $L$  ukupan broj kolona grafa i ujedno ukupan broj izlaznih čvorova,  $h$  ukupan broj vrsta i  $T$  ukupan broj čvorova grafa. Neka je  $\aleph(A)$  kardinalni broj (čita se *alef od A*), odnosno kardinalnost skupa  $A$  ([98, p. 28]). Za graf propagacije greške sa topologijom  $4^4$ , prikazanog na slici 4.13, funkcija  $\Gamma(\alpha)$  data je u [20] i može se predstaviti sledećom teoremom.

**Teorema 5.3** *Verovatnoća da čvor pripada netolerantnom delu digrafa  $\mathbf{G}$  topologije  $4^4$  je [20]*

$$\Gamma(\alpha) = \frac{\aleph(\mathbf{P}_{DT}(\alpha))}{T} = \begin{cases} 0, & \alpha = 0 \\ \frac{h+3}{2L}, & \alpha = 1 \\ \frac{h^2+h-2+\alpha(2h+3)-\alpha^2}{2T}, & \text{inače} \end{cases} \quad (5.18)$$



**Dokaz** Teoremu ćemo dokazati na primeru grafa propagacije greške sa topologijom  $4^4$  sa slike 4.13. Ukupan broj čvorova grafa sa slike 4.13 je

$$T = (m \cdot k) \cdot (m + l_0) = h \cdot L. \quad (5.19)$$

Po tvrđenju leme 4.3, skup  $\mathbf{P}_{DT}(\alpha)$ , dat izrazom (4.3) u definiciji 4.8, sadrži sve čvorove neotolerantnog dela grafa. Broj čvorova u skupu  $\mathbf{P}_{DT}(\alpha)$  jednak je  $\aleph(\mathbf{P}_{DT}(\alpha))$ , pa je verovatnoća da čvor pripada netolerantnom delu grafa jednaka

$$\Gamma(\alpha) = \frac{\aleph(\mathbf{P}_{DT}(\alpha))}{T}. \quad (5.20)$$

Svaki čvor grafa sa slike 4.13 ima uticaj na više izlaznih čvorova, odnosno pripada različitim mapama uticaja. To znači da skupovi  $\mathbf{M}_\eta$ ,  $\eta = 0, 1, \dots, L-1$  iz (4.3) imaju zajedničke elemente, odnosno važi da je

$$\mathbf{M}_i \cap \mathbf{M}_j \neq \emptyset, \quad i, j \in \{0, 1, \dots, L-1\},$$

što se vidi iz mapa uticaja greške  $M_\eta$ ,  $\eta = 0, 1, \dots, 5$ , koje odgovaraju skupovima  $\mathbf{M}_\eta$  u primeru datom u poglavlju 4.4. Kako ne bi više puta uzeli u obzir istu ćeliju prilikom određivanja kardinalnosti skupa  $\aleph(\mathbf{P}_{DT}(\alpha))$ , ova činjenica mora biti razmotrena. Kardinalnost skupa (4.3), imajući u vidu prethodno, može se odrediti na sledeći način:

$$\begin{aligned} \aleph(\mathbf{P}_{DT}(\alpha)) &= \aleph(\mathbf{M}_{L-1}) + \aleph(\mathbf{M}_{L-2} \setminus \mathbf{M}_{L-1}) + \\ &+ \aleph\left(\mathbf{M}_{L-3} \setminus \left(\mathbf{M}_{L-1} \cup \mathbf{M}_{L-2}\right)\right) + \\ &+ \aleph\left(\mathbf{M}_{L-\alpha} \setminus \bigcup_{i=L-1}^{L-\alpha-1} \mathbf{M}_i\right) = \\ &= \sum_{j=L-1}^{L-\alpha} \aleph\left(\mathbf{M}_j \setminus \left(\bigcup_{i=L-1}^{j-1} \mathbf{M}_i\right)\right), \end{aligned} \quad (5.21)$$

gde  $A \setminus B$  označava skup sa elementima koji pripadaju skupu  $A$ , a ne pripadaju skupu  $B$ . Drugim rečima,  $\aleph(\mathbf{P}_{DT}(\alpha))$  jednak je broju ćelija koje mogu prouzrokovati grešku na bitu  $y^{L-1}$  ( $\aleph(\mathbf{M}_{L-1})$  u izrazu (5.21)), plus broj ćelija polja koje mogu prouzrokovati grešku na izlaznom bitu  $y^{L-2}$ , bez ćelija ovog skupa koje su već uzete u obzir ( $\aleph(\mathbf{M}_{L-2} \setminus \mathbf{M}_{L-1})$ ), itd.

Kardinalnost pojedinog skupa  $\mathbf{M}_\eta$  može se odrediti iz tranzitivnog zatvaranja grafa sa slike 4.13, datog u poglavlju 4.3 jednačinom (4.9) u okviru teoreme 4.1.

Uzmimo primer relativno malog polja *bit-plane* FIR filtra datog na slici 4.16 u poglavlju 4.4, sa parametrima  $k = 2$ ,  $m = 2$ ,  $l_0 = 4$  and  $L = 6$ . Ukupan broj čvorova proširenog grafa sa slike 4.18 je  $L \cdot (k + m + 1) = 30$ , uključujući i izlazne čvorove. Transitivno zatvaranje (4.9) ovakvog grafa je matrica dimenzija 30x30, data na slici 4.19.

Kolona 24 tranzitivnog zatvaranja sa slike 4.19 odgovara mapi uticaja greške  $\mathbf{M}_5$  izlaznog bita  $y^5$ , i istaknuta je na slici 5.5. Broj jedinica u 24. koloni jednak je kardinalnom broju  $\aleph(\mathbf{M}_5)$ .

					$M_5 M_4 M_3 M_2 M_1 M_0$	
	0 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0	}
	0 0 0 0 0 0	1 1 0 0 0 0	1 1 0 0 0 0	1 1 0 0 0 0	1 1 0 0 0 0	
	0 0 0 0 0 0	0 1 1 0 0 0	1 1 1 0 0 0	1 1 1 0 0 0	1 1 1 0 0 0	
	0 0 0 0 0 0	0 0 1 1 0 0	0 1 1 1 0 0	1 1 1 1 0 0	1 1 1 1 0 0	
	0 0 0 0 0 0	0 0 0 1 1 0	0 0 1 1 1 0	0 1 1 1 1 0	1 1 1 1 1 0	
	0 0 0 0 0 0	0 0 0 0 1 1	0 0 0 1 1 1	0 0 1 1 1 1	0 1 1 1 1 1	
	0 0 0 0 0 0	0 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0	}
	0 0 0 0 0 0	0 0 0 0 0 0	1 1 0 0 0 0	1 1 0 0 0 0	1 1 0 0 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 1 1 0 0 0	1 1 1 0 0 0	1 1 1 0 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 1 1 0 0	0 1 1 1 0 0	1 1 1 1 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 1 1 0	0 0 1 1 1 0	0 1 1 1 1 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 1 1	0 0 0 1 1 1	0 0 1 1 1 1	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0	}
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	1 1 0 0 0 0	1 1 0 0 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 1 1 0 0 0	1 1 1 0 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 1 1 0 0	0 1 1 1 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 1 1 0	0 0 1 1 1 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 1 1	0 0 0 1 1 1	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	1 0 0 0 0 0	}
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	1 1 0 0 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 1 1 0 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 1 1 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 1 1 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 1 1	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	}
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	
	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	

$A^* =$ 
 $m \cdot k \cdot L$

1 2 3 4 ...
 $\uparrow$  24<sup>th</sup>

Slika 5.5: Transitivno zatvaranje i određivanje kardinalnosti skupa  $\aleph(\mathbf{P}_{DT}(1))$  *bit-plane* FIR filtra za  $k = 2$ ,  $m = 2$ ,  $n = 2$  i  $l_0 = 4$

Na osnovu (5.21) kardinalnost za  $\alpha = 1$  je  $\aleph(\mathbf{P}_{DT}(1)) = \aleph(\mathbf{M}_5)$ . Ovaj kardinalni broj može se odrediti prebrojavanjem jedinica u (4.9), odnosno sa

slike 5.5 i iznosi

$$\begin{aligned}\aleph(\mathbf{P}_{DT}(1)) &= 2 + 3 + \dots + (m \cdot k + 1) = \\ &= \sum_{i=2}^{m \cdot k + 1} i = \frac{m \cdot k \cdot (m \cdot k + 3)}{2}.\end{aligned}$$

Pored ćelija obuhvaćenih izrazom (5.22),  $m \cdot k$  ćelija treba biti uključeno da bi dobili kardinalnost  $\aleph(\mathbf{P}_{DT}(2))$ , (kolona  $\mathbf{M}_4$  na slici 5.5). Da bi odredili  $\aleph(\mathbf{P}_{DT}(3))$  treba dodati jednu manje, tj.  $m \cdot k - 1$ , itd.

$$\begin{aligned}\aleph(\mathbf{P}_{DT}(2)) &= \aleph(\mathbf{P}_{DT}(1)) + m \cdot k \\ \aleph(\mathbf{P}_{DT}(3)) &= \aleph(\mathbf{P}_{DT}(2)) + m \cdot k - 1 \\ &\dots \\ \aleph(\mathbf{P}_{DT}(L)) &= \aleph(\mathbf{P}_{DT}(L - 1)) + m \cdot k - (L - 2).\end{aligned}\quad (5.22)$$

Iz (5.22) i (5.22), za  $\alpha > 1$  imamo

$$\begin{aligned}\aleph(\mathbf{P}_{DT}(\alpha)) &= \sum_{i=2}^{m \cdot k + 1} i + (\alpha - 1) \cdot m \cdot k - \sum_{i=1}^{\alpha - 2} i = \\ &= \frac{m \cdot k \cdot (m \cdot k + 3)}{2} + (\alpha - 1) \cdot m \cdot k - \frac{(\alpha - 1)^2 - \alpha + 1}{2} = \\ &= \frac{k^2 m^2 + km - 2 + \alpha(2km + 3) - \alpha^2}{2}.\end{aligned}\quad (5.23)$$

Kombinovanjem izraza (5.22) i (5.23) dobijamo

$$\aleph(\mathbf{P}_{DT}(\alpha)) = \begin{cases} 0, & \alpha = 0 \\ \frac{m \cdot k \cdot (m \cdot k + 3)}{2}, & \alpha = 1 \\ \frac{k^2 m^2 + km - 2 + \alpha(2km + 3) - \alpha^2}{2}, & \text{inače} \end{cases}\quad (5.24)$$

Konačno, zamenom (5.19) i (5.24) u (5.20) dobija se tvrđenje teoreme.  $\square$

Prethodnu teoremu ilustrovaćemo na primeru. Na osnovu (5.24), numeričke vrednosti kardinalnosti  $\mathbf{P}_{DT}(\alpha)$ ,  $\alpha = 0, 1, 2, \dots, 6$ , za primer iz poglavlja 4.4 sa parametrima  $m = 2, k = 2, L = 6$  su

$$\aleph(\mathbf{P}_{DT}(0)) = 0, \quad \aleph(\mathbf{P}_{DT}(1)) = 14, \quad \aleph(\mathbf{P}_{DT}(2)) = 18, \dots \quad (5.25)$$

što je moguće proveriti iz tranzitivnog zatvaranja sa slike 5.5.

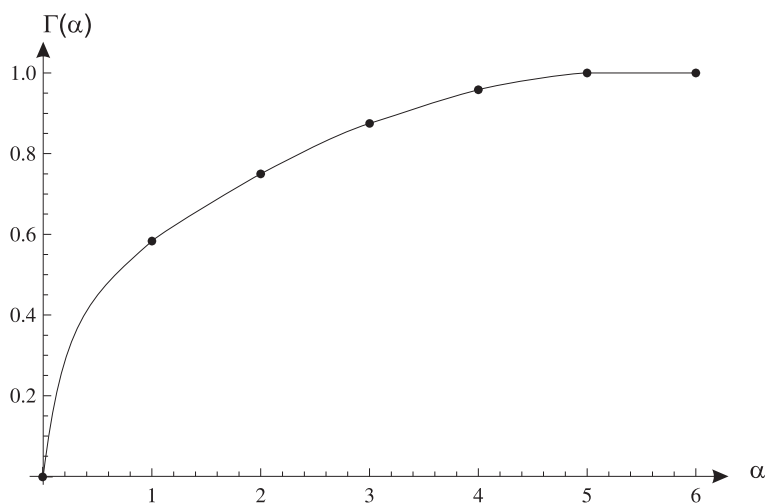
Na slici 5.6 prikazana je funkcija  $\Gamma(\alpha)$  za numeričke vrednosti (5.25). Slika ilustruje povećanje netolerantnog dela arhitekture sa povećanjem broja

značajnih bitova rezultata za primer sa  $m = 2, k = 2, L = 6$  polja sa slike 4.16. Potrebno je napomenuti da su vrednosti na slici 5.6 diskretne. Tačke su povezane linijom u cilju jasnijeg prepoznavanja trenda promene vrednosti funkcije sa promenom parametra  $\alpha$ .

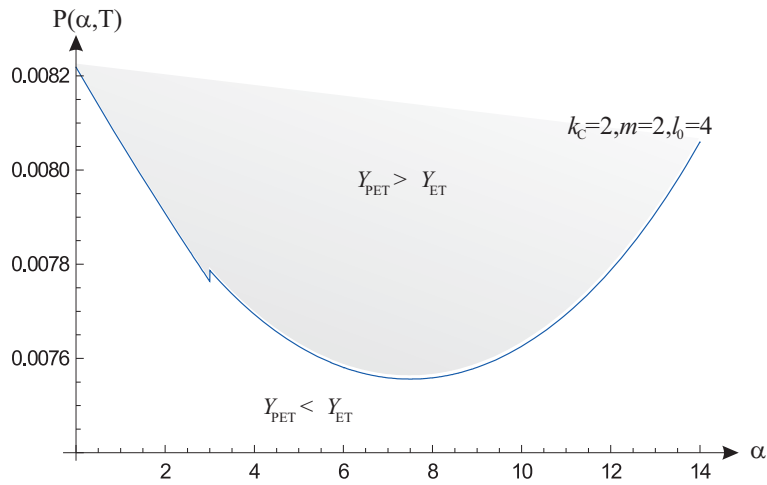
U cilju ilustracije zavisnosti verovatnoće  $p$  za koju se funkcije prinosa ET (5.9) i PDT tehnike (5.15) seku kod strogo regularnog ortogonalnog polja ( $4^4$ ), u odnosu na tolerantnost polja na greške ( $\alpha$ ), na slici 5.7 prikazana je promena tačke preseka u zavisnosti od  $\alpha$  [20]. Parametri polja sa slike 2.25 za koje je formiran grafik su  $k = 3, m = 4, n = 5$ , za koje je  $T = 168$ . Vrednosti na osi  $P(\alpha, T)$  su zumirane kako bi se istakla veoma mala promena  $p$ -a sa promenom tolerancije  $\alpha$ . Vrednost funkcije je prikazana u opsegu od  $p = 0.0076$  do  $p = 0.0082$ . Sa slike 5.7 se može uočiti relativno mala zavisnost verovatnoće pojave defekta od promene tolerancije polja na pojavu defekta. Drugim rečima, za date dimenzije polja, prinos PDT polja u odnosu na ET polje veći kada je verovatnoća pojave defekta veća od 0.0082, bez obzira na stepen tolerantnosti polja na defekte.

U ovom poglavlju određen je prinos i međusobni odnos ET i PDT tehnika projektovanja sistema. Posebna pažnja posvećena je određivanju isplatljivosti PDT tehnike u odnosu na ET u funkciji broja podsistema. Pokazano je da predloženi metod može značajno povećati prinos kod sistema sa velikim brojem komponenata, kao što je to slučaj u nanotehnologiji.

Po tvrđenju teorema 5.1 i 5.2, verovatnoće za koje PDT tehnika projektovanja daje veći prinos od ET tehnike, posredno preko funkcije  $\Gamma(\alpha)$  zavise



Slika 5.6: Grafik funkcije  $\Gamma(\alpha)$  bit-plane FIR filtra za  $m = 2, k = 2, L = 6$



Slika 5.7:  $P(\alpha)$  za  $k = 3, m = 4, n = 5$  i  $T = 168$  u funkciji parametra  $\alpha$

od tolerantnosti polja na greške  $\alpha$ . Ova zavisnost izvedena je za primer ortogonalnog polja i data u okviru teoreme 5.3. Naredna glava posvećena je projektovanju PDT polja i analizi uticaja topologije i izabrane metrike na oblik funkcije  $\Gamma(\alpha)$ . Biće razmotren uticaj karakterističnih 1D i 2D topologija na oblik funkcije  $\Gamma(\alpha)$ . Na primeru  $3^6$  2D polja biće razmotrena razlika primene Hamingove i Euklidske metrike.

# Glava 6

## Primeri primene parcijalno visokopouzdanih sistema

U ovoj glavi dati su primeri primene tehnike za projektovanje parcijalno visokopouzdanih polja. Na početku glave detaljno će biti dat tok projektovanja parcijalno visokopouzdanog jednosmernog i dvosmernog 1D polja za množenje matrice i vektora, predstavljenih u poglavlju 2.3.4. Za oba 1D polja biće analizirani tipovi grešaka koji se mogu javiti, i u zavisnosti od tipa greške biće izvedena analiza propagacije greške kroz polje. Drugi deo glave posvećen je projektovanju parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva. Projektovanje heksagonalnog PDT polja biće izvedeno za Hamingov i Euklidov tip otkaza sistema. Poslednji deo glave sadrži postupak projektovanja i analizu parcijalno visokopouzdanih sistema sa parcijalno visokopouzdanim podsistemima. Za sve primere projektovanja polja biće data analiza isplativosti primene PDT tehnike projektovanja.

### 6.1 Parcijalno visokopouzdana 1D polja

Diskretna kosinusna transformacija, kako je u poglavlju 2.1.4 rečeno, predstavlja transformaciju signala iz prostornog u frekventni domen. DCT transformaciju, datu definicijom 2.6, moguće je implementirati množenjem matrica, ili višestrukim množenjem matrice i vektora. U ovom poglavlju razmatraćemo implementaciju DCT transformacije na jednom od 1D polja za množenje matrice i vektora predstavljenih u poglavlju 2.3.4.

Kako je u poglavlju 2.1.4 opisano i ilustrovano na slici 2.4, inverzna DCT daće rezultat dovoljno sličan originalu, čak i u slučaju da su samo DC koeficijent i par okolnih koeficijenata tačni. Ovu činjenicu uspešno eksploatišu algoritmi za kompresiju slika i video signala [16].

Činjenica da su u matrici DCT koeficijena najznačajniji koeficijenti DC koeficijent i par okolnih koeficijena može biti iskorišćena i kod projektovanja visokopouzdanih polja za DCT transformaciju. Naime, ukoliko prva ćelija sa leve strane 1D polja za množenje matrice i vektora sa slike 2.18 određuje DC koeficijent, tada je polje moguće projektovati kao parcijalno visokopouzdanost i ostvariti određenu uštedu resursa potrebnih za implementaciju. Eventualna pojava greške neće znatno uticati na rekonstruisanu sliku.

### 6.1.1 Jednosmerno PDT polje za množenje matrice i vektora

U ovom poglavlju biće prikazan postupak projektovanja jednosmernog PDT polja za množenje matrice i vektora. Polje će biti projektovano po algoritmu sa slike 4.17. Za projektovano polje biće izvedena analiza isplativosti. Na kraju poglavlja biće dati rezultati implementacije.

#### Projektovanje polja

Blok dijagram linearnog jednosmernog polja za množenje matrice i vektora prikazan je na slici 2.18. Neka je ukupan broj ćelija polja  $T$ . Označimo sa  $X = [x_i], i = 0, 1, \dots, T - 1$ , elemente ulaznog niza, sa  $B = [b_{i,j}], (i, j) \in \{0, 1, \dots, T - 1\}^2$ , elemente matrice, a sa  $Y = [y_i], i = 0, 1, \dots, T - 1$ , elemente izlaznog niza. Neka binarna reprezentacija elemenata niza  $X$  ima  $n_x$  cifara, kao i reprezentacija elemenata matrice  $b_{i,j}$ , i neka binarna reprezentacija elemenata niza  $Y$  ima  $n_y = 2 \cdot n_x$  cifara. Polje realizuje funkciju

$$Y = BX, \quad y_i = \sum_{k=0}^{T-1} b_{i,k} \cdot x_k, \quad i = 0, 1, \dots, T - 1. \quad (6.1)$$

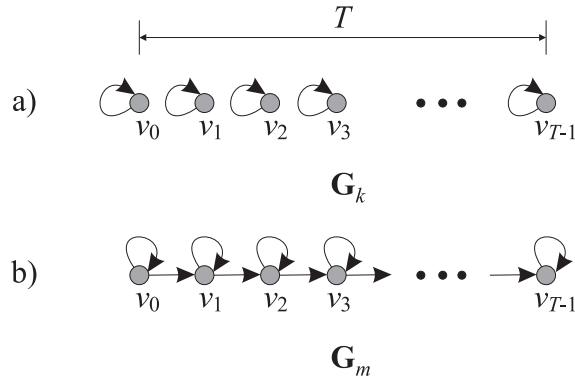
Prvi korak algoritma sa slike 4.17 je izbor minimalne jedinice zamene i izbor metrike otkaza sistema. Neka je minimalna jedinica zamene jedna ćelija polja sa slike 2.18.

Za izabranu jedinicu zamene, po definiciji savršene pouzdanosti (def. 3.8),  $i$ -ta ćelija polja ispravna je samo ukoliko su svi bitovi rezultata  $y_i$  ispravni. Kao rezultat izvršenja DSP algoritma na arhitekturi posmatramo niz izlaza  $Y = [y_i]$ . Elementi ovog niza nemaju međusobnu težinsku zavisnost, pa u ovom slučaju nije moguće primeniti Euklidsku metriku. Tako, izbor metrike definisan je uvedenim projektnim zahtevima. Izabrana metrika otkaza za ovaj primer je Hamingova metrika (def. 4.3).

Radi jednostavnosti, analiziraćemo jednostruke greške koje se mogu javiti u ćeliji. S jedne strane, greška se može javiti u kombinacionoj mreži ćelije. Ovakve greške su, na osnovu funkcije ćelije (slika 2.18), stacionarne za ćeliju i ne propagiraju se dalje kroz polje. Graf sa putevima propagacije grešaka ovog tipa ( $\mathbf{G}_k$ ), arhitekture sa slike 2.18, prikazan je na slici 6.1a, gde čvor grafa  $v_i$  odgovara  $i$ -toj ćeliji polja sa slike 2.18.

Projektovanje PDT arhitekture za ovaj tip grešaka je trivijalno, jer je matrica susedstva grafa  $\mathbf{G}_k$  jedinična matrica ( $I$ ), a samim tim je i tranzitivno zatvaranje  $A^*(\mathbf{G}_k) = I$ . Kako u ovom slučaju nema propagacije iz čvora u čvor, arhitektura  $\mathbf{P}_{DT}(i)$  jednaka je osnovnoj, sa  $i$ -tom visokopouzdanom ćelijom.

Sa druge strane, greška se može javiti na memorijskim elementima za pamćenje vrednosti elemenata  $x_i$ , koji se nalaze između ćelija i omogućavaju protočnost elemenata ulaznog niza  $X$  (slika 2.18). Graf propagacije grešaka ovog tipa, u oznaci  $\mathbf{G}_m$ , arhitekture sa slike 2.18, prikazan je na slici 6.1b.



Slika 6.1: Grafovi mogućih propagacija grešaka kroz jednosmerno 1D polje za množenje matrice i vektora sa slike 2.18: a) za greške u kombinacionoj mreži, b) za greške na memorijskim elementima

Matrica susedstva grafa  $\mathbf{G}_m$  sa slike 6.1b je [99]

$$A = \begin{bmatrix} 1 & 1 & \cdots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \\ 0 & 0 & & 1 & 1 \\ 0 & 0 & & 0 & 1 \end{bmatrix}_{T \times T}, \quad (6.2)$$



sa elementima  $a_{i,j}$  oblika

$$a_{i,j} = \begin{cases} 1, & i + 1 \geq j \geq i \\ 0, & \text{inače} \end{cases} .$$

**Teorema 6.1** *Opšti oblik matrice tranzitivnog zatvaranja grafa sa slike 6.1b je [99]*

$$A^*(\mathbf{G}_m) = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ 0 & 1 & & 1 & 1 \\ \vdots & & \ddots & & \\ 0 & 0 & & 1 & 1 \\ 0 & 0 & & 0 & 1 \end{bmatrix}_{T \times T}, \quad (6.3)$$

odnosno, elementi matrice su oblika

$$a_{i,j} = \begin{cases} 1, & i \leq j \\ 0, & \text{inače} \end{cases} .$$

**Dokaz** Da bi dokazali tvrđenje teoreme, krenućemo od matrice susedstva grafa  $\mathbf{G}_m$  date sa (6.2), koja opisuje sve puteve dužine jedan u grafu. Puteve dužine 2, na osnovu leme 4.5, dobijamo kvadriranjem matrice susedstva  $A$ . Elementi matrice  $A^2$  su oblika

$$a_{i,j}^{(2)} = \begin{cases} 1, & i + 2 \geq j \geq i \\ 0, & \text{inače} \end{cases} .$$

Da važi

$$a_{i,j}^{(k)} = \begin{cases} 1, & i + k \geq j \geq i \\ 0, & \text{inače} \end{cases} \quad (6.4)$$

pokazaćemo matematičkom indukcijom. Za  $k = 1$  izraz (6.4) postaje opšti oblik elemenata matrice susedstva (6.2). Stepenuvanjem (6.4) dobijamo

$$a_{i,j}^{(k+1)} = \begin{cases} 1, & i + k + 1 \geq j \geq i \\ 0, & \text{inače} \end{cases}, \quad (6.5)$$

što potvrđuje izraz (6.4).

Kako je maksimalna dužina puteva u grafu jednaka broju čvorova  $T$ , za tranzitivno zatvaranje važi  $A^*(\mathbf{G}_m) = A^{(T)}$ , pa je

$$a_{i,j}^* = a_{i,j}^{(T)} = \begin{cases} 1, & i + \mathbf{T} \geq j \geq i \\ 0, & \text{inače} \end{cases} = \begin{cases} 1, & j \geq i \\ 0, & \text{inače} \end{cases},$$

Tvrđenje teoreme dobija se iz tvrđenja leme 4.6 i oblika matrice  $A^i$ ,  $i = 1, 2, \dots, T$ .  $\square$

Da bi elemente matrice tranzitivnog zatvaranja  $(a_{i,j})$ , opisane izrazom (6.3), prostorno uredili tako da poziciono odgovaraju čvorovima grafa, potrebno je  $\eta$ -tu kolonu preslikati na vektor mape uticaja greške  $M_\eta = [m_{i,j}^\eta]$ ,  $\eta = 0, 1, \dots, T-1$ . Funkcija ovakvog prostornog uređenja je

$$f_0 : V \rightarrow \mathbf{N}^2, \quad (p, q) = f_0(v_i),$$

gde je  $p = c^{ta} = 0$ , a  $q = j$ .

Mape uticaja greške  $M_\eta$ ,  $\eta = 0, 1, \dots, T-1$ , jednosmernog 1D polja za množenje matrice i vektora sa slike 2.18, na osnovu tranzitivnog zatvaranja datog teoremom 6.1 i izabrane funkcije prostornog uređenja su oblika [99]

$$\begin{aligned} M_0 &= [1, 0, 0, 0, \dots, 0] \\ M_1 &= [1, 1, 0, 0, \dots, 0] \\ &\vdots \\ M_{T-1} &= [1, 1, 1, 1, \dots, 1]. \end{aligned} \quad (6.6)$$

Skup čvorova  $\mathbf{P}_{DT}(\alpha)$ , na osnovu (4.4), sadrži  $\alpha$  čvorova, odnosno

$$\{v_i \in \mathbf{P}_{DT}(\alpha) \mid 0 \leq i \leq \alpha \leq T\}. \quad (6.7)$$

Blok dijagram arhitekture jednosmernog PDT polja za množenje matrice i vektora za  $T = 4$  i  $\alpha = 2$ , sa TMR tehnikom za povećanje pouzdanosti ćelije, prikazan je na slici 6.2.

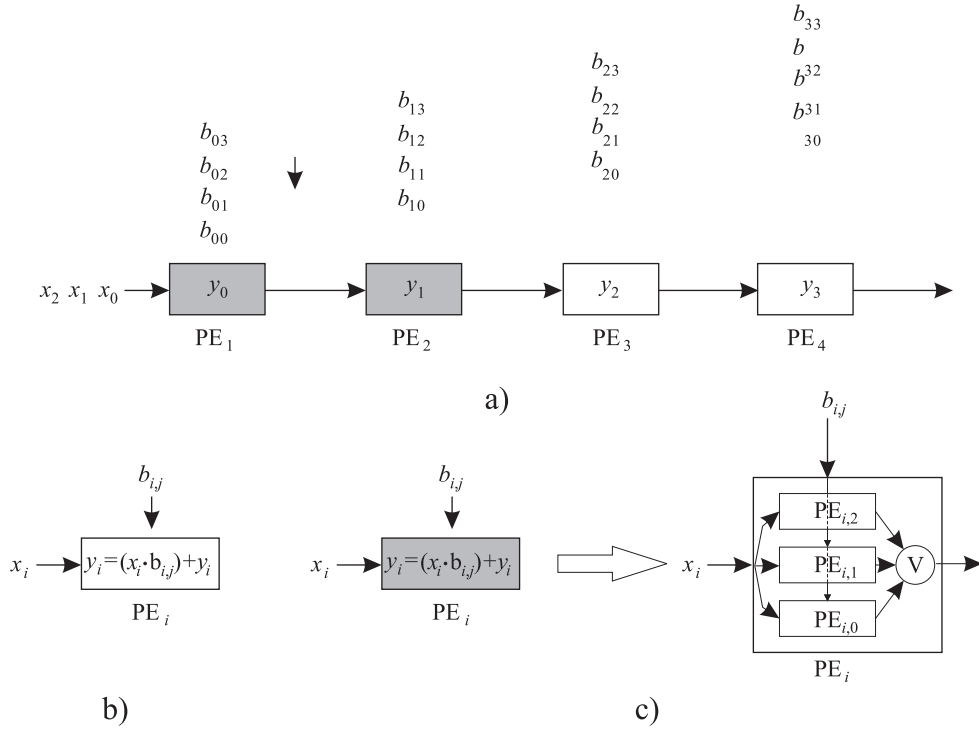
### Analiza prinosa polja

Analiza međusobnog odnosa prinosa ET i PDT pristupa projektovanju, datih izrazima (5.4) i (5.6), zahteva poznavanje veličine netolerantnog dela arhitekture  $\Gamma(\alpha)$ . Funkcija  $\Gamma(\alpha)$  jednosmernog 1D polja za množenje matrice i vektora sa slike 2.18 može se odrediti iz (6.7). Kako skup  $\mathbf{P}_{DT}(\alpha)$  na osnovu (6.7) sadrži tačno  $\alpha$  čvorova, verovatnoća da čvor pripada skupu je

$$\Gamma(\alpha) = \frac{\alpha}{T}. \quad (6.8)$$

Na slici 6.3 dat je uporedni prikaz  $\Gamma(\alpha)$  funkcija 1D jednosmernog polja i 2D polja sa  $4^4$  topologijom iz poglavlja 5.3 (slika 5.6). Sa slike 6.3, kao i iz jednačine (6.8), vidi se linearna promena vrednosti funkcije  $\Gamma(\alpha)$  u zavisnosti od argumenta  $\alpha$ . Takođe važi

$$\forall \alpha, \quad 0 \leq \alpha \leq T \Rightarrow \Gamma(\alpha)_{1D} \leq \Gamma(\alpha)_{4^4}.$$



Slika 6.2: Arhitektura parcijalno visokopouzdanog jednosmernog 1D polja za množenje matrice i vektora za  $T = 4$  i  $\alpha = 2$ : a) polje za množenje matrice i vektora, b) osnovna ćelija, c) visokopouzdana ćelija

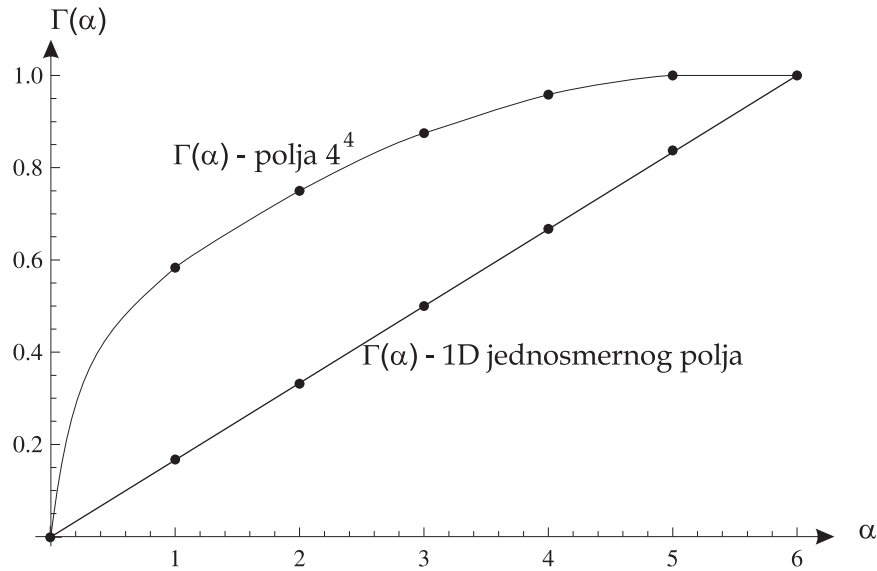
Ovo znači da za isti stepen tolerancije na greške 1D polje ima bolje karakteristike u odnosu na ortogonalno 2D polje u pogledu primene PDT tehnike, jer je manji broj ćelija obuhvaćen netolerantnom oblašću.

Razmotrimo promenu verovatnoće pojave defekta za koju se prinosi ET i PDT tehnike seku kod 1D jednosmernog polja. Neka je za projektovanje PDT jednosmernog 1D polja za množenje matrice i vektora u cilju povećanja pouzdanosti ćelije izabrana SC3 tehnika.

**Teorema 6.2** *Verovatnoća ( $p$ ) počev od koje je prinos PDT jednosmernog 1D polja sa SC3 ćelijama u netolerantnom delu veći u odnosu na prinos ET polja je*

$$p = \frac{1}{2} \left( \sqrt{4 \left(1 + 2\frac{\alpha}{T}\right)^{1/\alpha} - 3} - 1 \right). \quad (6.9)$$

**Dokaz** Tvrdjenje teoreme dobija se zamenom izraza (6.8) u (5.9).  $\square$



Slika 6.3: Usporedni prikaz  $\Gamma(\alpha)$  funkcija 1D jednosmernog polja i 2D polja sa  $4^4$  topologijom

Ukoliko je za povećanje pouzdanosti ćelija PDT polja izabrana TMR tehnika (slika 6.2) važi tvrđenje sledeće teoreme.

**Teorema 6.3** *Opseg verovatnoća ( $p$ ) za koje je prinos PDT jednosmernog 1D polja sa TMR ćelijama u netolerantnom delu veći u odnosu na prinos ET polja je*

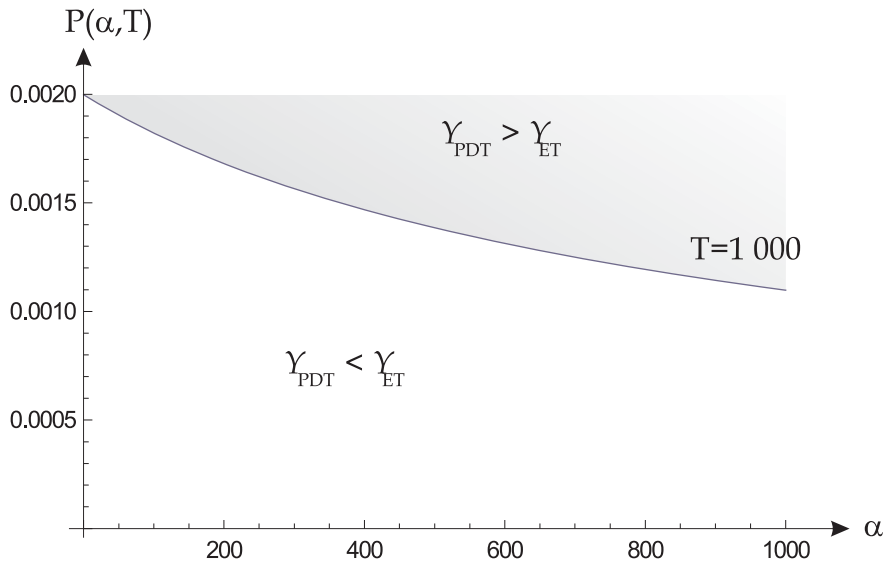
$$p \in [p_1, p_2], \quad p_{1,2} = \frac{1}{4} \left( 1 \pm \sqrt{9 - 8 \left( 1 + 2 \frac{\alpha}{T} \right)^{1/\alpha}} \right). \quad (6.10)$$

**Dokaz** Tvrđenje teoreme dobija se zamenom izraza (6.8) u (5.15).  $\square$

Na slici 6.4 prikazan je grafik funkcije (6.9) za promenu parametra  $\alpha$  u opsegu  $\alpha = 0, 1, \dots, T$ , za  $T = c^{ta} = 1000$ . Za razliku od funkcije 2D polja topologije  $4^4$ , date na slici 5.7, funkcija prikazana na slici 6.4 je u konstantnom opadanju. Ovo znači da će PDT tehnika dati veći prinos za manju verovatnoću pojave defekta ukoliko je sistem manje tolerantan na greške.

## Rezultati FPGA implementacije polja

U cilju ilustracije kompromisa između ET i PDT jednosmernog 1D polja za množenje matrice i vektora u pogledu zauzeća resursa, PDT polje sa slike



Slika 6.4: Promena verovatnoće pojave defekta za koju se prinosi ET i PDT tehnike seku kod 1D jednosmernog polja

6.2, korišćenjem Xilinx ISE 8.2 WebPack razvojnog okruženja, opisano je u VHDL-u i implementirana na Xilinx Spartan2 XC2S200 FPGA čipu [32]. Konstantama su u VHDL-u definisani parametri arhitekture ( $n_x$ ,  $n_y$ ,  $T$  i  $\alpha$ ), tako da je sintezu moguće izvršiti za proizvoljne dimenzije polja.

Parcijalna visoka pouzdanost ugrađena je u VHDL opis kroz implementaciju funkcije pridružene arhitekturi, koja u zavisnosti od zadatog parametra  $\alpha$  određuje i vraća mapu uticaja greške arhitekture (6.6). Formirana VHDL funkcija vraća vektor kod koga svaki element odgovara jednoj ćeliji polja. Ukoliko je vrednost elementa jednaka 1, na mestu te ćelije instancira se komponenta opisana u VHDL-u kao visokopouzdan sistem (slika 6.2). Ukoliko je odgovarajući element matrice 0, na mesto ćelije vrši se instanciranje komponente osnovne ćelije filtra opisane takođe u VHDL-u.

U tabeli 6.1 dati su rezultati implementacije jednosmernog 1D PDT polja za množenje matrice i vektora sa SC3 tehnikom za povećanje pouzdanosti ćelije. Tabela 6.2 sadrži rezultate implementacije PDT polja sa TMR tehnikom za povećanje pouzdanosti ćelije. Polje je za slučajeve primene SC3 i TMR tehnike implementirano za  $T = \{4, 8, 16\}$ , i za širinu ulaznih reči  $n_x = 8$ . Za navedene dimenzije polja implementacija je izvršena za različite veličine netolerantne oblasti i to 25%, 50% i 75% od ukupne veličine polja, odnosno za  $\gamma = \{0.25, 0.50, 0.75\}$ . Polja u koloni  $\Gamma(\alpha) = 0$  su osnovna polja, bez visokopouzdanih ćelija, i vrednosti date u ovim kolonama identične su

T	$n_x$	$\Gamma(\alpha)=0$			0.25		0.50		0.75		1.0	
		Broj ćelija	Impl. [kG]	Takt [ns]	Broj ćelija	Impl. [kG]	Broj ćelija	Impl. [kG]	Broj ćelija	Impl. [kG]	Broj ćelija	Impl. [kG]
4	8	4	9,5	15,0	6	12,8	8	16,8	10	19,3	12	24,3
8	8	8	17,8	16,7	12	24,1	16	30,6	20	37,0	24	40,2
16	8	16	36,6	20,0	24	48,4	32	/	40	/	48	/

Tabela 6.1: Rezultati implementacije jednosmernog 1D PDT polja za množenje matrice i vektora sa SC3 tehnikom za povećanje pouzdanosti ćelija

implementacionim rezultatima osnovnog polja datim u tabeli 2.3.

Za svako implementirano PDT polje dat je ukupan broj ćelija polja i zauzeće resursa izraženo u [kG]. Na primer, za PDT polje sa SC3 ćelijama, za koje je  $T = 8$  i  $\Gamma(\alpha) = 0.25$ , ukupan broj ćelija (tabela 6.1) je 12. Ova vrednost dobija se tako što je umesto 25% osnovnih ćelija instancirana triplicirana SC3 ćelija, pa je  $(8 \cdot 0.25) \cdot 3 + (8 \cdot 0.75) \cdot 1 = 12$ . Ovo polje na čipu zauzima 24.1kG (tabela 6.1).

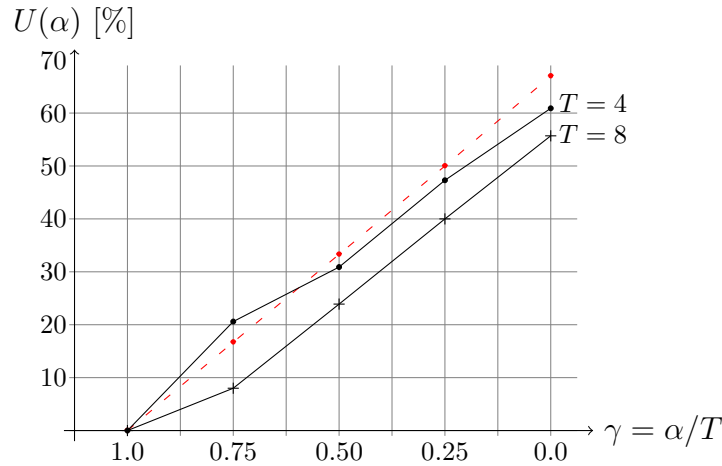
Trajanje taktnog intervala približno je isto kod svih implementiranih polja, pa su u cilju ilustracije reda veličine trajanja taktnog intervala u tabelama date vrednosti samo za osnovna polja  $\Gamma(\alpha) = 0$ . Ove vrednosti jednake su vrednostima datim u tabeli 2.3.

Na slici 6.5 grafički su prikazani rezultati iz tabele 6.1 za  $T = 4$  i  $T = 8$ . Rezultati su predstavljeni na isti način na koji su predstavljeni rezultati dati na slici 4.23, t.j. korišćenjem izraza (4.17), koji određuje relativnu uštedu  $U(\alpha)$  PDT polja  $\mathbf{P}_{DT}(\alpha)$ , u odnosu na polje  $\mathbf{P}_{DT}(L)$ . Isprekidanom linijom na slici 6.5 predstavljeni su numerički rezultati iz kolone "Broj ćelija" tabela 6.1 i 6.2. Relativni odnosi svih ovih vrednosti su po (4.17) isti, pa su ovi rezultati na slici predstavljeni jednom pravom. Rezultati implementacije dati na slici 6.5 ilustruju teoretske rezultate prikazane na slici 6.3.

U narednom poglavlju dat je postupak projektovanja i analiza prinosa PDT dvosmernog 1D polja za množenje matrice i vektora.

T	$n_x$	$\Gamma(\alpha)=0$			0.25		0.50		0.75		1.0	
		Broj ćelija	Impl. [kG]	Takt [ns]	Broj ćelija	Impl. [kG]	Broj ćelija	Impl. [kG]	Broj ćelija	Impl. [kG]	Broj ćelija	Impl. [kG]
4	8	4	9,5	15,0	6	12,6	8	15,7	10	18,8	12	22,0
8	8	8	17,8	16,7	12	23,8	16	29,8	20	35,8	24	38,8
16	8	16	36,6	20,0	24	48,2	32	57,9	40	/	48	/

Tabela 6.2: Rezultati implementacije jednosmernog 1D PDT polja za množenje matrice i vektora sa TMR tehnikom za povećanje pouzdanosti ćelija



Slika 6.5: Grafik uštede resursa potrebnih za implementaciju 1D jednosmernog PDT polja za množenje matrice i vektora

### 6.1.2 Dvosmerno PDT polje za množenje matrice i vektora

U ovom poglavlju predstavljeno je dvosmerno polje za množenje matrice i vektora. Postupak projektovanja i analiza polja biće prikazani na isti način kao kod polja projektovanog u prethodnom poglavlju.

#### Projektovanje polja

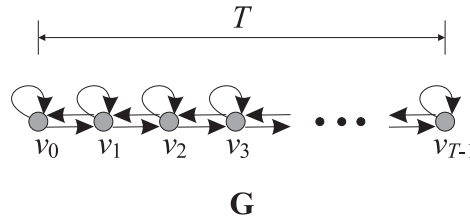
Blok dijagram linearnog dvosmernog polja za množenje matrice i vektora prikazan je na slici 2.19, a tok podataka kroz polje dat je na slici 2.20. Označimo sa  $X = [x_i], i = 0, 1, \dots, L - 1$ , elemente ulaznog niza, sa  $B = [b_{i,j}], i, j \in \{0, 1, \dots, L - 1\}$ , elemente matrice, a sa  $Y = [y_i], i = 0, 1, \dots, L - 1$ , elemente izlaznog niza. Tada je ukupan broj ćelija polja sa slike 2.19 jednak  $T = 2L - 1$ . Neka binarna reprezentacija elemenata niza  $X$  ima  $n_x$  cifara, kao i reprezentacija elemenata matrice  $b_{i,j}$ , i neka binarna reprezentacija elemenata niza  $Y$  ima  $n_y = 2 \cdot n_x$  cifara.

Prvi korak algoritma sa slike 4.17 je izbor minimalne jedinice zamene i izbor metrike otkaza sistema. Neka je minimalna jedinica zamene, kao u primeru iz prethodnog poglavlja, jedna ćelija polja sa slike 2.19, i neka je  $i$ -ta ćelija polja ispravna je samo ukoliko su svi bitovi rezultata  $y_i$  ispravni. Elementi izlaznog niza  $Y$  nemaju međusobnu težinsku zavisnost, pa je izabrana metrika otkaza za ovaj primer Hamingova metrika (def. 4.3).

Radi jednostavnosti, analiziraćemo jednostruke greške koje se mogu javiti

u ćeliji. S jedne strane, greška se može javiti u kombinacionoj mreži ćelije. Kako je svaka ćelija polja uključena u izračunavanje svakog elementa izlaznog vektora (slika 2.20), usled defekta u kombinacionoj logici bilo koje ćelije svi elementi izlaznog vektora  $y_i$  biće pogrešni. Graf propagacije  $\mathbf{G}$  za ovaj tip greške dat je na slici 6.6.

Sa druge strane, greška se može javiti na memorijskim elementima za pamćenje vrednosti elemenata  $x_i$ , koji se nalaze između ćelija i omogućavaju protočnost elemenata ulaznog niza  $X$  u polju (slika 2.19). Greška ovog tipa propagira se u desno, kao kod primera iz prethodnog poglavlja, ali zbog povratnog puta dodatno se propagira i u levo (slika 2.20). Graf propagacije grešaka ovog tipa, jednak je grafu  $\mathbf{G}$  sa slike 6.6.



Slika 6.6: Graf propagacije grešaka kroz dvosmerno 1D polje za množenje matrice i vektora sa slike 2.19

Matrica susedstva grafa  $\mathbf{G}$  sa slike 6.6 je [99]

$$A = \begin{bmatrix} 1 & 1 & \cdots & 0 & 0 \\ 1 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \\ 0 & 0 & & 1 & 1 \\ 0 & 0 & & 1 & 1 \end{bmatrix}_{T \times T}, \quad (6.11)$$

sa elementima  $a_{i,j}$  oblika

$$a_{i,j} = \begin{cases} 1, & i - 1 \leq j \leq i + 1 \\ 0, & \text{inače} \end{cases}.$$

**Teorema 6.4** *Opšti oblik matrice tranzitivnog zatvaranja grafa sa slike 6.6 je*

$$A^*(\mathbf{G}) = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & & 1 & 1 \\ \vdots & & \ddots & & \\ 1 & 1 & & 1 & 1 \\ 1 & 1 & & 1 & 1 \end{bmatrix}_{T \times T}, \quad (6.12)$$



odnosno, elementi matrice tranzitivnog zatvaranja su oblika

$$a_{i,j} = 1, \quad \forall (i, j) \in \{0, 1, \dots, T-1\}^2.$$

**Dokaz** Da bi dokazali tvrđenje teoreme, krenućemo od matrice susedstva grafa  $\mathbf{G}$  date sa (6.11). Puteve dužine 2, na osnovu leme 4.5, dobijamo kvadriranjem matrice susedstva  $A$ . Elementi matrice  $A^2$  su oblika

$$a_{i,j}^{(2)} = \begin{cases} 1, & i-2 \leq j \leq i+2 \\ 0, & \text{inače} \end{cases},$$

odnosno matrici se dodaje još jedna kvazi-dijagonala sa jedinicama ispod i iznad glavne dijagonale.

Da važi

$$a_{i,j}^{(k)} = \begin{cases} 1, & i-k \leq j \leq i+k \\ 0, & \text{inače} \end{cases} \quad (6.13)$$

pokazaćemo matematičkom indukcijom. Za  $k=1$  izraz (6.13) postaje opšti oblik elemenata matrice susedstva (6.2). Pretpostavimo da jednakost (6.13) važi za  $k$ . Stepenovanjem (6.13) dobijamo

$$a_{i,j}^{(k+1)} = \begin{cases} 1, & i-(k+1) \leq j \leq i+(k+1) \\ 0, & \text{inače} \end{cases}, \quad (6.14)$$

što potvrđuje izraz (6.13).

Kako je maksimalna dužina puteva u grafu jednaka broju čvorova  $T$ , tvrđenje teoreme sledi iz leme 4.6 i oblika matrica  $A^i$ ,  $i=1, 2, \dots, T$ .  $\square$

Kako su svi elementi tranzitivnog zatvaranja (6.12) jednaki 1, sem za  $\alpha=0$ , skup  $\mathbf{P}_{DT}(\alpha)$ ,  $0 < \alpha \leq T$ , sadrži sve čvorove grafa, odnosno

$$\forall v_i \in \mathbf{G} \Rightarrow v_i \in \mathbf{P}_{DT}(\alpha), \quad 0 < \alpha \leq T, \quad i=0, 1, \dots, T-1. \quad (6.15)$$

Ovo znači da ukoliko se želi postići pouzdanost bar jednog elementa izlaznog niza, sve ćelije polja je potrebno projektovati kao visokopouzdana.

### Analiza prinosa polja

Na osnovu (6.15), veličinu netolerantne oblasti dvosmernog 1D polja možemo opisati funkcijom

$$\Gamma(\alpha) = \begin{cases} 0, & \alpha = 0 \\ 1, & \text{inače} \end{cases}. \quad (6.16)$$

Iz izraza (6.16) i funkcija  $\Gamma(\alpha)$  1D jednosmernog PDT polja i 2D  $4^4$  polja (slika 6.3), važi da je

$$\forall \alpha, 0 \leq \alpha \leq T \Rightarrow \Gamma(\alpha)_{1D-jedn.} \leq \Gamma(\alpha)_{4^4} \leq \Gamma(\alpha)_{1D-dvosm.},$$

odnosno da za isti stepen tolerancije na greške dvosmerno 1D polje ima lošije karakteristike od ostalih polja u pogledu primene PDT tehnike, jer je za isto  $\alpha$  veći broj ćelija obuhvaćen netolerantnom oblašću. Jednakost važi samo za granične slučajeve  $\alpha = 0$  i  $\alpha = T$ .

Razmotrimo promenu verovatnoće pojave defekta za koju se prinosi ET i PDT tehnike seku kod 1D dvosmernog polja, kada je u cilju povećanja pouzdanosti ćelije izabrana SC3 tehnika.

**Teorema 6.5** *Verovatnoća ( $p$ ) počev od koje je prinos dvosmernog 1D PDT polja sa SC3 ćelijama u netolerantnom delu veći u odnosu na prinos ET polja je*

$$p = \frac{1}{2} \left( \sqrt{4(1+3)^{1/T} - 3} - 1 \right). \quad (6.17)$$

**Dokaz** Tvđenje teoreme dobija se zamenom izraza (6.16) u (5.9).  $\square$

Za razliku od prethodno projektovanih polja, kod dvosmernog 1D polja verovatnoća za koju se prinosi ET i PDT polja seku ne zavisi od parametra  $\alpha$  (slike 5.7 i 6.4). Promena verovatnoće  $p$  u (6.17) u funkciji od  $T$  data je za opšti slučaj na slici 5.3.

## Rezultati FPGA implementacije polja

Arhitektura dvosmernog 1D PDT polja za množenje matrice i vektora opisana je u VHDL-u i implementirana na Xilinx Spartan2 XC2S200 FPGA čipu [32]. Konstantama su u VHDL-u definisani parametri arhitekture ( $n_x, n_y, T$  i  $\alpha$ ), tako da je sintezu moguće izvršiti za proizvoljne dimenzije polja.

U tabeli 6.3 dati su rezultati implementacije dvosmernog 1D PDT polja za množenje matrice i vektora sa SC3 tehnikom za povećanje pouzdanosti ćelije. Tabela 6.4 sadrži rezultate implementacije PDT polja sa TMR tehnikom za povećanje pouzdanosti ćelije. Rezultati implementacije polja  $\Gamma(\alpha) = 0$  jednake su rezultatima implementacije osnovnog polja datim u tabeli 2.4.

U narednom poglavlju dat je postupak projektovanja i analiza prinosa parcijalno visokopouzdanog 2D  $3^6$  polja za množenje brojeva.

T	$n_x$	$\Gamma(\alpha)=0$			$0 < \Gamma(\alpha) < 1$		1.0	
		Broj ćelija	Impl. [kG]	Takt [ns]	Broj ćelija	Impl. [kG]	Broj ćelija	Impl. [kG]
3	8	3	5,2	16,3	9	12,5	9	12,5
7	8	7	10,7	14,5	21	25,4	21	25,4
15	8	15	23,5	15,2	45	-	45	-

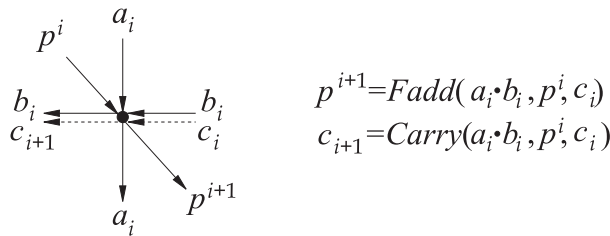
Tabela 6.3: Rezultati implementacije dvosmernog 1D PDT polja za množenje matrice i vektora sa SC3 tehnikom za povećanje pouzdanosti ćelija

T	$n_x$	$\Gamma(\alpha)=0$			$0 < \Gamma(\alpha) < 1$		1.0	
		Broj ćelija	Impl. [kG]	Takt [ns]	Broj ćelija	Impl. [kG]	Broj ćelija	Impl. [kG]
3	8	3	5,2	16,3	9	12,7	9	12,7
7	8	7	10,7	14,5	21	26,3	21	26,3
15	8	15	23,5	15,2	45	-	45	-

Tabela 6.4: Rezultati implementacije dvosmernog 1D PDT polja za množenje matrice i vektora sa TMR tehnikom za povećanje pouzdanosti ćelija

## 6.2 Projektovanje parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva

U ovom poglavlju biće dat postupak projektovanja i analiza parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva prikazanog na slici 2.22. Neka polje ima  $N$  vrsta i kolona, i ukupno  $T = N^2$  ćelija, među kojima je  $L = 2N$  izlaznih (izlazi  $p_i, i = 0, 1, \dots, L - 1$ , slika 2.22). Neka je minimalna jedinica zamene jedna ćelija polja sa slike 2.22. Ćelija polja prikazana je na slici 6.7.



Slika 6.7: Ćelija heksagonalnog polja za množenje brojeva sa slike 2.22

Metrika otkaza sistema može biti Hamingova (def. 4.3), ili Euklidska (def. 4.2). Po tvrdnji leme 4.1 Hamingova metrika je stroža, pa sistem ispravan

po Hamingovoj, sigurno je ispravan i po Euklidskoj metrici. U narednom poglavlju razmotrićemo projektovanje 2D PDT polja za množenje brojeva sa Hamingovom metrikom otkaza.

### 6.2.1 Primer primene Hamingove metrike

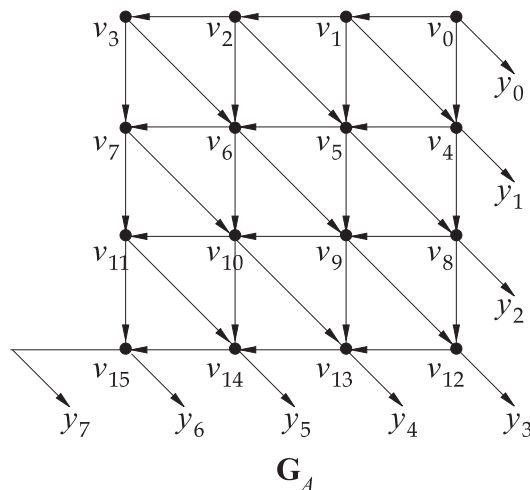
Hamingova metrika razmatra da li se odgovarajući bit rezultata razlikuje od tačnog ili ne, bez ulaženja u detalje o težinskom uticaju posmatranog bita na ukupnu grešku. Problem težinskog uticaja biće razmotren u narednom poglavlju na primeru Euklidske metrike.

Ćelija  $3^6$  polja prikazana na slici 6.7 ima tri ulazna i tri izlazna pravca toka signala, sa ukupno četiri ulazna i četiri izlazna toka podataka. Za ulaze  $a_i, b_i, p^i$  i  $c_i$ , ćelija daje dva bita rezultata: proizvod  $p^{i+1}$  i prenos  $c_{i+1}$ , i prenosi dalje ulaze  $a_i$  i  $b_i$  (slika 6.7). Radi jednostavnosti, analiziraćemo jednostruke greške koje se mogu javiti u ćeliji.

Na osnovu funkcije ćelije sa slike 6.7, greška na ulazima  $a_i$  ili  $b_i$  izazvaće greške na najviše izlaznih tokova podataka.

**Definicija 6.1 (Greške tipa "A")** Greške na ulazu  $a_i$ , koje izazivju greške na izlazima  $a_i, p^{i+1}$  i  $c_{i+1}$ , odnosno, koje se propagiraju po svim izlaznim pravcima toka signala nazivamo greške tipa "A".

Graf  $G_A$  propagacije grešaka ovog tipa prikazan je na slici 6.8.



Slika 6.8: Graf propagacije grešaka tipa "A" kroz 2D  $3^6$  polje za množenje brojeva

Bez određivanja matrice susedstva, na osnovu grafa  $\mathbf{G}_A$  sa slike 6.8 može se zaključiti sledeće:

**Lema 6.1** *Za elemente poslednje kolone matrice tranzitivnog zatvaranja  $A^*(\mathbf{G}_A)$ , koja se odnosi na puteve do čvora  $v_{T-1}$  grafa  $\mathbf{G}_A$  sa slike 6.8, važi*

$$a_{T-1,j}^* = 1, \quad j = 0, 1, \dots, T-1. \quad (6.18)$$

**Dokaz** Za čvorove  $v_i$  i  $v_{i+1}$  važi da je grana  $(v_i, v_{i+1}) \in \mathbf{G}_A$ ,  $i = 0, 1, \dots, T-2$ , sem za poslednje čvorove u svakoj vrsti za koje je  $(i+1) \bmod N = 0$ . Vrste su povezane granama  $(v_{i \cdot N+j}, v_{(i+1) \cdot N+j}) \in \mathbf{G}_A$  za  $\forall i, j \in \{0, 1, \dots, N-1\}$ .

Na osnovu ovoga sledi da do poslednjeg čvora  $v_{T-1}$  postoje putevi iz svakog čvora poslednje vrste, a preko grana koje povezuju vrste i iz svih ostalih čvorova grafa, čime je dokazano tvrđenje leme.  $\square$

**Teorema 6.6** *Kod heksagonalnog  $P_{DT}(\alpha)$  polja, bez obzira na  $\alpha$ , za tip grešaka "A" sve ćelije su visokopouzdanae.*

**Dokaz** Neka je funkcija prostornog uređenja  $f_o : V \rightarrow \mathbf{N}^2$ ,  $(p, q) = f_o(v_i)$ , data sa

$$p = \left\lfloor \frac{i_v}{N} \right\rfloor, \quad q = N - (i_v \bmod N) - 1, \quad (6.19)$$

gde je  $i_v = 0, 1, \dots, T-1$ , indeks vrste matrice tranzitivnog zatvaranja. Mapa uticaja greške na najviši bit rezultata  $M_{L-1}$  je na osnovu (6.18)

$$M_{L-1} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & & 1 \\ \cdots & & \ddots & \\ 1 & 1 & & 1 \end{bmatrix}. \quad (6.20)$$

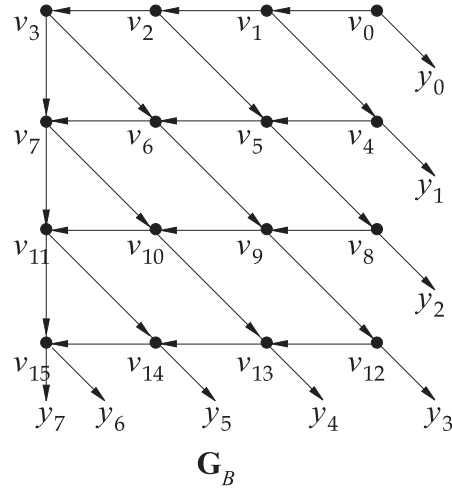
Kako je za svako  $\alpha \geq 1$  u skup  $P_{DT}(\alpha)$  uključen podskup  $\mathbf{M}_{L-1}$  (def. 4.8), na osnovu tvrđenja leme 4.4 važi

$$\forall v_i \Rightarrow v_i \in P_{DT}(\alpha), \quad \alpha \geq 1, \quad i = 0, 1, \dots, T-1,$$

što potvrđuje teoremu.  $\square$

Razmotrimo sa aspekta Hamingove metrike otkaza propagaciju grešaka usled defektne kombinacione mreže ćelije.

**Definicija 6.2** *Jednostruke defekte, koji prouzrokuju pojavu greške na izlazu  $p^{i+1}$  ili izlazu  $c_{i+1}$  (slika 6.7), zovemo greške tipa "B".*



Slika 6.9: Graf propagacije grešaka tipa "B" kroz 2D  $3^6$  polje za množenje brojeva za Hamingovu metriku

Na osnovu puteva propagacije signala kroz ćeliju (slika 6.7), graf propagacije grešaka tipa "B" kroz polje ( $\mathbf{G}_B$ ) ima topologiju prikazanu na slici 6.9.

Topologija grafa  $\mathbf{G}_B$  sa slike 6.9 je u osnovi  $4^4$ , sa neregularnostima na levoj strani polja. Graf sa slike 6.9 razlikuje se od grafa sa topologijom  $4^4$  sa slike 4.13 po izboru izlaznih čvorova.

**Lema 6.2** Za elemente poslednje kolone matrice tranzitivnog zatvanja  $A^*(\mathbf{G}_B)$  grafa  $\mathbf{G}_B$  sa slike 6.9 važi

$$a_{T-1,j}^* = 1, \quad j = 0, 1, \dots, T - 1. \quad (6.21)$$

**Dokaz** Za čvorove  $v_i$  i  $v_{i+1}$  važi da je grana  $(v_i, v_{i+1}) \in \mathbf{G}_B$ ,  $i = 0, 1, \dots, T - 2$ , sem za poslednje čvorove u svakoj vrsti za koje je  $(i + 1) \bmod N = 0$ . Vrste su povezane granama iz krajnje levih čvorova (slika 6.9), odnosno  $(v_{(i+1) \cdot N - 1}, v_{(i+2) \cdot N - 1}) \in \mathbf{G}_A$  za  $\forall(i, j) \in \{0, 1, \dots, N - 2\}^2$ .

Na osnovu ovoga sledi da do poslednjeg čvora  $v_{T-1}$  postoje putevi iz svakog čvora poslednje vrste, a preko grana koje povezuju vrste  $i$  iz svih ostalih čvorova grafa, čime je dokazano tvrđenje leme.  $\square$

**Teorema 6.7** Kod heksagonalnog  $P_{DT}(\alpha)$  polja za množenje brojeva, bez obzira na  $\alpha$ , za tip grešaka "B" sve ćelije su visokopouzdanane.

**Dokaz** Dokaz teoreme je identičan dokazu teoreme 6.6.  $\square$

Iz tvrđenja teorema 6.6 i 6.7 sledi da se na heksagonalnom 2D polju za množenje brojeva sa slike 2.22 ne može parcijalno primeniti visoku pouzdanost za Hamingovu metriku, bez obzira na toleranciju polja na greške ( $\alpha$ ).

## 6.2.2 Min-plus algebra

Tranzitivno zatvaranje grafa sadrži informacije o postojanju puteva u grafu, bez razmatranja težine puteva (def. 4.12). Tranzitivno zatvaranje je moguće odrediti *Warshallovim* algoritmom na osnovu matrice susedstva (slika 4.10). Za slučaj primene parcijalne visoke pouzdanosti sa Hamingovom metrikom otkaza nad arhitekturom, imajući u vidu osobine ove metrike, tranzitivno zatvaranje grafa daje sve informacije neophodne za projektovanje PDT arhitekture. Međutim, prilikom projektovanja parcijalno visokopouzdanog sistema sa Euklidskom metrikom otkaza potrebno je razmatrati težinski graf uticaja greške, pa se potrebne informacije o uticajima i propagaciji greške mogu dobiti iz matrice najkraćih puteva (4.15). Matrica najkraćih puteva grafa može se odrediti na osnovu matrice susedstva primenom *Floyd-Warshallovog* algoritma sa slike 4.9.

Nasuprot algoritamskom iteriranju sa slike 4.9, gde se u svakoj iteraciji približava rešenju, algebarsko određivanje matrice najkraćih puteva u algebarskoj strukturi  $(\mathbb{R}, +, \cdot)$  jeste izvodljivo uz poznavanje strukture grafa, ali je znatno komplikovanije. Ilustracije radi, kvadriranjem težinske matrice susedstva dobijamo sumu proizvoda težina grana na putevima dužine dva (lema 4.5, za  $k = 2$ ). Ovo je jednostavno pokazati iz definicije množenja matrica  $C = AxB$ , gde su elementi matrice  $C$  oblika

$$c_{i,j} = \sum_{k=0}^{N-1} a_{i,k} \cdot b_{k,j}. \quad (6.22)$$

Stepenovanjem matrice susedstva, s jedne strane, i određivanjem ukupnog broja puteva, s druge strane, moguće je algebarski odrediti matricu najkraćih puteva regularnih struktura.

Relativno nova oblast matematike, poznata kao "min-plus", ili "tropska algebra" pogodna je za algebarsko određivanje osobina grafa potrebnih za projektovanje parcijalno visokopouzdanih sistema. Naziv "tropska" ovaj oblasti dao je francuski matematičar *Jean-Eric Pin* [100] u čast brazilskom matematičaru *Imre Simonu*, koji je postavio temelje ove oblasti osamdesetih godina prošlog veka [101].

Kod tropske algebre suma dva broja zamenjena je njihovim minimumom,

a proizvod sumom [102]. Tropska algebra je algebarska struktura

$$(\mathbb{R} \cup \infty, \oplus, \odot),$$

gde su operacije  $\oplus$  i  $\odot$  (tropsko sabiranje i množenje, respektivno) definisane kao

$$\begin{aligned} x \oplus y &= \min \{x, y\}, \\ x \odot y &= x + y. \end{aligned} \tag{6.23}$$

Obe operacije su asocijativne [102]

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z \quad \text{i} \quad x \odot (y \odot z) = (x \odot y) \odot z,$$

komutativne [102]

$$x \oplus y = y \oplus x \quad \text{i} \quad x \odot y = y \odot x,$$

a operacija  $\odot$  je distributivna u odnosu na  $\oplus$ , odnosno važi [102]

$$x \odot (y \oplus z) = x \odot y \oplus x \odot z.$$

Neutralni element za sabiranje je beskonačno, a nula za množenje:

$$x \oplus \infty = x \quad \text{i} \quad x \odot 0 = x.$$

Neka su  $A = [a_{i,j}]$  i  $B = [b_{i,j}]$  matrice dimenzija  $N \times K$  i  $K \times M$ , respektivno.

**Definicija 6.3 (Množenje matrica u min-plus algebri)** *Proizvod matrica  $A$  i  $B$ , u min-plus algebri je matrica  $C = AxB$  dimenzija  $N \times M$ , ili kraće  $C = AB$ , čiji su elementi oblika*

$$c_{i,j} = \bigoplus_{k=0}^{K-1} (a_{i,k} \odot b_{k,j}), \tag{6.24}$$

za  $i = 0, 1, \dots, N - 1$  i  $j = 0, 1, \dots, M - 1$  [102].

**Definicija 6.4 (Neutralni element za množenje matrica)** *Neutralni element za operaciju množenja matrica u min-plus algebri je jedinična matrica  $I = [u_{i,j}]$ , sa elementima*

$$u_{i,j} = \begin{cases} 0, & i = j \\ \infty, & i \neq j \end{cases}.$$



**Lema 6.3** *Kvadrat matrice susedstva  $A$  grafa  $\mathbf{G}$  u min-plus algebri je matrica najkraćih puteva dužine dva u grafu  $\mathbf{G}$ .*

**Dokaz** Na osnovu (6.24), drugi stepen matrice  $A^{(2)} = AxA$  je

$$a_{i,j}^2 = \bigoplus_{k=0}^{K-1} (a_{i,k} \odot a_{k,j}),$$

odnosno, u algebri  $(\mathbb{R}, +, \cdot)$

$$a_{i,j} = \min \{a_{i,0} + a_{0,j}, a_{i,1} + a_{1,j}, \dots, a_{i,K-1} + a_{K-1,j}\}$$

što predstavlja minimalni put između čvora  $v_i$  i  $v_j$ , odnosno jednu iteraciju *Floyd-Warshall*ovog algoritma sa slike 4.9 [102].  $\square$

**Lema 6.4** *Matrica najkraćih puteva  $S(\mathbf{G})$ , grafa  $\mathbf{G}$  u min-plus algebri je matrica*

$$S(\mathbf{G}) = \bigoplus_{i=1}^T A^{(i)} = \bigoplus_{i=1}^{\infty} A^{(i)}, \quad (6.25)$$

gde je  $A^{(i)}$   $i$ -ti stepen matrice  $A$  u min-plus algebri, a  $T$  maksimalna dužina puta u grafu  $\mathbf{G}$  [102].

**Dokaz** Na osnovu definicije operacija (6.23), definicije množenja matrica (6.24), tvrđenja leme 6.3 i *Floyd-Warshall*ovog algoritma sa slike 4.9, sledi da je  $i$ -ti stepen matrice  $A$  matrica najkraćih puteva dužine  $i$ .

Najkraći put između čvora  $v_i$  i čvora  $v_j$  je minimalni put od puteva dužine  $l = 1, 2, \dots, T$ , što potvrđuje teoremu.

Kako u grafu ne postoje putevi duži od  $T$ , odnosno  $a_{i,j}^k = \infty, k = T + 1, T + 2, \dots, \infty$ , tropska suma u izrazu (6.25) može se proširiti od  $T$  na  $\infty$  sabiraka.  $\square$

U cilju određivanja mape težinskog uticaja greške sistoličkog polja u min-plus algebri, razmotrićemo neke osobine operacije množenja matrica. Na osnovu definicije 6.3, množenje matrica je asocijativna operacija, odnosno važi

$$Ax(BxC) = (AxB)xC.$$

Neka je matrica  $A = [a_{i,j}]$ , dimenzija  $N \times K$ , predstavljena svojim podmatricama  $A_{p,q}$  kao

$$A = \begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,k-1} \\ A_{1,0} & A_{1,1} & & A_{1,k-1} \\ \vdots & & \ddots & \\ A_{n-1,0} & A_{n-1,1} & & A_{n-1,k-1} \end{bmatrix}_{N \times K},$$

i neka je matrica  $B = [b_{i,j}]$ , dimenzija  $K \times M$ , predstavljena svojim podmatricama  $B_{p,q}$

$$B = \begin{bmatrix} B_{0,0} & B_{0,1} & \cdots & B_{0,m-1} \\ B_{1,0} & B_{1,1} & & B_{1,m-1} \\ \vdots & & \ddots & \\ B_{k-1,0} & B_{k-1,1} & & B_{k-1,m-1} \end{bmatrix}_{K \times M}.$$

Iz definicije množenja matrica može se pokazati da u min-plus algebri važi tvrđenje sledeće leme.

**Lema 6.5** Podmatrica  $C_{p,q}$  matrice  $C = AxB$  jednaka je

$$C_{p,q} = \bigoplus_{i=0}^k A_{p,i} \odot B_{i,q}. \quad (6.26)$$

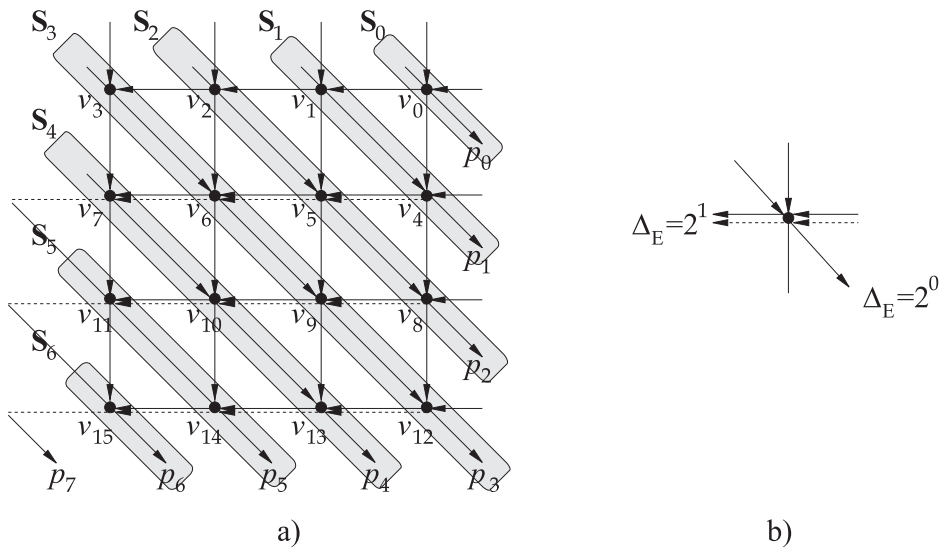
U narednom poglavlju dat je primer projektovanja parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva sa Euklidskom metrikom otkaza.

### 6.2.3 Primer primene Euklidske metrike

Razmotrimo sa aspekta Euklidske metrike otkaza propagaciju grešaka usled defekta na kombinacionoj mreži ćelije, koje mogu prouzrokovati greške na izlaznim bitovima ćelije  $p^{i+1}$  i  $c_{i+1}$  (slika 6.7). Graf propagacije za ovaj tip grešaka prikazan je na slici 6.9. U cilju formiranja grafa propagacije greške za Euklidsku metriku, svakoj grani grafa sa slike 6.9 potrebno je dodeliti težinu.

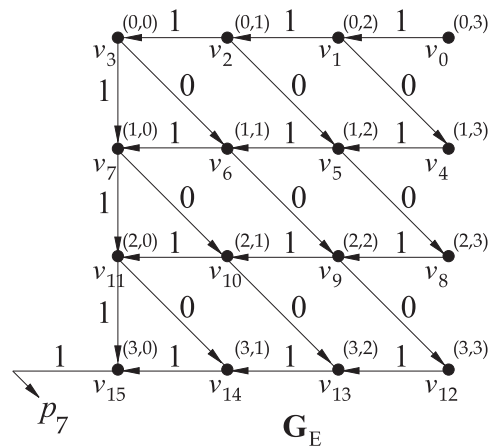
Na slici 6.10a ilustrovan je težinski uticaj grešaka nastalih usled defekta u ćelijama heksagonalnog 2D polja za množenje brojeva. Sivom bojom na slici 6.10a označeni su skupovi čvorova  $\mathbf{S}_k$ ,  $k = 0, 1, \dots, L - 2$ , koji formiraju tokove podataka na kojima se vrši sumiranje parcijalnih proizvoda  $p^i$ . Na primer, u skupu  $\mathbf{S}_2$ , u okviru koga se vrši sumiranje parcijalnih proizvoda izlaznog bita  $p_2$ , nalaze se čvorovi  $v_2, v_5$  i  $v_8$  (slika 6.10a). Kako se na osnovu funkcije ćelije sa slike 6.7 u okviru skupa  $\mathbf{S}_k$  vrši sumiranje parcijalnih proizvoda, greška na izlazu  $p^{i+1}$  čvora  $v_i \in \mathbf{S}_k$  može prouzrokovati grešku  $\Delta_E = \pm 2^0$  na odgovarajućem izlaznom bitu  $p_k$ .

Tok prenosa  $c_i$  (slika 6.7) horizontalan je u odnosu na polje i prenosi se u levo, odnosno na bit veće težine. Na isti način kao za greške na izlazima  $p^i$  može se doći do zaključka da greška ovog tipa ima uticaj  $\Delta_E = \pm 2^1$  u odnosu na odgovarajući bit rezultata. Težine grešaka koje ćelija može da propagira



Slika 6.10: Težinski uticaj grešaka kod heksagonalnog 2D polja za množenje brojeva: a) oblasti propagacije greške, b) težine grešaka na izlazima ćelije

prikazane su na slici 6.10b, a težinski graf propagacije greške  $\mathbf{G}_E = \{V, E\}$  prikazan je na slici 6.11. Za težine grana grafa  $\mathbf{G}_E$  uzeti su eksponenti grešaka sa slike 6.11b.



Slika 6.11: Težinski graf propagacije grešaka kroz heksagonalno 2D polje za množenje brojeva

Graf  $\mathbf{G}_E$  sa slike 6.11 formiran je na osnovu grafa uticaja grešaka tipa "B" sa slike 6.9 i težina grešaka na izlazu ćelija datih na slici 6.10b, odnosno,

za težine  $w_{i,j}$  grafa  $\mathbf{G}_E$  važi

$$\forall e_{i,j} \in \mathbf{G}_B, \quad \Delta_E(v_i \rightarrow v_j) = 2^k \Rightarrow w_{i,j} = k.$$

Matrica susedstva  $A(\mathbf{G}_E)$  težinskog grafa sa slike 6.11 je

$$A(\mathbf{G}_E) = \begin{bmatrix} A_R & A_C & \infty & \cdots & \infty \\ \infty & A_R & A_C & & \infty \\ \infty & \infty & A_R & & \infty \\ \vdots & & & \ddots & \\ \infty & \infty & \infty & & A_R \end{bmatrix}_{T \times T}, \quad (6.27)$$

gde su  $A_R$  i  $A_C$  podmatrice oblika

$$A_R = \begin{bmatrix} 0 & 1 & \infty & \cdots & \infty & \infty \\ \infty & 0 & 1 & & \infty & \infty \\ \infty & \infty & 0 & & \infty & \infty \\ \vdots & & & \ddots & & \\ \infty & \infty & \infty & & 0 & 1 \\ \infty & \infty & \infty & & \infty & 0 \end{bmatrix}_{N \times N} \quad A_C = \begin{bmatrix} \infty & \infty & \infty & \cdots & \infty & \infty \\ 0 & \infty & \infty & & \infty & \infty \\ \infty & 0 & \infty & & \infty & \infty \\ \vdots & & & \ddots & & \\ \infty & \infty & \infty & & \infty & \infty \\ \infty & \infty & \infty & & 0 & 1 \end{bmatrix}_{N \times N}.$$

Da bi odredili opšti oblik mape težinskog uticaja grafa sa slike 6.11 ispitaćemo osobine matrice  $A(\mathbf{G}_E)$  iz (6.27) u min-plus algebri.

Neka je  $A^{(n)} = \bigodot_{i=1}^n A$  tropski  $n$ -ti stepen matrice susedstva  $A(\mathbf{G}_E)$ .

**Lema 6.6** Za podmatricu  $A_{p,q}^{(n)}$ , na poziciji  $(p, q)$  u matrici  $A^{(n)}$  važi

$$A_{p,q}^{(n)} = A_{p,q}^{(n-1)} A_R \oplus A_{p,q-1}^{(n-1)} A_C, \quad (6.28)$$

gde je  $A_{p,q}^{(1)} = A_{p,q}$ .

**Dokaz** Da tvrđenje leme važi pokazaćemo matematičkom indukcijom. Na osnovu (6.26) i (6.27) za  $n = 2$  imamo

$$A \times A = A^{(2)}(\mathbf{G}_E) = \begin{bmatrix} A_R^{(2)} & A_R A_C \oplus A_C A_R & A_C^{(2)} & \cdots & 0 \\ 0 & A_R^{(2)} & A_R A_C \oplus A_C A_R & & 0 \\ 0 & 0 & A_R^{(2)} & & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & & A_R^{(2)} \end{bmatrix}_{T \times T}.$$

Zamenom  $n = 2$  u (6.28) dobijamo podmatrice

$$\begin{aligned} A_{1,1}^{(2)} &= A_{1,1}A_R \oplus A_{1,0}A_C = A_R^{(2)}, \\ A_{1,2}^{(2)} &= A_{1,2}A_R \oplus A_{1,1}A_C = A_C A_R \oplus A_R A_C, \\ A_{1,3}^{(2)} &= A_{1,3}A_R \oplus A_{1,2}A_C = A_C^{(2)}, \\ &\dots \end{aligned}$$

što potvrđuje tačnost izraza (6.28) za  $n = 2$ .

Pretpostavimo da izraz (6.28) važi za  $n = k$ , odnosno

$$A_{p,q}^{(k)} = A_{p,q}^{(k-1)}A_R \oplus A_{p,q-1}^{(k-1)}A_C.$$

Na osnovu tvrđenja leme 6.5 imamo da je

$$A_{p,q}^{(k+1)} = \bigoplus_{i=1}^N \left( A_{p,i}^{(k)} \odot A_{i,q} \right),$$

gde su samo dva člana tropske sume različita od  $\infty$ , i to podmatrica  $A_{i,q} = A_R$  za  $i = q$ , i podmatrica  $A_{i,q-1} = A_C$  za  $i = q - 1$  (6.27), pa je

$$A_{p,q}^{(k+1)} = A_{p,q}^{(k)}A_{q,q} \oplus A_{p,q-1}^{(k)}A_{q-1,q} = A_{p,q}^{(k)}A_R \oplus A_{p,q-1}^{(k)}A_C,$$

što potvrđuje teoremu.  $\square$

Za  $p = q$ , elementi podmatrice  $A_{p,q-1}^{(n-1)}$  iz (6.28) su jednaki  $\infty$  (neutralni element za  $\oplus$ ), pa je oblik podmatrica na glavnoj dijagonali

$$A_{p,q}^{(n)} = A_{p,q}^{(n-1)}A_R = A_R^{(n)}. \quad (6.29)$$

Kako su sve podmatrice  $A_{p,q}^{(n)}$  na dijagonali matrice  $A^{(n)}$  jednake, a podmatrica na osnovu (6.28) zavisi samo od podmatrice iz prethodne iteracije na istoj poziciji  $(p, q)$  i podmatrice iz prethodne iteracije na susednoj poziciji sa leve strane u istoj vrsti  $(p, q - 1)$ , izraz (6.28) možemo pisati kao

$$A_m^{(n)} = A_m^{(n-1)}A_R \oplus A_{m-1}^{(n-1)}A_C, \quad (6.30)$$

gde je  $m$  horizontalno rastojanje podmatrice  $A_{p,q}^{(n)}$  od dijagonale matrice  $A^{(n)}$ , odnosno  $m = q - p$ .

U cilju projektovanja PDT polja sa Euklidskom metrikom, matricu tranzitivnog zatvaranja  $A^*(\mathbf{G})$  menjamo matricom težina najkraćih puteva  $S(\mathbf{G})$ , gde vrednost elementa  $s_{i,j}$  označava zbir težina svih grana na najkraćem putu od čvora  $v_i$  do čvora  $v_j$ . Neka je  $S_m$  matrica najkraćih puteva u matrici  $S(\mathbf{G})$  koja poziciono odgovara matrici  $A_m^{(n)}$  iz izraza (6.30).

**Lema 6.7** Za podmatricu  $S_m$  matrice  $S(\mathbf{G})$  važi

$$S_m = S_{m-1}A_C, \quad (6.31)$$

gde je

$$S_0 = \bigoplus_{i=1}^{\infty} A_R^{(i)}. \quad (6.32)$$

**Dokaz** Elementi podmatrice  $S_m$  su po tvrđenju leme 6.4 jednaki minimumu odgovarajućih podmatrica, odnosno

$$S_m = \bigoplus_{i=1}^{\infty} A_m^{(i)}. \quad (6.33)$$

Zamenom (6.30) u (6.33) imamo

$$\begin{aligned} S_m &= \bigoplus_{i=1}^{\infty} \left( A_m^{(i-1)} A_R \oplus A_{m-1}^{(i-1)} A_C \right) = \\ &= \bigoplus_{i=1}^{\infty} A_m^{(i-1)} A_R \oplus \bigoplus_{i=1}^{\infty} A_{m-1}^{(i-1)} A_C = \\ &= \bigoplus_{i=0}^{\infty} A_m^{(i)} A_R \oplus \bigoplus_{i=0}^{\infty} A_{m-1}^{(i)} A_C = \\ &= \left( A_m^{(0)} A_R \oplus \bigoplus_{i=1}^{\infty} A_m^{(i)} A_R \right) \oplus \left( A_{m-1}^{(0)} A_C \oplus \bigoplus_{i=1}^{\infty} A_{m-1}^{(i)} A_C \right) = \\ &= S_m A_R \oplus S_{m-1} A_C. \end{aligned} \quad (6.34)$$

S obzirom da matrica najkraćih puteva ostaje matrica najkraćih puteva i posle min-plus množenja, važi da je

$$S_m = S_m A_R. \quad (6.35)$$

Imajući u vidu osobine sabiranja u min-plus algebri, tvrđenje leme dobija se zamenom (6.34) u (6.35).

Početnu podmatricu  $S_0$  iz izraza (6.32) dobijamo zamenom opšteg oblika podmatrice  $A_{m=0}^{(n)} = A_R^{(n)}$ , koja se nalazi na dijagonali matrice  $A^{(n)}$ , iz izraza (6.29) u izraz (6.33).  $\square$

**Teorema 6.8** *Opšti oblik podmatrice  $S_m$  matrice najkraćih puteva  $S(\mathbf{G}_E)$  grafa  $\mathbf{G}_E$ , opisanog matricom susjedstva (6.27), je*

$$S_m = \begin{bmatrix} m & m+1 & \cdots & m+N-1 \\ \vdots & & & \\ 2 & 3 & & 2+N-1 \\ 1 & 2 & & 1+N-1 \\ 0 & 1 & & N-1 \\ \vdots & & & \\ \infty & \infty & \cdots & m \end{bmatrix}_{N \times N}, \quad (6.36)$$

odnosno, opšti oblik elemenata  $s_{i,j}$  matrice  $S_m$  je

$$s_{i,j}^m = \begin{cases} j - i + m, & i \leq j + m \\ \infty, & \text{inače} \end{cases}. \quad (6.37)$$

**Dokaz** Na osnovu (6.24) i opšteg oblika podmatrice  $A_R$  moguće je pokazati da su elementi podmatrice  $A_R^{(n)} = [r_{i,j}^{(n)}]$  oblika

$$r_{i,j}^{(n)} = \begin{cases} j - i, & i \leq j \\ \infty, & \text{inače} \end{cases}. \quad (6.38)$$

Tvrđenje teoreme dokazaćemo matematičkom indukcijom. Kako (6.38) ne zavisi od  $n$ , zamenom (6.38) u (6.32) dobijamo

$$s_{i,j}^0 = \begin{cases} j - i, & i \leq j \\ \infty, & \text{inače} \end{cases}, \quad (6.39)$$

što je jednako izrazu (6.37) za  $m = 0$ .

Neka važi

$$s_{i,j}^k = \begin{cases} j - i + k, & i \leq j + k \\ \infty, & \text{inače} \end{cases}. \quad (6.40)$$

Na osnovu leme 6.7 imamo da je  $S_{k+1} = S_k A_C$ , odnosno, iz definicije množenja matrica u min-plus algebri (6.24)

$$s_{i,j}^{k+1} = \bigoplus_{l=1}^N s_{i,l}^k \odot c_{l,j}, \quad (6.41)$$

gde je  $c_{i,j}$  element matrice  $A_C$  na poziciji  $(i, j)$ . Kako je  $c_{l,j} = 0$  samo za  $l = j + 1$  u svim kolonama matrice  $A_C$  sem u poslednjoj, a za sve ostale

parove  $(l, j)$  element  $c_{l,j} = \infty$ , tropska suma iz izraza (6.41) se svodi samo na jedan sabirak, i to

$$s_{i,j}^{k+1} = s_{i,j+1}^k \odot 0 = s_{i,j+1}^k = \begin{cases} j+1-i+k, & i \leq j+1+k \\ \infty, & \text{inače} \end{cases}.$$

Za poslednju kolonu, gde je jedino  $c_{N,N} = 1$ , suma (6.41) postaje

$$s_{i,N}^{k+1} = \bigoplus_{l=1}^N s_{i,l}^k \odot c_{l,j} = s_{i,N}^k \odot 1 = \begin{cases} N-i+k, & i \leq N+k \\ \infty, & \text{inače} \end{cases},$$

čime je dokazano tvrđenje leme.  $\square$

**Primer** Za težinski graf propagacije greške tipa "B", heksagonalnog polja za množenje brojeva, prikazanog na slici 6.11, na osnovu 6.31, matrica najkraćih puteva  $S(\mathbf{G})$  je oblika

$$S(\mathbf{G}) = \begin{bmatrix} S_0 & S_1 & S_2 & S_3 \\ \infty & S_0 & S_1 & S_2 \\ \infty & \infty & S_0 & S_1 \\ \infty & \infty & \infty & S_0 \end{bmatrix},$$

odnosno, na osnovu (6.36)

$$S(\mathbf{G}) = \begin{bmatrix} 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 & 2 & 3 & 4 & 5 & 3 & 4 & 5 & 6 \\ \infty & 0 & 1 & 2 & 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 & 2 & 3 & 4 & 5 \\ \infty & \infty & 0 & 1 & \infty & 0 & 1 & 1 & 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 \\ \infty & \infty & \infty & 0 & \infty & \infty & 0 & 2 & \infty & 0 & 1 & 2 & 0 & 1 & 2 & 3 \\ \hline \infty & \infty & \infty & \infty & 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 & 2 & 3 & 4 & 5 \\ \infty & \infty & \infty & \infty & \infty & 0 & 1 & 2 & 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & \infty & 0 & 1 & 1 & 0 & 1 & 2 & 3 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty & \infty & 0 & 2 & \infty & 0 & 1 & 2 \\ \hline \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & 2 & 0 & 1 & 2 & 3 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & \infty & 0 & 1 & 1 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty & \infty & 0 & 2 \\ \hline \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & 2 & 3 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & 2 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix}.$$

Poslednji element prve vrste matrice,  $s_{0,15} = 6$ , odgovara zbiru težina grana na najkraćem putu između čvora  $v_0$  i  $v_{15}$ , što je moguće proveriti na slici 6.11.  $\triangle$



Neka je funkcija prostornog uređenja  $f_o : V \rightarrow \mathbf{N}^2$ ,  $(p, q) = f_o(v_i)$ , data sa

$$p = \left\lfloor \frac{i_v}{N} \right\rfloor, \quad q = N - (i_v \bmod N) - 1, \quad (6.42)$$

gde je  $i_v = 0, 1, \dots, T - 1$ , indeks vrste matrice tranzitivnog zatvaranja. Uređeni parovi  $(p, q)$  naznačeni su na slici 6.11 u gornjem desnom uglu odgovarajućeg čvora.

Označimo sa  $M_W = [m_{i,j}^W]$  težinsku mapu uticaja greške, čiji element  $m_{i,j}$  označava maksimalnu težinu greške na izlaznom rezultatu, izazvanu defektom na čvoru  $v_i$ , koji po funkciji prostornog uređenja (6.42) odgovara elementu  $m_{i,j}$ .

**Lema 6.8** *Elementi težinske mape uticaja greške  $M_W = [m_{i,j}^W]$  grafa  $\mathbf{G}_E$  sa topologijom  $4^4$ , prikazanom na slici 6.11, su oblika*

$$m_{i,j}^W = N + i - j, \quad (6.43)$$

za svaki uređeni par  $(i, j) \in \{0, 1, \dots, N - 1\}^2$ .

**Dokaz** Na osnovu funkcije uređenja (6.42), elementi poslednje kolone podmatrice  $S_{N-1}$  odgovaraju elementima prve vrste težinske mape uticaja, elementima druge vrste odgovaraju elementi poslednje kolone podmatrice  $S_{N-2}$ , itd. Poslednjoj vrsti težinske mape uticaja odgovara poslednja kolona podmatrice  $S_0$ . Odnosno, na osnovu funkcije uređenja (6.42) elementu  $m_{i,j}^W$  odgovara element  $s_{N-1-j,N}^{N-1}$  podmatrice  $S_{N-1-i}$ .

Maksimalna greška polja je  $2^L$ , pa je vrednost elementa  $m_{i,j}^W$  u težinskoj mapi uticaja, koji predstavlja eksponent greške, jednaka razlici eksponenta maksimalne greške i rastojanja posmatranog čvora od nje, odnosno

$$m_{i,j}^W = L - s_{N-1-j,N}^{N-1} = 2N - (N - i + j) = N + i - j$$

□

**Primer** Mapa težinskog uticaja greške  $M_W$  grafa  $\mathbf{G}_E$  za  $N = 4$ , prikazanog na slici 6.11 je

$$M_W = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 5 & 4 & 3 & 2 \\ 6 & 5 & 4 & 3 \\ 7 & 6 & 5 & 4 \end{bmatrix}. \quad (6.44)$$

Na osnovu težinske mape uticaja greške (6.44) može se zaključiti da će greška izazvana defektom na čvoru  $v_{15}$  (slika 6.11), kome odgovara element  $m_{3,0}$  matrice  $M_W$ , biti  $\Delta_E = 2^7$ . Greška izazvana defektom na čvoru  $v_{14}$ , kome odgovara element  $m_{3,1}$  biće  $\Delta_E = 2^6$ , itd. △

Na osnovu mape težinskog uticaja  $M_W = [m_{i,j}^W]$ , čiji je oblik dat lemom 6.8, formiraćemo mape uticaja greške na osnovu definicije 4.7, u skladu sa definicijom otkaza sistema po Euklidskoj metrici (def. 4.2).

**Teorema 6.9** *Opšti oblik matrice uticaja greške  $M_\alpha = [m_{i,j}^\alpha]$  heksagonalnog polja za množenje brojeva, za otkaz sistema definisan Euklidskom metrikom, je*

$$m_{i,j} = \begin{cases} 1, & \alpha \geq N - i + j \\ 0, & \text{inače} \end{cases} \quad (6.45)$$

gde je  $\alpha$  tolerancija sistema na greške.

**Dokaz** Po Euklidskoj metrici, do otkaza sistema došlo je ukoliko važi

$$d_E(A_t, A_{err}) \geq \Delta_E = 2^{L-\alpha} = 2^{2N-\alpha}.$$

Greška koju može izazvati čvor  $v_i$ , koji odgovara elementu težinske mape uticaja  $M_W$  je  $\Delta_E = 2^{m_{i,j}^W}$ . Po tvrđenju leme 6.8, opšti oblik elementa  $m_{i,j}^W$  je  $m_{i,j}^W = N + i - j$ , pa će do otkaza sistema doći usled defekata na čvorovima kojima odgovaraju elementi  $m_{i,j}$  za koje važi

$$N + i - j \geq 2N - \alpha.$$

Sređivanjem prethodnog izraza po  $\alpha$  dobija se tvrđenje teoreme.  $\square$

**Primer** Tvrđenje teoreme 6.9 ilustrovaćemo na primeru polja za  $N = 4$ , prikazanom na slici 2.22, čiji je težinski graf uticaja  $\mathbf{G}_E$  dat na slici 6.11.

Za  $\alpha = 0$  polje predstavlja ET polje, odnosno za svaki uređeni par  $i, j \in 0, 1, \dots, N - 1$  važi  $m_{i,j}^0 = 0$ . Mapa uticaja greške  $M_1$ , za  $\alpha = 1$ , na osnovu teoreme 6.9, je

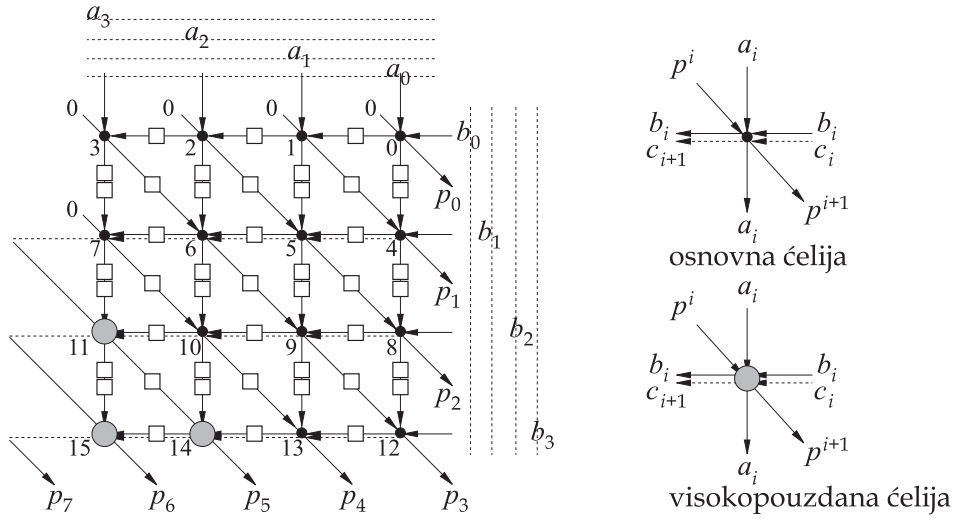
$$M_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Mapa uticaja greške za  $\alpha = 2$  je

$$M_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \quad (6.46)$$

itd.  $\triangle$

DFG parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva za  $N = 4$  i  $\alpha = 2$  prikazan je na slici 6.12.



Slika 6.12: DFG parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva za  $N = 4$  i  $\alpha = 2$

### 6.2.4 Analiza prinosa parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva

U prethodnom poglavlju prikazan je postupak projektovanja parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva. S jedne strane, PDT polje je projektovano uzeti za otkaz sistema definiciju 4.3 sa Hamingovom metrikom, a sa druge strane otkaz po Euklidskoj metrici (def. 4.2). U ovom poglavlju biće data analiza prinosa parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva za oba slučaja.

Odnos prinosa ET i PDT polja dat je teoremama 5.1 i 5.2. Prinos zavisi od tolerantnosti sistema na greške  $\alpha$ , posredno preko funkcije  $\Gamma(\alpha)$ , koja predstavlja meru netolerantne oblasti sistema.

**Teorema 6.10** *Veličina netolerantnog dela heksagonalnog PDT polja za množenje brojeva sa Hamingovom metrikom je*

$$\Gamma_H(\alpha) = \begin{cases} 0, & \alpha = 0 \\ 1, & \text{inače} \end{cases} \quad (6.47)$$

**Dokaz** Na osnovu tvrđenja teorema 6.6 i 6.7, za  $\alpha \geq 1$  sve ćelije polja je potrebno projektovati kao visokopouzdana, što potvrđuje teoremu.  $\square$

**Teorema 6.11** *Veličina netolerantnog dela heksagonalnog PDT polja za množenje brojeva sa Euklidskom metrikom je*

$$\Gamma_E(\alpha) = \begin{cases} \frac{\alpha(\alpha+1)}{2N^2}, & \alpha \leq N \\ -\frac{\alpha+\alpha^2-4\alpha N+2(-1+N)N}{2N^2}, & \alpha > N \end{cases}. \quad (6.48)$$

**Dokaz** Veličina netolerantnog dela heksagonalnog polja sa slike 6.12 jednaka je

$$\Gamma_E(\alpha) = \frac{J(M_\alpha)}{N^2}, \quad (6.49)$$

gde je  $J(M_\alpha)$  funkcija čija je vrednost jednaka broju jedinica u matrici  $M_\alpha$ , datoj sa (6.45). Na osnovu (6.45), elementi matrice  $M_\alpha$  za koje važi  $m_{i,j} = 1$  su elementi na i ispod kvazi-dijagonale matrice za koju važi

$$i = j + (N - \alpha).$$

Za polje  $N = 4$  i  $\alpha = 2$  matrica  $M_2$  data je izrazom (6.46). Generalno, za  $\alpha \leq N$  važi

$$J(M_\alpha) = 1 + 2 + \dots + \alpha = \sum_{i=1}^{\alpha} i = \frac{\alpha(\alpha + 1)}{2}. \quad (6.50)$$

Ukoliko je  $\alpha > N$ , tada je na sumu elemenata koji se nalaze na i ispod glavne dijagonale potrebno je dodati elemente koji se nalaze iznad glavne dijagonale

$$\begin{aligned} J(M_\alpha) &= \sum_{i=1}^N i + \sum_{i=N-1}^{2N-\alpha} i = \frac{\alpha(\alpha + 1)}{2} + \frac{(1 + \alpha - 3N)(N - \alpha)}{2} = \\ &= -\frac{\alpha + \alpha^2 - 4\alpha N + 2(-1 + N)N}{2}. \end{aligned} \quad (6.51)$$

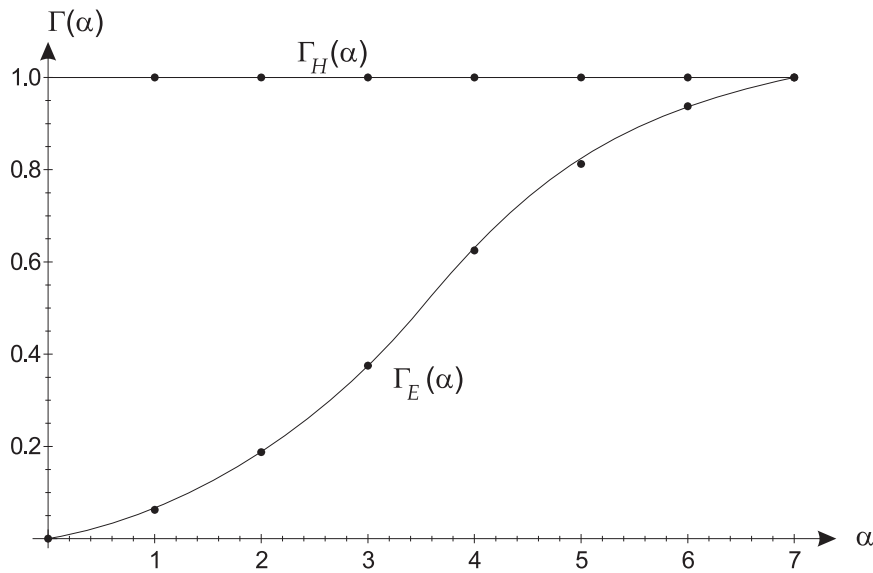
Iz (6.49), (6.50) i (6.51) sledi tvrđenje teoreme.  $\square$

**Primer** Skup uređenih parova  $(\alpha, J(M_\alpha))$  za polje  $N = 4$  sa slike 6.12 je

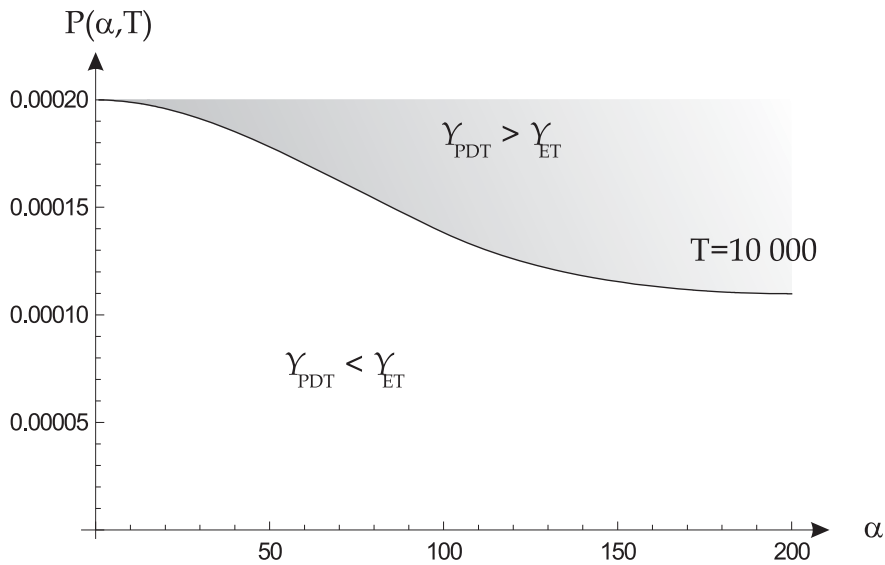
$$\{(0, 0), (1, 1), (2, 3), (3, 6), (4, 10), (5, 13), (6, 15), (7, 16)\},$$

a grafički prikaz funkcije  $\Gamma_E(\alpha)$  za ovo polje dat je na slici 6.13. Na istom grafiku prikazana je i funkcija  $\Gamma_H(\alpha)$ , data izrazom (6.47). Potrebno je napomenuti da su vrednosti date na grafiku diskretne, a da je linija koja ih povezuje prikazana radi lakšeg praćenja oblika funkcije.  $\triangle$

Zamenom (6.48) u (5.9) dobija se verovatnoća pojave defekta, u funkciji od  $\alpha$ , za koju ET i PDT tehnike imaju isti prinos. Na slici 6.14 prikazan je grafik funkcije (5.9) za  $\Gamma(\alpha)$  dato sa (6.48), za  $N = 100$ ,  $L = 2N = 200$ ,  $T = N^2 = 10.000$  i  $\alpha = 0, 1, \dots, L - 1$ .



Slika 6.13: Grafički prikaz funkcije  $\Gamma(\alpha)$  parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva, za  $N = 4$  i  $\alpha = 0, 1, \dots, L - 1$



Slika 6.14: Verovatnoća pojave defekta za koju ET i PDT tehnike imaju isti prinos u funkciji od  $\alpha$  kod PDT heksagonalnog polja za množenje brojeva sa  $T = 10.000$  ćelija

N	$\alpha/L = 0$		0.25		0.50		0.75		1.0				
	Broj ćelija	Impl. [kG]	$\Gamma(\alpha)$	Broj ćelija	Impl. [kG]	$\Gamma(\alpha)$	Broj ćelija	Impl. [kG]	$\Gamma(\alpha)$	Broj ćelija	Impl. [kG]		
4	16	0,35	0,19	22	0,40	0,62	36	0,53	0,93	46	0,59	48	0,60
8	64	1,36	0,16	84	1,93	0,56	136	2,17	0,90	180	2,33	192	2,40
16	256	5,19	0,14	328	5,75	0,53	528	7,46	0,89	712	9,06	768	9,55

Tabela 6.5: Rezultati implementacije parcijalno visokopouzdanog heksagonalnog polja za množenje matrice i vektora sa Euklidskom metrikom i TMR tehnikom za povećanje pouzdanosti ćelija

### 6.2.5 Rezultati implementacije parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva

U cilju ilustracije kompromisa između ET i PDT heksagonalnog polja za množenje brojeva, u ovom poglavlju dati su rezultati FPGA implementacije.

Arhitektura prikazana na slici 6.12 opisana je u VHDL-u i implementirana na Xilinx Spartan2 XC2S200 FPGA čipu. Kao i kod VHDL opisa prethodno projektovanih polja, parcijalna visoka pouzdanost ugrađena je u VHDL u obliku funkcije, koja u zavisnosti od dimenzija polja  $N$  i tolerancije sistema na greške  $\alpha$  određuje mape uticaja greške (6.45).

U tabeli 6.5 dati su rezultati FPGA implementacije parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva sa Euklidskom metrikom i TMR tehnikom za povećanje pouzdanosti ćelije, dok su u tabeli 6.6 dati rezultati polja sa SC3 tehnikom za povećanje pouzdanosti. U oba slučaja implementacija je izvršena za polja  $N = \{4, 8, 16\}$ .

Kod prethodno projektovanih polja, rezultati implementacije dati su u zavisnosti od promene vrednosti  $\Gamma(\alpha)$  (tabele 4.2, 6.2 i 6.4). Kod polja sa slike 6.12 nemoguće je naći vrednosti  $\alpha_1$  i  $\alpha_2$ , za koje važi

$$\Gamma_{N=4}(\alpha_1) = \Gamma_{N=8}(\alpha_2).$$

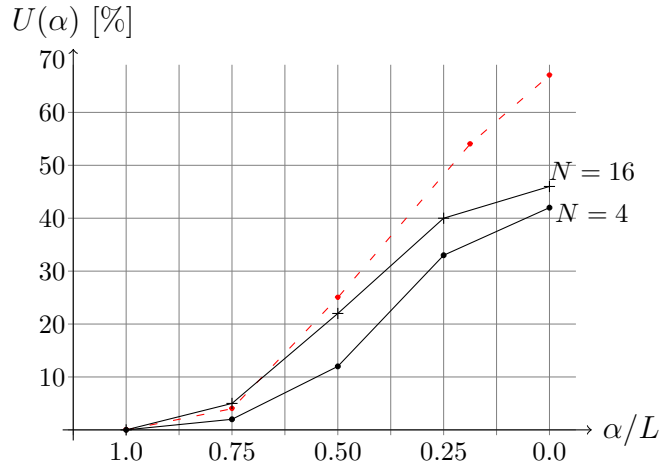
N	$\alpha/L = 0$		0.25		0.50		0.75		1.0				
	Broj ćelija	Impl. [kG]	$\Gamma(\alpha)$	Broj ćelija	Impl. [kG]	$\Gamma(\alpha)$	Broj ćelija	Impl. [kG]	$\Gamma(\alpha)$	Broj ćelija	Impl. [kG]		
4	16	0,35	0,19	22	0,40	0,62	36	0,53	0,93	46	0,59	48	0,60
8	64	1,36	0,16	84	1,92	0,56	136	2,16	0,90	180	2,32	192	2,40
16	256	5,19	0,14	328	5,74	0,53	528	7,45	0,89	712	9,04	768	9,54

Tabela 6.6: Rezultati implementacije parcijalno visokopouzdanog heksagonalnog polja za množenje matrice i vektora sa Euklidskom metrikom i SC3 tehnikom za povećanje pouzdanosti ćelija

Zbog toga, po vrstama tabela su dati rezultati implementacije polja za konstantno  $N$  i različite vrednosti koločnika  $\alpha/L$ , umesto  $\Gamma(\alpha)$ . Na primer, za  $\alpha = 0$ , vrednost  $\alpha/L$  je 0 za sve arhitekture, pa ove kolone sadrže rezultate implementacije osnovng ET polja. Vrednost  $\alpha/L = 0.25$ , kod polja  $L = 2N = 8$ , znači da je vrednost  $\alpha = 2$ . Kod polja  $N = 8$  i  $L = 16$  za  $\alpha/L = 0.25$  vrednost  $\alpha$  je 4, itd.

Odgovarajuće vrednosti  $\Gamma(\alpha)$  su takođe date. Za polje dimenzije  $N = 4$  i  $\alpha/L = 0.25$ , kod koga je  $\alpha = 2$  (tabela 6.5), broj potrebnih ćelija je 22. Ova vrednost dobijena je na osnovu (6.46), jer je samo 3 ćelije, od ukupno 16, potrebno projektovati kao visokopouzdana, pa je  $(16 - 3) \cdot 1 + 3 \cdot 3 = 22$ . Vrednost  $\Gamma(\alpha)$  je u ovom slučaju  $3/16 = 0,1875 \approx 0,19$ .

Na slici 6.15 grafički su ilustrovani rezultati implementacije polja iz tabele 6.4. Rezultati su formirani korišćenjem izraza (4.17) i dati kao ušteda u odnosu na FT polje za koje važi  $\alpha/L = 1$ .



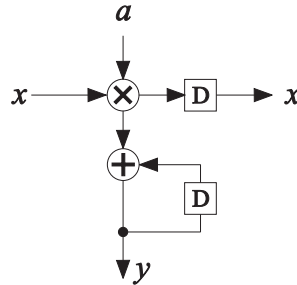
Slika 6.15: Grafik uštede resursa potrebnih za implementaciju parcijalno visokopouzdanog heksagonalnog polja za množenje brojeva

### 6.3 Projektovanje parcijalno visokopouzdanog polja sa PDT ćelijama

U poglavlju 6.1.1 dat je postupak projektovanja parcijalno visokopouzdanog jednosmernog 1D polja za množenje matrice i vektora. Za minimalnu jedinicu zamene u primeru projektovanja 1D PDT polja sa slike 6.2 izabrana je jedna ćelija polja. Ukoliko se ćelija dalje posmatra kao podsistem polja koji,

pored ostalog, sadrži množač, uštedu ilustrovanu na slici 6.5 je dalje moguće povećati na taj način što će se i ćelija polja projektovati kao parcijalno visokopouzdan sistem. U ovom poglavlju dat je postupak projektovanja i analiza parcijalno visokopouzdanog jednosmernog 1D polja za množenje matrice i vektora sa parcijalno visokopouzdanim ćelijama.

Kao polaznu tačku za projektovanje parcijalno visokopouzdanog jednosmernog 1D polja za množenje matrice i vektora sa parcijalno visokopouzdanim ćelijama uzećemo PDT jednosmerno 1D polje sa slike 6.2. Podsystem koji čini jednu osnovnu ćeliju polja sa slike 2.18 prikazan je na slici 6.16. Ćelija realizuje funkciju  $y_i = a \cdot x + y_{i-1}$ . Kod PDT polja sa slike 6.2 minimalna jedinica zamene je ćelija, pa je predložena visokopouzdana ćelija TMR, odnosno SC3 sistem.



Slika 6.16: Ćelija jednosmernog 1D polja za množenje matrice i vektora

Neka je množač u ćeliji sa slike 6.16 implementiran heksagonalnim poljem za množenje brojeva sa slike 2.22, i neka je minimalna jedinica zamene u ovakvom sistemu jedna ćelija množača. Razmotrimo primer jednosmernog 1D polja za množenje matrice i vektora, koji može tolerisati greške na izlaznim rečima

$$y_i, \quad i = 0, 1, \dots, L - \alpha_1 - 1,$$

dok na izlazima

$$y_i, \quad i = L - 1, L - 2, \dots, L - \alpha_1,$$

može tolerisati samo greške za koje važi

$$d_E(y_t, y_{err}) < \Delta_E = 2^{L-\alpha_2}.$$

Ovo polje ima  $\alpha_1$  visokopouzdanih ćelija, na čijim izlazima greška usled defekata ne prelazi  $2^{\alpha_2}$ . Veličina netolerantnog dela polja je data sa (6.8) i obuhvata

$$\Gamma_1(\alpha_1) = \frac{\alpha_1}{T}$$



ćelija. Ukoliko netolerantni deo svake ćelije označimo sa  $\Gamma_2(\alpha_2)$ , tada je netolerantni deo celog sistema

$$\Gamma(\alpha_1, \alpha_2) = \Gamma_1(\alpha_1) \cdot (\beta \cdot \Gamma_2(\alpha_2)), \quad (6.52)$$

gde je  $\beta \in (0, 1]$  korektivni faktor. Ovaj faktor izražava činjenicu da ćelija, pored PDT podsistema, sadrži sabirač i memorijske elemente (slika 6.16). Vrednost  $\beta$  je veličina množača u odnosu na sve komponente ćelije.

**Teorema 6.12** *Za parcijalno visokopouzdan sistem, kod koga su podsistemi u netolerantnom delu projektovani kao parcijalno visokopouzdan podsistemi, važi*

$$\Gamma(\alpha_1, \alpha_2) \leq \Gamma_1(\alpha_1),$$

gde je  $\Gamma_1(\alpha_1)$  procenat sistema obuhvaćen netolerantnim delom. Jednakost važi za  $\alpha_2 = 1$  i  $\beta = 1$ .

**Dokaz** Kako je  $\Gamma_2(\alpha_2) \leq 1$ , zavisno od topologije i izabranog  $\alpha_2$ , tvrđenje teoreme proizilazi iz (6.52).  $\square$

Po tvrđenju teoreme 6.12, uvođenjem PDT podsistema u netolerantnom delu sistema moguće je postići dodatnu uštedu resursa. Ova ušteda dolazi na račun daljeg smanjenja visokopouzdanosti oblasti arhitekture.

Parcijalno visokopouzdan jednostrano 1D polje za množenje matrice i vektora, sa  $T = 4$  i  $\alpha_1 = 2$ , sa parcijalno visokopouzdanim ćelijama sa  $N = 4$  i  $\alpha_2 = 2$  prikazano je na slici 6.17.

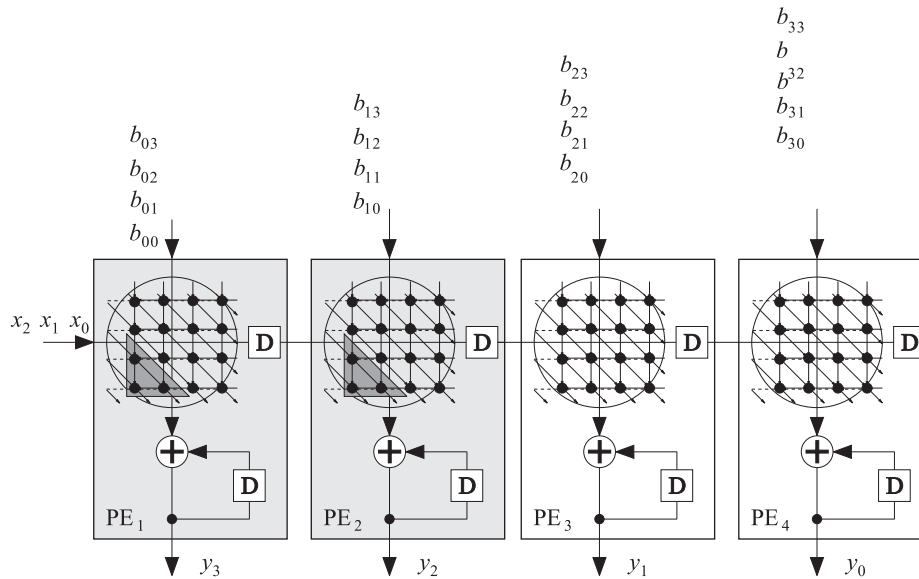
U cilju ilustracije kompromisa između pouzdanosti sistema za zadate tolerancije na greške ( $\alpha_1, \alpha_2$ ) i zauzeća resursa, sistem sa slike 6.17 opisan je u VHDL-u i implementiran na Xilinx Spartan2 XC2S200 FPGA čipu. Visokopouzdanosti ćelije implementirane su kao SC3 sistemi. Rezultati implementacije dati su u tabeli 6.7.

U tabeli 6.7 prikazani su rezultati implementacije 1D polja sa  $T_{1D} = 8$  ćelija polja i širinom ulaznih podataka  $n_x = 8$  sa  $\alpha_1 = 0, 2, \dots, 8$  i  $\alpha_2 = 0, 4, \dots, L - 1$ . Za izabranu širinu ulaznih podataka implementirani množač ima  $T_{2D} = N^2 = n_x^2$  ćelija. Potrebno je napomenuti da se potrebni resursi za implementaciju osnovnog ET polja ( $\alpha_1 = \alpha_2 = 0$ ), vrednost 29,19kG u tabeli 6.7, razlikuju od rezultata implementacije polja istih dimenzija datih u tabeli 6.2, jer u implementacijama nisu korišćeni isti tipovi množača.

U tabeli 6.7 je za svaki par ( $\alpha_1, \alpha_2$ ) dat ukupan broj ćelija množača u celom sistemu, i zauzeće resursa na čipu, izraženo u ekvivalentnom broju gejtova. Ukupan broj ćelija množača u sistemu određen je kao

$$T = ((1 - \Gamma_1(\alpha_1)) \cdot T_{1D} \cdot T_{2D}) + \Gamma_1(\alpha_1) \cdot T_{1D} \cdot (T_{2D} + \Gamma_2(\alpha_2) \cdot T_{2D} \cdot 2),$$

gde je  $(1 - \Gamma_1(\alpha_1)) \cdot T_{1D}$  ukupan broj ćelija tolerantnog dela polja, kod koga svaka ćelija polja sadrži  $T_{2D}$  polja množača, a  $(T_{2D} + \Gamma_2(\alpha_2) \cdot T_{2D} \cdot 2)$  broj ćelija netolerantnog množača kod  $\Gamma_1(\alpha_1) \cdot T_{1D}$  ćelija polja. Na primer, za  $\alpha_1 = 0$  i  $\alpha_2 = 0$  nema visokopouzdanih ćelija, jer su  $\Gamma_1(\alpha_1) = \Gamma_2(\alpha_2) = 0$ , pa je ukupan broj ćelija množača  $8 \cdot 8^2 = 512$ . Za  $\alpha_1 = 2$  i  $\alpha_2 = 4$ , dve ćelije polja imaju visokopouzdanu množače, takve da svaki od njih ima po deset visokopouzdanih ćelija. Kako visokopouzdanu množači u ovom slučaju imaju



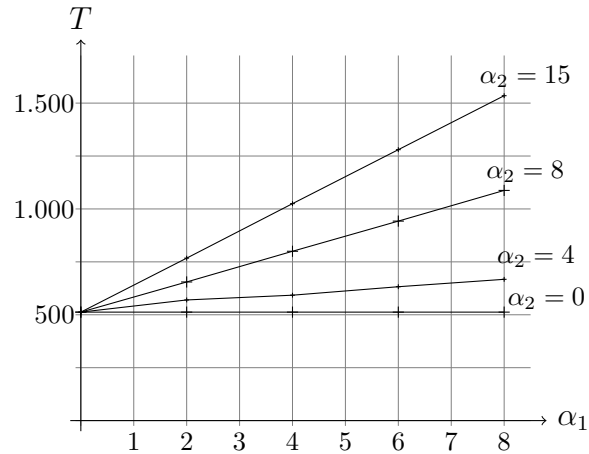
Slika 6.17: Parcijalno visokopouzdanu jednosmerno 1D polje za množenje matrice i vektora, sa  $T = 4$  i  $\alpha_1 = 2$ , sa parcijalno visokopouzdanim ćelijama sa  $N = 4$  i  $\alpha_2 = 2$

$\alpha_2$	$\alpha_1 = 0$		2		4		6		8	
	$T$	Impl. [kG]	$T$	Impl. [kG]	$T$	Impl. [kG]	$T$	Impl. [kG]	$T$	Impl. [kG]
0	512	29,19	512	29,19	512	29,19	512	29,19	512	29,19
4	512	29,19	552	30,39	592	31,57	632	32,30	672	32,32
8	512	29,19	656	30,81	800	32,23	944	32,70	1088	32,55
12	512	29,19	744	31,16	976	32,48	1208	33,08	1440	33,31
15	512	29,19	768	31,30	1024	32,63	1280	33,32	1536	33,66

Tabela 6.7: Rezultati FPGA implementacije parcijalno visokopouzdanog jednosmernog 1D polja za množenje matrice i vektora sa parcijalno visokopouzdanim ćelijama za  $T_{1D} = 8$  i  $n_x = 8$

$(64 - 10) \cdot 1 + 10 \cdot 3 = 84$  ćelije, to je ukupan broj ćelija  $6 \cdot 64 + 2 \cdot 84 = 552$  (tabela 6.7).

Grafička ilustracija teoretskih rezultata implementacije iz tabele 6.7 data je na slici 6.3. Slika 6.3 ilustruje i tvrđenje teoreme 6.12. Naime, izborom vrednost  $\alpha_2$  u intervalu  $0 < \alpha_2 < L - 1$ , dodatno je moguće podešavati kompromis između zauzeća i stepena pouzdanosti, između dva granična slučaja  $\alpha_2 = 0$  i  $\alpha_2 = L - 1$ .



Slika 6.18: Grafički prikaz rezultata implementacije PDT jednosmernog 1D polja za množenje matrice i vektora sa PDT ćelijama, za  $T_{1D} = 8$  i  $T_{2D}$

Promena rezultata FPGA implementacije datih u tabeli 6.7 nije izražena kao promena teoretskih rezultata ilustrovanih na slici 6.3. Promena potrebnih resursa za FPGA implementaciju, sa promenom parametra  $\alpha_2$  umanjena je za faktor  $\beta$ , uveden u izrazu (6.52).

Parcijalno visokopouzdan sistem, kod koga su podsistemi projektovani kao parcijalno visokopouzdan podsistemi, uvodi novi stepen slobode između potrebnih resursa i pouzdanosti sistema, i omogućava dodatnu uštedu resursa kod sistema kod kojih je tolerancija na greške veća.

# Glava 7

## Zaključak

U cilju smanjenja uticaja defekta u procesu proizvodnje, a samim tim i povećanja prinosa, u ovom radu predložena je primena metoda za povećanje pouzdanosti dela arhitekture, označenog kao deo od najvećeg značaja za datu primenu. Za razliku od ET koncepta, kod koga je projektant svestan mogućnosti postojanja defekta, zna njegovu poziciju i uticaj, i ne menja svojstva arhitekture kako bi otklonio posledice defekta, kod predložene tehnike parcijalnog povećanja pouzdanosti projektant na sistematski način određuje uticaje defekta na delove arhitekture i određuje koje delove treba projektovati kao visokopouzdanost. Ovakvim kompromisom dveju tehnika uveden je novi stepen slobode kod projektovanja sistema, koji rezultuje arhitekturom parcijalno tolerantnom na defekte.

Kako bi se definisale greške nastale usled defekata i formalno postavile osnove tehnike za analizu propagacije grešaka, u radu su predloženi metrički sistemi. Otkaz sistema je precizno definisan u odnosu na Hamingovu i Euklidsku metriku, a parcijalna visokopouzdanost definisana je kao osobina arhitekture za izabranu metriku. Parcijalna visokopouzdanost definisana je formalnim matematičkim jezikom, a uz definiciju parcijalne visoke pouzdanosti predložene su i sistematske tehnike za markiranje dela arhitekture od najvećeg značaja za datu primenu.

Tehnika za projektovanje parcijalno visokopouzdanih polja ilustrovana je na primeru semi-sistoličke arhitekture BP FIR filtra. Matematički aparat i postupak za analizu arhitektura je izložen i dat u formi algoritma. Postupak za projektovanje ilustrovan je na primeru relativno složene topologije BP filtra.

Na primeru BP polja pokazano je da PDT dizajn uvek donosi uštedu u poređenu sa arhitekturom koja je visokopouzdana u celosti. Međutim, ključno pitanje na koje je u ovom radu dat odgovor je da li je, i ako jeste kada je PDT dizajn isplativiji u odnosu na ET dizajn? Formalnim matematičkim

jezikom definisan je prinos ET i PDT sistema. Izvedena je i detaljno prikazana analiza isplativosti PDT tehnike projektovanja u odnosu na ET tehniku. Analiza je izvedena za polja u opštem obliku, bez obzira na topologiju i izabranu metriku otkaza, i rezultovala je obrascem za izračunavanje parametara proizvodnog procesa za koje PDT daje veći prinos u odnosu na ET tehniku. Obrazac za izračunavanje parametara proizvodnje primenljiv je na sisteme proizvoljne topologije uz proizvoljno izabranu metriku otkaza sistema. Korišćenjem predloženog matematičkog aparata za analizu PDT sistema pokazano je da za verovatnoće pojave defekata tipične za nanotehnologiju predloženi PDT pristup projektovanju arhitektura donosi veću ekonomsku dobit u odnosu na ET pristup.

Matematički aparat za analizu prinosa predložene tehnike ilustrovan je na primeru ortogonalnog semi-sistoličkog polja BP FIR filtra. Za topologiju BP FIR filtra određena je verovatnoća pojave defekta počev od koje PDT tehnika daje veći prinos u odnosu na ET tehniku. Verovatnoća je data u formi funkcije kod koje su parametri dimenzije polja.

Kako bi se odgovorilo na pitanje kako topologija polja utiče na isplativost primene PDT metoda, u ovom radu su detaljno analizirane različite topologije sistema i pravci prostiranja grešaka kroz njih. U cilju povećanja prinosa razmotrena je veličina i pozicija dela arhitekture gde je potrebno primeniti metode visoke pouzdanosti, za zadata verovatnoću greške procesa proizvodnje i nivo greške koji se u primeni algoritma može tolerisati.

Metod je proveren na više primera parcijalno visokopouzdanih polja, a za karakteristične topologije pokazani su trendovi kretanja granica defekata za koje je prinos kod primene predložene tehnike pozitivan. Isplativost primene predložene tehnike projektovanja visokopouzdanih sistema proverena je na primerima jednodimenzionalnih i dvodimenzionalnih sistoličkih i semi-sistoličkih polja.

Za tipove grešaka koje se kod 1D polja propagiraju samo u jednom smeru pokazano je da predloženi metod daje dobre rezultate u zavisnosti od otpornosti aplikacije na greške. Takođe je pokazano da kod 1D polja, za tipove grešaka koje se propagiraju u dva smera, nije moguće postići uštedu u odnosu na polja u celosti projektovana kao visokopouzdana. Analiza 2D topologija je izvršena nad dva tipa ortogonalnih i jednom heksagonalnom topologijom. Za analizu ortogonalnih topologija uzete su topologije BP FIR filtra sa Hamingovom metrikom otkaza i topologija za specifičan tip grešaka u heksagonalnom polju za množenje brojeva. Pokazano je da se za slučaj topologije BP FIR filtra sa Hamingovom metrikom može postići ušteda u odnosu na polje u celosti projektovano kao visokopouzdan, dok u slučaju ortogonalne topologije množača za Hamingovu metriku nije moguće postići uštedu. Takođe je dokazano da nije moguće postići uštedu kod heksagonalne topologije, ukoliko

je metrika otkaza Hamingova metrika.

Kod heksagonalne topologije polja za množenje brojeva sa Euklidskom metrikom otkaza moguće je postići značajnu uštedu resursa pri primeni tehnike projektovanja parcijalno visokopouzdanog polja. U cilju analize propagacije grešaka za Euklidsku metriku razmatrani su težinski grafovi uticaja greške. Da bi se pojednostavile analitičke operacije nad grafovima u matricnoj reprezentaciji, analiza za Euklidsku metriku izvedena je u tropskoj algebri. Dat je set lema i teorema sa dokazima u tropskoj algebri koji opisuju oblike težinskih matrica najkraćih puteva, težinskih mapa uticaja i mapa uticaja grafova. Korišćenjem rezultata dobijenih tropskom algebrom nad grafovima propagacije dat je postupak projektovanja heksagonalnog parcijalno visokopouzdanog polja za množenje brojeva.

Koncept projektovanja parcijalno visokopouzdanih sistema primenjen je nad podsistemima sistema, čime je definisana nova klasa parcijalno visokopouzdanih sistema sa parcijalno visokopouzdanim podsistemima. Pokazano je da je na račun daljeg smanjenja pouzdanosti moguće postići dodatnu uštedu primenom predložene tehnike nad podsistemima PDT sistema.

Teorijski rezultati dobijeni projektovanjem parcijalno visokopouzdanih polja različitih topologija eksperimentalno su provereni. Sva predložena polja su opisana u VHDL-u, a koncept parcijalne visoke pouzdanosti je ugrađen u odgovarajuće VHDL funkcije. Ove funkcije za zadata topologiju polja i zadate parametre polja određuju deo koji je potrebno projektovati kao visokopouzdan. Rezultati implementacije polja opisanih u VHDL-u na Xilinx FPGA čipu su prikazani tabelarno i grafički, a uz svaki eksperimentalno dobijeni rezultat data je i teorijski određena vrednost.

Rad je podeljen u sedam glava.

Nakon uvodne glave, u drugoj glavi dat je pregled sistema za digitalnu obradu signala. Prikazani su karakteristični DSP algoritmi i opisan je uobičajeni način reprezentacije DSP algoritama. Prikazana je operacija konvolucije i korelacije, digitalno filtriranje i diskretna kosinusna transformacija. Opisan je način reprezentacije DSP algoritama blok dijagramima, grafom toka signala, grafom toka podataka i grafom zavisnosti. Date su karakteristične klase i topologije sistoličkih polja i semi-sistoličkih polja. Detalno je prikazana implementacija DSP algoritama na sistoličkim i semi-sistoličkim poljima, koja su korišćene u narednim glavama kao osnova za projektovanje PDT polja. Prikazano je 1D jednosmerno i dvosmerno polje za množenje matrice i vektora, ortogonalno semi-sistoličko *bit-plane* polje za digitalno FIR filtriranje i heksagonalno sistoličko polje za množenje brojeva. U cilju ilustracije rezultata implementacije polja na FPGA čipu ukratko je predstavljena arhitektura Xilinx Spartan2 FPGA čipa, na kome je izvršena implementacija svih polja datih u ovom radu. Rezultati FPGA implementacije osnovnih

polja dati su na kraju druge glave.

U trećoj glavi prikazane su osnovne tehnike projektovanja visokopouzdanih sistema. Na početku glave je data i precizno definisana uobičajena terminologija koja je u upotrebi kod potencijalno defektnih sistema. Od tehnika za povećanje pouzdanosti sistema detaljno je opisana hardverska redundansa, koja je korišćena za projektovanje PDT sistema, a informaciona, vremenska i softverska redundansa su pomenute radi celovitosti izlaganja. Za hardverske metode je data analiza pouzdanosti sistema. Definisana je minimalna jedinica zamene, i u odnosu na minimalnu jedinicu zamene izvedena je verovatnoća otkaza celog sistema u zavisnosti od verovatnoće pojave defekta na jednom podsistemu. Poslednji deo glave posvećen je konceptu tolerancije sistema na greške i nepreciznim izračunavanjima, kao uvodnom delu u projektovanje parcijalno visokopouzdanih sistema. Neprecizna izračunavanja su ilustrovan na primeru telefonske sekretarice sa defektnom memorijom.

Četvrta, peta i šesta glava sastoje se od originalnih rezultata ovog rada. Četvrto poglavlje posvećeno je formalno-matematičkoj definiciji parcijalne visoke pouzdanosti. Uvedena je metrika otkaza i formalno definisana kao Hamingova i Euklidska metrika otkaza. Definicija PDT sistema data je kroz set definicija i lema, ilustrovanih primerima. U okviru lema su opisane i dokazane osobine neophodne za projektovanje PDT sistema. Tranzitivno zatvaranje, kao operacija neophodna za određivanje uticaja čvorova grafa, detaljno je prikazana i ilustrovan na primeru tranzitivnog zatvaranja relativno složene topologije BP FIR filtra. Postupak određivanja tranzitivnog zatvaranja BP arhitekture detaljno je prikazan. Tehnika projektovanja parcijalno visokopouzdanih arhitektura uobličena je u formu algoritma i prikazana dijagramom toka. Na kraju trećeg poglavlja detaljno je prikazan postupak projektovanja parcijalno visokopouzdanog BP FIR filtra.

Peta glava daje odgovor na pitanje o isplativosti predložene tehnike projektovanja. Na primeru je pokazano da primena tehnika za povećanje pouzdanosti, u cilju smanjenja uticaja defekta nastalih u procesu proizvodnje, može dovesti do ekonomskih gubitaka. Prvi deo glave posvećen je definiciji i određivanju prinosa različitih tehnika, dok je u centralnom delu glave dat postupak određivanja parametara procesa fabrikacije čipova za koje je prinos fabrikacije sistema u predoženoj tehnici bolji u odnosu na postojeće tehnike projektovanja. Određen je i dat je obrazac u generalnom obliku koji daje ovu zavisnost za bilo koju topologiju i obim sistema. Pokazano je da je za obim sistema, koji je tipičan za nano-tehnologiju, predloženi metod može dati određeni doprinos zavisno od topologije sistema. Na primeru topologije BP FIR filtra ilustrovani su rezultati vezani za prinos parcijalno visokopouzdanog arhitekture.

Uticaj topologije na prinos parcijalno visokopouzdanih polja razmatran

je u šestoj glavi. Prvi deo glave posvećen je jednodimenzionalnim poljima. Za jednosmerno i dvosmerno 1D polje za množenje matrice i vektora data je analiza uticaja i propagacije različitih tipova grešaka kroz polje. Projektovanje parcijalno visokopouzdanih 1D polja izvedeno je za izabrane klase defekta koji se mogu javiti u polju. Procedura projektovanja primenjena je nad jednosmernom i dvosmernom 1D polju. Projektovana PDT polja predstavljena su DFG-om. Pokazano je da u pogledu primene PDT tehnike bolje karakteristike imaju tipovi defekata koji propagiraju greške samo u jednom smeru. Izvedena je analiza prinosa, i dati su rezultati FPGA implementacije polja. U drugom delu ove glave predstavljeno je parcijalno visokopouzdanu heksagonalno polje za množenje brojeva. Polje je projektovano u odnosu na Hamingovu i Euklidsku metriku otkaza. Data je analiza propagacije grešaka nastalih od različitih tipova defekata. PDT polje projektovano u odnosu na Hamingovu metriku je analizirano sa aspekta dva tipa defekta. Za oba tipa defekta je pokazano da predloženom metodom nije moguće postići dodatnu uštedu na ovoj topologiji. Da bi se pojednostavile analitičke operacije nad grafovima u matricnoj reprezentaciji, analiza za Euklidsku metriku izvedena je u tropskoj algebri. Dat je pregled osnovnih definicija tropske algebre i izveden set lema koje povezuju tropsku algebru sa algoritmima za obilazak grafova. U tropskoj algebri su izvedeni opšti oblici mapa težinskih uticaja greške date topologije. Heksagonalno PDT polje je predstavljeno DFG-om. Analiza uticaja tolerantnosti aplikacije na visokopouzdanu deo PDT arhitekture je detaljno prikazana. Projektovano polje je opisano u VHDL-u i implementirano na FPGA čipu. Rezultati implementacije su dati tabelarno i grafički.

Primenom tehnika za povećanje pouzdanosti nad delom sistema dat je još jedan stepen slobode pri projektovanju računarskih arhitektura i omogućen je kompromis između tolerancije aplikacije na greške i potrebnih resursa za implementaciju. Rezultati analize različitih topologija pokazali su da povećanje broja tokova kroz 1D polja ima negativan efekat na rezultate primene tehnike. Pokazano je da kod 2D polja, zavisno od orijentacije polja i izbornog skupa izlaznih čvorova, efekti primene predložene tehnike značajno variraju. Euklidska metrika daje veću uštedu u slučajevima kada je moguće primeniti. Rezultati ovog rada pokazuju da, bez obzira na topologiju, predložena tehnika projektovanja parcijalno visokopouzdanih polja može povećati prinos i smanjiti uticaj defekta nastalih u procesu fabrikacije naročito kod sistema sa velikim brojem podsistema, tipičnim za nano-tehnologiju.





# Literatura

- [1] Keshab K. Parhi. *VLSI Digital Signal Processing Systems: Design and Implementation*. John Wiley & Sons, 1999.
- [2] Albert Y. Zomaya. *Parallel and Distributed Computing Handbook*. McGraw-Hill Professional, 1996.
- [3] K. Hwang and F. Briggs. *Computer Architectures and Parallel Processing*. McGraw Hill, New York, 1984.
- [4] S. Zhang, M. Choi, and N. Park. Defect characterization and yield analysis of array-based nanoarchitecture. In *Proceedings on the 4th IEEE Conference on Nanotechnology*, pages 50–52, Munich, Germany, August 2004.
- [5] International Technology Roadmap for Semiconductors. Recommendations. <http://public.itrs.net/>, 2001.
- [6] Cristian Constantinescu. Trends and challenges in VLSI circuit reliability. pages 14–19, July-August 2003.
- [7] M. Breuer, S. Gupta, and T. Mark. Defect and error tolerance in the presence of massive numbers of defects. *IEEE Transactions on Design & Test of Computers, Vol. 21*, pages 216–227, 2004.
- [8] I. Koren and C. Krishna. *Fault-Tolerant Systems*. Morgan Kaufmann, San Francisco, US, March 2007.
- [9] B. W. Johnson. *Fault Tolerance: The Electrical Engineering Handbook*. CRC Press, U.S., 1993.
- [10] J. von Neumann. *Automata Studies*, chapter Probabilistic Logic and Synthesis of Reliable Organisms from Unreliable Components, pages 43–98. Princeton University Press, Princeton, US, C. Shannon and J. McCarthy edition, 1956.

- [11] A. DeHon. *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computing*, chapter Defect and Fault Tolerance. Morgan Kaufmann, San Francisco, US, 2008.
- [12] Barry W. Johnson. *Design and Analysis of Fault-Tolerant Digital Systems*. Addison-Wesley Publishing Company, 1989.
- [13] M. Breuer. Multimedia applications and imprecise computation. In *Proceedings on the 8th Euromicro conference on Digital System Design, Euromicro*, pages 0–7695–2433–8/05, Porto, Portugal, September 2005.
- [14] Vladimir Ćirić, Jelena Kolokotronis, and Ivan Milentijević. Partial error-tolerance for bit-plane FIR filter architecture. *International Journal of Electronics and Communication, Elsevier Science (AEU)*, Vol. 63:398–405, May 2009.
- [15] M. A. Breuer and H.H. Zhu. Error-tolerance and multi-media. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IHH-MSP '06.*, pages 521 – 524, Pasadena, CA, USA, 2006.
- [16] I. Richardson. *H.264 and MPEG-4 Video Compression Video Coding for Next Generation Multimedia*. John Wiley & Sons, In., New York, USA, 2003.
- [17] Vladimir Ćirić and Ivan Milentijević. Area-time tradeoffs in H.264/AVC deblocking filter design for mobile devices. In *Proceedings of IEEE Conference on Signal Processing and its Applications - ISSPA 2007*, pages 1–4, Sharjah, United Arab Emirates, February 2007.
- [18] Vladimir Ćirić and Ivan Milentijević. Design of area-time efficient H.264/AVC deblocking filter for mobile devices. In *Proceedings of 51st ETRAN Conference*, pages 1–4, Herceg Novi - Igalo, Montenegro, Jun 2007.
- [19] T.-Yu Hsieh, K.-Jong Lee, and M. Breuer. Reduction of detected acceptable faults for yield improvement via error-tolerance. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1599 – 1604, Nice, France, 2007.
- [20] Vladimir Ćirić, Aleksandar Cvetković, and Ivan Milentijević. Yield analysis of partial defect tolerant bit-plane array. *Elsevier, Int. J., Computers and Mathematics with Applications*, Vol. 59(Number 1):98–107, January 2010.

- [21] Vladimir Ćirić and Ivan Milentijević. Configurable folded array for FIR filtering. *Journal of Systems Architecture, An International Journal, Elsevier Science*, Vol. 54(Issue 1-2):177–196, January/February 2008.
- [22] T. Noll. Semi-systolic maximum rate transversal filters with programmable coefficients. In *Workshop of Systolic Architectures*, pages 103–112, Oxford, 1986.
- [23] B. Parhami and D. Kwai. Parallel architectures and adaptation algorithms for programmable FIR digital filters with fully pipelined data and control flows. 2003.
- [24] M. Renfors and Y. Neuvo. The maximum sampling rate of digital filters under hardware speed constraints. *IEEE Trans. on Circuits and Systems*, CAS-28(No. 3):196–202, March 1981.
- [25] E. Hewitt and K. Ross. Abstract harmonic analysis. analysis on locally compact abelian groups. *Die Grundlehren der mathematischen Wissenschaften, Band 152 Springer-Verlag, New York-Berlin 1970*, Volume 2, Structure and analysis for compact groups:771–780, 1970.
- [26] K.K.Parhi and D.G. Messerschmitt. Static rate-optimal scheduling of iterative data-flow programs via optimum unfolding. *IEEE Trans. on Computers*, Vol. 40(No. 2):178–195, February 1991.
- [27] P. R. Wallace. *Mathematical analysis of physical problems*. Courier Publications, Dover, UK, 1984.
- [28] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice-Hall, New Jersey, 2002.
- [29] A. Oppenheim and R. Schaffer. *Discrete-Time Signal Processing*. Prentice Hall, 1989.
- [30] P. Hilfinger. A high-level language and silicon compiler for digital signal processing. In *Proc. of IEEE Custom Integrated Circuits Conference*, pages 213–216, 1985.
- [31] H. De Man, J. Rabaey, P. Six, and L. Claesen. Cathedral-II: a silicon compiler for digital signal processing. *IEEE design and testing*, Vol. 3(No. 6):13–25, Dec. 1986.
- [32] Volnei A. Pedroni. *Circuit Design with VHDL*. Massachusetts Institute of Technology, Cambridge, Massachusetts, London, England, 2004.

- [33] R. Crochiere and A. Oppenheim. Analysis of linear digital networks. In *Proc. of the IEEE*, volume Vol. 63, pages 581–595, 1975.
- [34] K.K. Parhi. Algorithm transformation techniques for concurrent processors. In *Proceedings of the IEEE*, volume Volume 77, pages 1879–1895, Dec 1989.
- [35] Ivan Milentijević, Igor Nikolić, Oliver Vojinović, Vladimir Ćirić, and Teufik Tokić. Folded bit-plane architectures. In *Proceedings of the 2nd International Conference on Informatics and Information Technology - CIIT*, pages 282–292, Molika, Macedonia, December 2001.
- [36] Ivan Milentijević, Igor Nikolić, Vladimir Ćirić, Oliver Vojinović, and Teufik Tokić. Synthesis of folded fully pipelined bit-plane architecture. In *Proc. 22th International Conference on Microelectronics (MIEL 2002)*, Vol 2, pages 683–686, Nis, Serbia, May 2002.
- [37] Ivan Milentijević, Teufik Tokić, Igor Nikolić, Oliver Vojinović, and Vladimir Ćirić. Synthesis of folded FIR filter architecture with re-ordered partial products. In *Proceedings of a Workshop on Computational Intelligence and Informational Technologies*, pages 155–160, Nis, June 2001.
- [38] M. Stojcev, T. Tokic, and I. Milentijevic. The limits of semiconductor technology and oncoming challenges in computer microarchitectures and architectures. *Facta Universitatis, Ser. Electronics and Energetics*, Vol. 17:285–312, 2004.
- [39] H. T. Kung. Notes on vlsi computation. *Parallel Processing Systems*, Cambridge University Press, New York, pages 339–356, 1983.
- [40] H. T. Kung. Why systolic architectures? *IEEE Computer*, Vol. 15:37–46, 1982.
- [41] H. T. Kung and C. E. Leiserson. Systolic arrays (for VLSI). Tech. report CS-79-103, Carnegie Mellon University, Pittsburgh, Pa., 1978.
- [42] Y-C. Lin and F-C. Lin. Classes of systolic arrays for digital filtering. *Int. J. Electronics*, Vol. 70, No. 4:729–737, 1991.
- [43] Allen D. Malony. Regular processor arrays. In *Proceedings on International Conference on Parallel Processing*, page Rpt. No. 734, St. Charles, Illinois, August 1988.

- [44] Kung S. Y. *VLSI Array Processors*. Prentice Hall, Englewood Cliffs, N.Y., 1988.
- [45] J. Smith. *Introduction to Digital Filters with Audio Applications*. Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, California, USA, May 2004.
- [46] D. Gibbon. *Convolution: filtering*. Acoustic Phonetics and Speech Technology, University of Bielefeld, Germany, May, 1996.
- [47] P. Denyer and D. Renshaw. *VLSI Signal Processing: A Bit-Serial Approach*. Addison-Wesley, 1985.
- [48] J. Hein. *Discrete Mathematics*. Jones & Bartlett Publishers, 2nd edition, August 2002.
- [49] Branko Grünbaum and Geoffrey Shephard. Tilings by regular polygons. *Mathematics Magazine*, Vol. 50(No. 5):227–247, Nov., 1977.
- [50] Emina Milovanović and Vladimir Ćirić. *Praktikum za laboratorijske vežbe iz Paralelnih računarskih sistema*. Elektronski fakultet u Nišu, Niš, Srbija, 2004.
- [51] Ivan Milentijević, Vladimir Ćirić, Oliver Vojinović, and Teufik Tokić. Folded semi-systolic FIR filter architecture with changeable folding factor. *Neural, Parallel & Scientific Computations, Dynamic Publishers, Atlanta*, Vol. 10, No 2:235–247, 2002.
- [52] Ivan Milentijević, Vladimir Ćirić, Teufik Tokić, and Oliver Vojinović. Folded bit-plane FIR filter architecture with changeable folding factor. In *Proceedings of EUROMICRO Conference on Digital System Design - DSD 2002*, pages 45–52, Dortmund, Germany, September 2002.
- [53] Vladimir Ćirić and Ivan Milentijević. Coefficient bit reordering method for configurable FIR filtering on folded bit-plane array. In *Proceedings on 8th EUROMICRO Conference on Digital System Design - DSD*, pages 135–138, Porto, Portugal, September 2005.
- [54] Ivan Milentijević and Vladimir Ćirić. Synthesis of coefficient bit reordering module for folded bit-plane arrays. In *Proceedings of 4rd Int. Conf. CiiT*, pages 332–342, Molika, Macedonia, December 2003.
- [55] I.Milentijevic, I.Milovanovic, E.Milovanovic, M.Tosic, and M.Stojcev. Two-level pipelined systolic arrays for matrix-vector multiplication.

- Journal of Systems Architecture, The Euromicro Journal*, Vol. 44(No. 5):383–388, 1998.
- [56] I.Milentijevic, I.Milovanovic, E.Milovanovic, and M.Stojcev. The design of optimal planar systolic arrays for matrix multiplication. *Computers Mat. Applic*, Vol. 33(Mo. 6):17–35, 1997.
- [57] I.Milovanovic, E.Milovanovic, I.Milentijevic, and M.Stojcev. Designing of processor-time optimal systolic arrays for band matrix-vector multiplication. *Computers Math. Applic*, Vol. 32(No. 2):21–31, 1996.
- [58] E.Milovanovic, I.Milentijevic, and I.Milovanovic. Designing of processor-time optimal hexagonal systolic array for matrix multiplication. *Computers and Artificial Intelligence*, Vol. 16(No. 1):1–11, 1997.
- [59] M.Tosic, I.Milentijevic, and E.Milovanovic. Unidirectional two-way pipelined linear systolic array for matrix-vector multiplication. *Facta Universitatis, Nis, Ser. Math. Inform*, Vol. 9(No. 11):129–142, 1996.
- [60] Ivan Mielntijevic. *Projektovanje visoko pouzdanog semi-sistoličkog polja sa cifarsko-serijskim prenosom podataka za realizaciju konvolucije*. Magistarski rad, Elektronski fakultet u Nišu, 1993.
- [61] Vladimir M. Ciric. *Projektovanje savijenih semi-sistoličkih polja za FIR filtriranje*. Magistarski rad, Elektronski fakultet u Nišu, 2005.
- [62] Bagad V. S. *VLSI design*. Technical Publications Pune, Pune, India, 2009.
- [63] Xilinx inc. Spartan-II FPGAs: Spartan-II architecture. Xilinx Products and Services, Silicon Solutions, USA, [http://www.xilinx.com/xlnx/xil\\_prodcatt\\_product.jsp?iso...7862](http://www.xilinx.com/xlnx/xil_prodcatt_product.jsp?iso...7862), 2001.
- [64] D. Phatak and I. Koren. Complete and partial fault tolerance of feed-forward neural nets. *IEEE Transactions on Neural Networks*, Vol. 6, Issue 2:446–456, Mar 1995.
- [65] I.Milentijevic, D.Maksimovic, M.Tosic, and M.Stojcev. Fault - tolerant semi-systolic convolver of type f based on digit - serial arithmetic. In *Preprints of Synopsis Papers of 1st Balcan IFAC-type Conf. on Applied Automatic Systems*, pages 27–29, Ohrid, Macedonia, September 1993.

- [66] M. Breuer. Intelligible test techniques to support error-tolerance. In *Proceedings on the 13th Asian Test Symposium (ATS 2004)*, pages 0–7695–2235–1/04, IEEE Computer Society, 2004.
- [67] Michael Haselman and Scott Hauck. The future of integrated circuits: A survey of nanoelectronics. *Proceedings of the IEEE*, Vol. 98, Issue 1:11–38, January 2010.
- [68] Jude Rivers and Prabhakar Kudva. Reliability challenges and system performance at the architecture level. *Design & Test of Computers, IEEE*, Volume 26, Issue 6:62 – 73, Nov.-Dec. 2009.
- [69] Gordon Moore. Cramming more components into integrated circuits. *Electronics*, vol. 38, no. 8, 1965.
- [70] Vančo Litovski and Slobodan Lazović. *Osnovi elektronike*. Čuperak plavi, Elektronski fakultet u Nišu, 1996.
- [71] S. Iijima. Helical microtubules of graphitic carbon. *Nature*, vol. 354, no. 6341:56–58, 1991.
- [72] C.L. Brown et al. Introduction of catenanes into langmuir film and langmuir-blodgett multilayers. a possible strategy for molecular information storage materials. vol. 16, no. 4:1924–1930, 2000.
- [73] Tatjana Nikolić. *Projektovanje kombinacionih i sekvencijalnih mreža sa samoproverom na osnovu VHDL opisa*. Magistarski rad, Elektronski fakultet u Nišu, 2005.
- [74] Daniel P. Siewiorek and Robert S. Swartz. *Reliable Computer Systems: Design and Evaluation*. Bedford, MA: Digital Press, 1992.
- [75] Luis Entrena, Celia López, and Emilio Olías. Automatic generation of fault tolerant VHDL designs in RTL. In *Forum on Design Languages, FDL*, Lyon, France, Septembre 2001.
- [76] P.K. Lala. *Self-checking and fault-tolerant digital system design*. Morgan Kaufman Publishers, San Francisco, 2001.
- [77] K. De, C. Natarajan, D. Nair, and P. Banerjee. RSYN: A system for automated synthesis of reliable multilevel circuits. *IEEE Trans. on VLSI Systems*, Vol. 2, No. 2:184–195, June 1994.



- [78] Chaohuang Zeng, Nirmal R. Saxena, and Edward J. McCluskey. Finite state machine synthesis with concurrent error detection. In *Proceedings IEEE International Test Conference*, pages 672–679, USA, 27-30 September 1999.
- [79] N. Stojanovic, E. Milovanovic, I. Stojmenovic, I. Milovanovic, and T. Tokic. Mapping matrix multiplication algorithm onto fault-tolerant systolic array. *Elsevier, Computers & Mathematics with Applications*, Volume 48, Issues 1-2:275–289, July 2004.
- [80] Dhiraj K. Pradhan. *Fault-Tolerant Computer System Design*. Prentice Hall, U.S., 1996.
- [81] J. Bernoulli. *Ars Conjectandi*. Impensis thurnisiorum, Basel, Switzerland, 1713.
- [82] M. H. Azadmanesh and R. M. Kieckhafer. Exploiting omissive faults in synchronous approximate agreement. *IEEE Trans. on Computers*, Vol. 49(No. 10):1031–1042, October 2000.
- [83] O. Port. Chips that thrive on uncertainty. *Business Week Online*, March 26th, 2005.
- [84] D. Ernst et al. Razor: Circuit-level correction of timing errors for low-power operation. *IEEE Micro*, Vol. 24(No.6):10–20, November/December 2004.
- [85] M. Sadiku and M. Mazzara. Computing with neural networks. *IEEE Potentials*, pages 14–16, October 1993.
- [86] H. Chung and A. Ortega. System-level fault tolerance for motion estimation. *Tech. report USC-SIPI, #354*, Electrical Eng. Dept., Univ. of Southern California, 2002.
- [87] Vesna Smiljković, Vladimir Ćirić, Milan Pavlović, Nebojša Miletić, and Ivan Milentijević. Model virtuelnog komunikacionog kanala između test aplikacije i simulatora hardvera H.264 video kodeka. In *Proceedings of 52st ETRAN Conference*, Palić, Srbija, Jun 2008.
- [88] H. H. Kuok. Audio recording apparatus using an imperfect memory circuit. United States Patent 5,414,758, May 9, 1995.
- [89] Recommendation ITU-T G.723.1. Speech coders: Dual rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kbit/s. Int'l Telecommunication Union, 1996.

- [90] Stephan Weiss, Robert Stewart, and Gillian Davis. *Noise Reduction in Speech Applications*, chapter Noise and Digital Signal Processing, pages 3–46. CRC Press, U.S., 2002.
- [91] G. Milovanović and R. Đorđević. *Matematika za studente tehničkih fakulteta: I deo*. IP "Nauka", Beograd, Srbija, 1992.
- [92] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. Elsevier Science Publishing Co., Inc., New York, 1976.
- [93] Jay Yellen and Rollins College. *Basic Digraph Models and Properties, in Handbook of Graph Theory (Jonathan L. Gross and Jay Yellen, Eds.)*. CRC Press, Florida, 2003.
- [94] S. Warshall. A theorem of boolean matrices. *Journal of the ACM* 9, pages 11–12, 1962.
- [95] Alfred V. Aho. *Computer Representations of Graphs, in Handbook of Graph Theory (Jonathan L. Gross and Jay Yellen, Eds.)*. CRC Press, Florida, 2003.
- [96] S. Skiena. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Addison-Wesley, Reading, MA, 1990.
- [97] Jelena Kolokotronis, Vladimir Ćirić, and Ivan Milentijević. Error significance map for bit-plane FIR filtering array. In *Proc. 26th International Conference on Microelectronics (MIEL 2008)*, volume Vol. 2, pages 429–432, Nis, Serbia, May 2008.
- [98] F. Hausdorff. *Set Theory*. published in 1937, translated by J. R. Aumann, AMS, Bookstore, 2005.
- [99] Vladimir Simić, Vladimir Ćirić, and Ivan Milentijević. Partial defect tolerant 1d array for matrix-vector multiplication. In *Proceedings of 54th ETRAN Conference (RT8.3)*, pages 1–4, Donji Milanovac, 7-10. Jun 2010.
- [100] Jean-Eric Pin. Tropical semirings. *Idempotency, Publ. Newton Inst., Cambridge Univ. Press, Cambridge*, Vol. 11:50–69, 1998.
- [101] Imre Simon. Recognizable sets with multiplicities in the tropical semiring. *Mathematical foundations of computer science, Lecture Notes in Comput. Sci.*, Vol. 324:50–69, 1998.

- [102] Diane Maclagan and Bernd Sturmfels. *Introduction to Tropical Geometry*. University of Warwick, U.K., Nov. 2009.

# Dodatak A

## VHDL opis funkcije za određivanje mape uticaja greške *bit-plane* arhitekture

```
1  -- Package File Template
2  --
3  -- Purpose: This package defines supplemental types, subtypes,
4  -- constants, and functions
5
6
7  library IEEE;
8  use IEEE.STD_LOGIC_1164.all;
9
10 package My_package is
11
12     constant n : integer := 5;
13     constant m : integer := 2;
14     constant k : integer := 2;
15     constant lo : integer := 4;
16
17     type matrix_A is array ((m*k+1)*(m+lo) -1 downto 0, (m*k+1)*
18         (m+lo) -1 downto 0) of std_logic;
19     type matrix_M is array (m*k-1 downto 0, m+lo-1 downto 0) of
20         std_logic;
21
22     function ESM (m, k, lo, n: integer) return matrix_M;
23
24 end My_package;
25
26 package body My_package is
27
28     function ESM (m, k, lo, n: integer) return matrix_M is
29         variable A_var : matrix_A := (others => (others => '0'));
30         variable M_var : matrix_M := (others => (others => '0'));
31         variable A_dim : integer := m*k+1;
32         variable G_dim : integer := m+lo;
```

```

33  --- formiranje matrice A ---
34  for i in 0 to A_dim-1 loop
35      for j in 0 to A_dim-1 loop
36          --- formiranje blokova matrice A ---
37              for p in 0 to G_dim-1 loop
38                  for q in 0 to G_dim-1 loop
39                      --- formiranje pojedinih blokova Gc=Ac
40                      if j > i then
41                          if (q+(j-i) >= p) and (p >= q) then
42                              A_var(i*G_dim + p, j*G_dim + q) :=
43                                  '1';
44                          end if;
45                      end if;
46                      --- formiranje pojedinih blokova Gc=Ac
47                  end loop;
48              end loop;
49          --- formiranje blokova matrice A ---
50      end loop;
51  --- formiranje matrice A ---
52
53  --- formiranje matrice M ---
54  for k in 0 to A_dim-2 loop
55      for l in 0 to G_dim-1 loop
56          M_var(k, l) := A_var(k*G_dim + l, G_dim*A_dim-1-n);
57      end loop;
58  end loop;
59  --- formiranje matrice M ---
60
61  return M_var;
62  ---return A_var;
63 end ESM;
64
65 end My_package;

```



**Универзитет у Нишу**  
Универзитетска библиотека

Овај текст је део Дигиталног репозиторијума, јавно је доступан, и може се слободно користити за личне потребе, у образовне и научне сврхе. Ако користите текст, наведите извор.

Комерцијална употреба текста није дозвољена.

**University of Niš**  
University Library

This text is a part of the Digital repository of public domain. Permission is granted for personal, educational and scientific use. If you do use the document, indicate the source.

No permission is granted for commercial use.

