

UNIVERZITET U BEOGRADU  
ELEKTROTEHNIČKI FAKULTET

Vladimir M. Rajović

HARDVERSKA REALIZACIJA  
JEDNOPROLAZNOG BRZOG KODEKA  
SA VISOKIM STEPENOM KOMPRESIJE I  
MINIMALNIM ZAHTEVANIM RESURSIMA

doktorska disertacija

Beograd, 2014

UNIVERSITY OF BELGRADE  
SCHOOL OF ELECTRICAL ENGINEERING

Vladimir M. Rajović

HARDWARE REALIZATION OF ONE  
PASS FAST CODEC WITH A HIGH  
COMPRESSION RATIO AND MINIMAL  
REQUIRED RESOURCES

Doctoral Dissertation

Belgrade, 2014

# Podaci o mentoru i članovima komisije

## **Mentor:**

Prof. dr Milan Prokin, Univerzitet u Beogradu, Elektrotehnički fakultet

## **Članovi komisije:**

Prof. dr Milan Prokin, Univerzitet u Beogradu, Elektrotehnički fakultet

Prof. dr Miodrag Popović, Univerzitet u Beogradu, Elektrotehnički fakultet

Prof. dr Miroslav Lutovac, Univerzitet Singidunum, Tehnički fakultet

Prof. dr Irini Reljin, Univerzitet u Beogradu, Elektrotehnički fakultet

Doc dr Jelena Popović-Božović, Univerzitet u Beogradu, Elektrotehnički fakultet

## **Datum odbrane:**

---

# **Hardverska realizacija jednoprolaznog brzog kodeka sa visokim stepenom kompresije i minimalnim zahtevanim resursima**

## **Rezime doktorske disertacije**

Kompresija i kodovanje digitalne slike bave se minimizacijom memorijskih resursa potrebnih za predstavljanje, pohranjivanje i prenos digitalne slike. Korišćenje tehnika za kompresiju dovodi do značajnih ušteda. Brzi algoritmi i arhitekture za kompresiju i dekompresiju su od primarnog značaja za primene u kojima su potrebne performanse u stvarnom vremenu.

Postojeći postupci za kompresiju digitalnih slika sa visokim stepenom kompresije su dugotrajni i zahtevaju složene procesore i veliku memoriju velike propusne moći, što takođe povećava potrošnju. I vreme procesiranja i kapacitet memorije postojećih postupaka za kompresiju slike sa gubicima rastu sa stepenom kompresije.

U ovoj doktorskoj disertaciji pristupilo se rešavanju problema hardverske realizacije kodeka koji bi minimizirao potrebne memorijske resurse, bio brz i imao visok stepen kompresije. Korišćenjem kombinovanih naučnih metoda u tome se i uspelo. Minimizacija potrebnih memorijskih resursa postignuta je korišćenjem optimizovanih nestacionarnih filtara i novog algoritma dvodimenzionalne podopsežne transformacije, kao i jednoprolaznom realizacijom procenjivača verovatnoća. Brzina rada postignuta je korišćenjem jednoprolaznog algoritma za procenu verovatnoća i izbegavanjem deljenja ulazne slike na blokove. Visok stepen kompresije postignut je sjedinjenim delovanjem sastavnih delova rešenja: podopsežnog transformatora, kvantizera, jednoprolaznog procenjivača verovatnoća i entropijskog enkodera.

Hardverska realizacija iz ove disertacije verifikovana je u primeni sistema za kompresiju slike sa mrežnim interfejsom, gde je pokazano da je moguće u jedno programabilno logičko kolo nižeg cenovnog razreda smestiti ceo enkoder slike u boji, bez korišćenja spoljašnje memorije.

**Ključne reči:** kompresija slike, hardverska realizacija

**Naučna oblast:** tehničke nauke, elektrotehnika

**Uža naučna oblast:** elektronika

**UDK broj:** 621.3

# **Hardware Realization Of One Pass Fast Codec With A High Compression Ratio And Minimal Required Resources**

## **Abstract of the Doctoral Dissertation**

Digital image compression and coding deal with minimization of the memory resources needed for presentation, storage and transmission of digital image. Use of compression techniques leads to substantial savings. Fast algorithms and architectures for compression and decompression are of a primary importance in the applications where real time performance is required.

Existing techniques for digital image compression with high compression ratio are slow and require complex processors and large amounts of high bandwidth memory, at the same time increasing power consumption. Both processing time and memory capacity increase as the compression ratio increases.

In this doctoral dissertation, we approached the problem of hardware realization of a codec which minimizes the required memory resources, is fast and has a high compression ratio. The task has been accomplished by means of combined scientific methods. The minimization of the required memory resources is achieved by means of optimized nonstationary filters and a new algorithm of two-dimensional subband transformation, as well as a one pass realization of probability estimators. The speed of operation is achieved by means of one pass algorithm for probabilities estimation and the fact that the input image is not processed partially. The high compression ratio is achieved by joint effects of all the subsystems in the realization: subband transformer, quantizer, one pass probabilities estimator and entropy encoder.

The hardware realization from this dissertation has been verified by means of image compression system with network interface. It is shown that it is possible to implement

entire color image encoder within a low cost programmable logic device, with no external memory components.

**Key words:** image compression, hardware realization

**Scientific area:** technical sciences, electrical engineering

**Specific scientific area:** electronics

**UDK number:** 621.3

# Sadržaj

<b>1. <u>UVOD</u></b>	<b>1</b>
<b>2. <u>DIREKTNI DVODIMENZIONALNI PODOPSEŽNI TRANSFORMATOR</u></b>	<b>9</b>
Direktna jednodimenzionalna podopsežna transformacija	10
Direktna dvodimenzionalna podopsežna transformacija	21
Hardverska realizacija direktnog dvodimenzionalnog podopsežnog transformatora	48
<b>3. <u>KVANTIZATOR</u></b>	<b>59</b>
Hardverska realizacija kvantizatora sa baferovanjem	74
<b>4. <u>ENKODERSKI PROCENJIVAČ VEROVATNOĆA I ENTROPIJSKI ENKODER</u></b>	<b>82</b>
Enkoderski procenjivač verovatnoća	83
Entropijski enkoder	96
Hardverska realizacija enkoderskog procenjivača verovatnoća i entropijskog enkodera	104
<b>5. <u>SIMULACIJA HARDVERSKOG ENKODERA JEDNE RAVNI BOJE</u></b>	<b>114</b>



<b><u>6. IMPLEMENTACIJA SISTEMA ZA KOMPRESIJU</u></b>	
<b><u>SLIKE U BOJI</u></b>	<b><u>123</u></b>
<b><u>7. ZAKLJUČAK</u></b>	<b><u>132</u></b>
<b><u>LITERATURA</u></b>	<b><u>134</u></b>
<b><u>BIOGRAFIJA AUTORA</u></b>	<b><u>145</u></b>

## Uvod

**K**ompresija i kodovanje digitalne slike bave se minimizacijom memorijskih resursa potrebnih za predstavljanje i pohranjivanje digitalne slike. Nekomprimovana digitalna slika zauzima veliku količinu memorije. Na primer, slika u boji dimenzija 1920 x 1080 osmootnih piksela zahteva gotovo 6 MB memorije za pohranjivanje. HDTV video zapis učestanosti 30 slika/s zahteva gotovo 180 MB memorije po sekundi video zapisa. Količina potrebne memorije predstavlja problem prilikom masovnog pohranjivanja digitalne slike za potrebe arhiviranja u raznovrsnim aplikacijama, kao što su kancelarijsko poslovanje, medicinska dijagnostika, pohranjivanje video zapisa i daljinsko očitavanje. Brzi algoritmi za kompresiju i naročito dekompresiju su od primarne važnosti kod aplikacija koje rade sa arhivama slika.

Prenos digitalne slike je još jedna oblast gde su potrebne tehnike za kodovanje i kompresiju slike. Za komprimovanu sliku je potrebno kraće vreme prenosa. Korišćenje tehnika za kompresiju dovodi do značajnih ušteda kod video konferencije, prenosa medicinskih slika, daljinskog video nadzora i slično. Brzi algoritmi i arhitekture za kompresiju i dekompresiju su od primarnog značaja za primene u kojima su potrebne performanse u stvarnom vremenu, na primer kod video konferencije i televizije visoke rezolucije (HDTV). Kako je šum prisutan u većini komunikacionih kanala, robustnost tehnika za kodovanje i kompresiju na prisustvo šuma je takođe važna.

Tehnike za digitalnu kompresiju slike mogu biti podeljene u dve klase, *kompresija bez gubitaka* i *kompresija sa gubicima*. Kompresija bez gubitaka se koristi u aplikacijama gde je teško ili nemoguće naknadno pribaviti originalnu sliku, ili kada slike sadrže važne informacije koje mogu biti uništene tokom kompresije, na primer u medicinskoj dijagnostici ili kod satelitskih snimaka. Kompresija sa gubicima se može koristiti kada se originalna slika može ponovo pribaviti kao i kada se gubitak dela informacija na prijemnoj strani može tolerisati. Tipičan primer su video konferencija i digitalna televizija gde je krajnji prijemnik ljudsko oko. Poznato je da oko toleriše, odnosno ne primećuje, izvesne nesavršenosti slike, kao što je blagi gubitak informacije o boji. Ipak, postoje nesavršenosti koje su neprihvatljive, kao što su zamrljanost ivica i treperenje u video zapisu. Očigledno se karakteristike ljudske vizuelne percepcije (Human Visual System) moraju uzeti u obzir prilikom projektovanja koda slike kako bi se smanjilo vidljivo pogoršanje slike.

Sve tehnike za kompresiju digitalne slike zasnivaju se na iskorišćavanju informacione redundantnosti koja postoji u većini digitalnih slika. Ova redundantnost je posledica jake prostorne korelacije piksela unutar jedne slike. Redundantnost slike se može opisati na različite načine. Svaki od njih dovodi do posebne klase algoritama za kompresiju digitalne slike. *Statistička redundantnost* je direktno povezana sa distribucijom verovatnoće podataka slike i može se tretirati entropijskim tehnikama teorije informacija. Uklanjanje statističke redundantnosti dovodi do tehnika za kompresiju slike bez gubitaka (Hafmanovo kodovanje, kodovanje nizova simbola). Drugi način za opis redundantnosti slike je korišćenjem *predviđanja* u lokalnoj oblasti slike. Lokalni modeli predviđanja koriste jaku prostornu korelaciju i teže da dekorelišu piksele u lokalnoj oblasti. Prenos ili pohranjivanje modela predviđanja i dekorelisanih podataka rezultuju kompresijom slike. Većina predviđajućih metoda kompresije slike dovodi do kompresije sa gubicima. Konačno, kompresija slike može se postići pakovanjem informacija korišćenjem *transformacija slike*, na osnovu činjenice da pojedine ortogonalne transformacije (diskretna kosinusna transformacija, Karhunen-Loeve transformacija, diskretna sinusna transformacija, Walsh-Hadamard transformacija, transformacija sa talasićima (vejvlet)) mogu da koncentrišu energiju slike u nekoliko transformacionih koeficijenata. Stoga, kodovanje transformacionih koeficijenata može

dovesti do značajne kompresije podataka. Transformacioni algoritmi se uglavnom koriste kod kompresije sa gubicima. Kompresija pomoću transformacije slike postiže veliko smanjenje količine podataka tako da je trenutno jedna od najboljih poznatih tehnika za kodovanje [1].

U slučaju video zapisa, pored prostorne redundantnosti postoji i *vremenska redundantnost*, kao posledica činjenice da se susedne slike u zapisu najčešće malo razlikuju. Vremenska redundantnost otklanja se otkrivanjem i kompenzacijom pokreta, čime se postiže povećanje kompresije [2].

Postojeći postupci za kompresiju podataka sa visokim stepenom kompresije su dugotrajni i zahtevaju složene procesore i veliku memoriju velike propusne moći, što takođe povećava potrošnju [3]-[7].

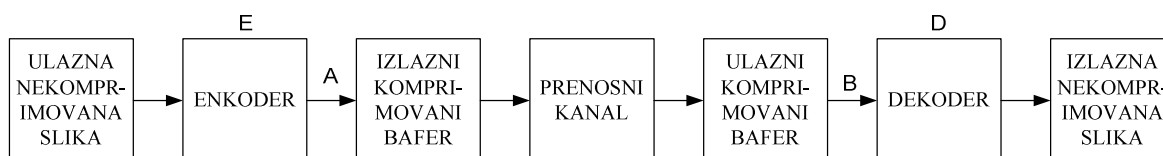
Dobar model za prirodne slike je baziran na spektru snage proporcionalnom  $f^{-2}$ , gde je  $f$  učestanost [8]. To znači da je veći deo energije koncentrisan na nižim učestanostima. Odatle bi pogodna podela učestanosti trebalo da bude finija u području nižih učestanosti i krupnija u području viših učestanosti.

Za većinu tipova slika, direktno kodovanje korišćenjem entropijskog koda ne ostvaruje zadovoljavajući stepen kompresije, tako da je potrebno prethodno razlaganje [3]-[5]. Postupci razlaganja za kompresiju nepokretne slike su: predviđajući, blokovski i podopsežne transformacije. Predviđajući postupci su pogodni za primene sa kompresijom bez gubitaka i sa malim stepenom kompresije. Glavni nedostaci blokovskih transformacionih postupaka, kao što je diskretna kosinusna transformacija (DCT), su blokovski artefakti pri većim stepenima kompresije, što je specijalno primetno u područjima slike sa niskom lokalnom varijansom. Nažalost, ljudski vizuelni sistem je veoma osetljiv na taj tip izobličenja slike.

Na **Slici 1.1** je prikazan blok dijagram komunikacionog sistema prema postojećem stanju tehnike koji komprimuje ulaznu nekomprimovanu sliku unutar enkodera  $E$ . Izlazni komprimovani bafer se popunjava izlaznom komprimovanom slikom  $A$ , koja se

prenosi kroz prenosni kanal. Ulazni komprimovani bafer se popunjava izlaznom komprimovanom slikom  $B$ , koja se dekomprimuje unutar dekodera  $D$ , da bi se konačno proizvela izlazna nekomprimovana slika. Ulazna nekomprimovana slika je preferencijalno u boji ili u nijansama sive u YUV 4:4:4 ili 4:2:2 formatu. Međutim, bilo koji format ulazne slike, kao što su RGB ili YCrCb, je takođe primenljiv uz upotrebu odgovarajućeg konvertora prostora boje za konverziju formata ulazne slike u YUV formate.

Podopsežni transformacioni postupci su primenljivi i na kompresiju bez gubitaka i sa gubicima, pri čemu je jedini vidljiv artefakt pri visokim stepenima kompresije Gibsov fenomen za linearne filtre, nazvan efekat talasanja (“ringing effect”), objašnjen u [9]. Transformacioni koeficijenti podopsega se izračunavaju rekursivnim filtriranjem prvo ulazne slike a zatim narednih rezultatnih slika pomoću filtra propusnika niskih učestanosti i filtra propusnika visokih učestanosti i smanjenjem učestanosti odabiranja rezultata. Svaki podopseg se odvojeno kodira za brzinu prenosa koja odgovara vizuelnom značaju podopsega. Ovo dovodi do vizualno prijatne rekonstrukcije slike i ne proizvodi blokovske artefakte. Podopsežno enkodovanje se sastoji od sledeća četiri koraka: (1) podopsežnog razlaganja; (2) kvantizacije; (3) procene verovatnoća; i (4) entropijskog kodovanja podopsega. Proces dekodiranja obuhvata inverzni redosled inverznih koraka. Koncept podopsežne transformacije je prvo uveden za kodovanje govora u [10], [11].

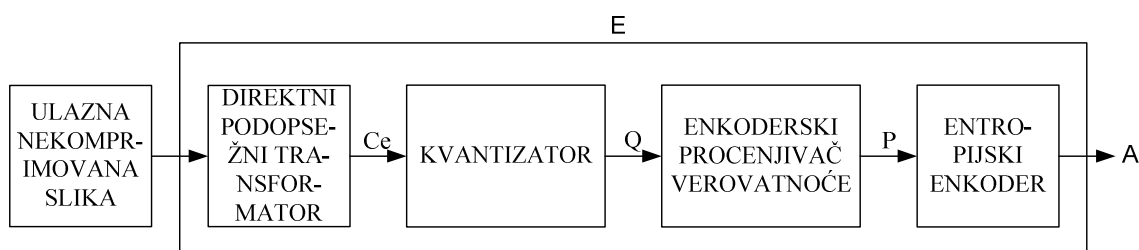


**Slika 1.1** Blok dijagram komunikacionog sistema prema postojećem stanju tehnike

Na **Slici 1.2** je predstavljen blok dijagram enkodera  $E$  sa slike 1.1, prema postojećem stanju tehnike. Ulazna nekomprimovana slika se prihvata od strane direktnog podopsežnog transformatora. Izlaz direktnog podopsežnog transformatora su transformacioni koeficijenti  $C_e$ , koji mogu biti kvantizovani u kvantizovane transformacione koeficijente  $Q$  u kvantizatoru, u slučaju kompresije sa gubicima, ili

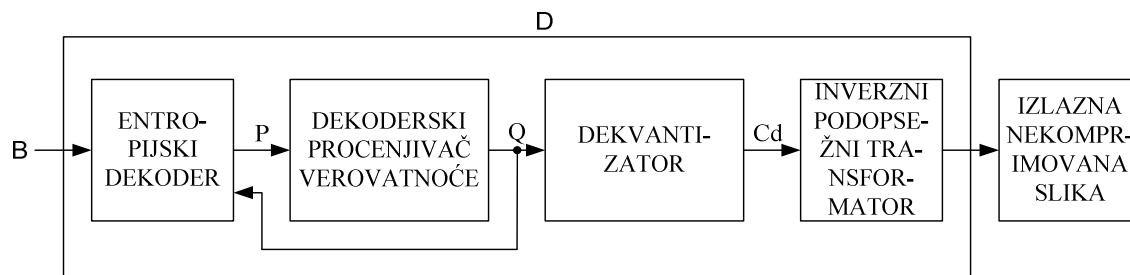
samo propušteni ka enkoderskom procenjivaču verovatnoća, u slučaju kompresije bez gubitaka. Izlazi enkoderskog procenjivača verovatnoća su verovatnoće simbola  $P$  unutar specificiranog konteksta, koje se koriste od strane entropijskog enkodera, da bi se proizvela izlazna komprimovana slika  $A$ .

Na **Slici 1.3** je prikazan blok dijagram dekodera **D** prema postojećem stanju tehnike. Ulazna komprimovana slika  $B$  se prima od strane entropijskog dekodera. Izlaz entropijskog dekodera  $P$  se prihvata od strane dekoderskog procenjivača verovatnoća, da bi se rekonstruisale verovatnoće simbola unutar specificiranog konteksta, i vratile natrag u entropijski dekodera. Izlaz entropijskog dekodera  $Q$  se takođe prima od strane dekvantizatora, da bi se proizveli dekvantizovani transformacioni koeficijenti  $Cd$ , u slučaju kompresije sa gubicima, ili se samo propušta prema inverznom podopsežnom transformatoru, u slučaju kompresije bez gubitaka. Izlaz inverznog podopsežnog transformatora je izlazna nekomprimovana slika.



**Slika 1.2** Blok dijagram enkodera prema postojećem stanju tehnike

Algoritmi za kompresiju prema postojećem stanju tehnike se mogu podeliti u tri osnovne grupe: jednoprolazne, dvoprolazne i višeprolazne. Jednoprolazni algoritmi enkoduju/dekoduju sliku korišćenjem jednog pristupa svakom transformacionom koeficijentu u memoriji [12]. Ovi algoritmi su obično ograničeni prethodnim statističkim modelom sa fiksnim parametrima, što tipično dovodi do nižih stepena kompresije od onih koje postižu ostali postupci.



**Slika 1.3** Blok dijagram dekodera prema postojećem stanju tehnike

Dvoprolazni algoritmi enkoduju/dekoduju sliku pomoću dva pristupa svakom transformacionom koeficijentu u memoriji. Dakle, oni mogu koristiti prethodni statistički model sa promenljivim parametrima, što dovodi do većeg stepena kompresije nego u jednoprolaznom slučaju. Međutim, potrebno je čuvati sve transformacione koeficijente u memoriji, da bi se izvršio drugi prolaz, što zahteva dodatni kapacitet memorije reda ulazne nekomprimovane slike.

Višeprolazni algoritmi enkoduju/dekoduju sliku na osnovu implicitno definisanog statičkog modela (JPEG2000, SPIHT i EZW). JPEG2000 je objašnjen u [13], [14], algoritam sa podelom na grupe u hijerarhijskim stablima (SPIHT) je obelodanjen u [15]-[17]. Podela transformacionih koeficijenata prema alfabetu i grupi je obelodanjena u [18]. Algoritam sa talasićima ("wavelet") i ugrađenim nultim stablima (EZW) je objašnjen u [19]. EZW tehnika je zasnovana na: (1) delimičnom uređivanju transformacionih koeficijenata prema amplitudi korišćenjem grupe oktavno smanjujućih pragova; (2) slanju uređenog niza pomoću algoritma za podelu na podgrupe, koji je dupliciran u dekodera; (3) uređenom prenosu bita za doterivanje po bitskim ravnima; i (4) korišćenju sopstvene sličnosti transformacionih koeficijenata unutar različitih podopsega. Ugrađeni koder slike sa talasićima i predviđanjem (EPWIC), zasnovan na modelu uslovne verovatnoće kao dodatkom na EZW, je obelodanjen u [20], [21]. Svi ovi postupci čuvaju kompletnu sliku u memoriji i zahtevaju relativno veliki broj prolazaka za enkodovanje/dekodovanje slike.

I vreme procesiranja i kapacitet memorije postojećih postupaka za kompresiju slike sa gubicima rastu sa stepenom kompresije. Postojeći mikroprocesori, procesori signala i

čak mikrokontroleri imaju mali kapacitet brze memorije (procesorskih registara opšte namene i unutrašnje ili spoljašnje keš memorije), i veliki kapacitet nekoliko puta sporije memorije (spoljašnje sistemske memorije).

Minimalno iskorišćenje memorije prema postojećem stanju tehnike postignuto je sa enkoderima baziranim na linijama, gde se ulazna nekomprimovana slika čita liniju po liniju [12], [22]-[25].

Uobičajen pristup za smanjenje zahtevanog kapaciteta memorije je podela velike slike na blokove i nezavisno enkodovanje svakog bloka. Svi najbolji postupci za kompresiju nepokretne slike (JPEG2000, JPEG, itd.) i postupci za kompresiju pokretne slike (MPEG-4, MPEG-2, MPEG-1, itd.) prema postojećem stanju tehnike su ili zasnovani na blokovima ili imaju blokovski mod [26].

Metodi za kompresiju slike sa velikom stepenom kompresije prema trenutnom stanju tehnike su dugotrajni i zahtevaju složene procesore i veliku količinu memorije velikog propusnog opsega.

Naučne metode koje su primenjene u izradi ove doktorske disertacije obuhvataju primenu kombinovanih teorijskih, simulacionih i eksperimentalnih metoda u cilju rešavanja definisanih problema.

U okviru ove doktorske disertacije je:

- izvršena optimizacija direktnih filtara, koji izvršavaju direktnu podopsežnu transformaciju, za implementaciju u programabilnim logičkim kolima
- optimizovano izračunavanje konteksta za implementaciju u programabilnim logičkim kolima
- optimizovan estimator verovatnoća simbola zasnovan na adaptivnim histogramima za implementaciju u programabilnim logičkim kolima
- izvršena optimizacija entropijskog enkodera za implementaciju u programabilnim logičkim kolima



- optimizovan protok podataka unutar enkodera za implementaciju u programabilnim logičkim kolima, čime se minimiziraju zahtevani memorijski resursi i propusna moć magistrala, i omogućava realizacija kompletnog enkodera unutar jednog integrisanog kola bez upotrebe spoljašnje memorije

Doktorska disertacija sadrži sedam poglavlja, uključujući uvod kao prvo poglavlje i zaključak kao sedmo poglavlje. U drugom poglavlju je predstavljena analiza i hardverska realizacija direktnog dvodimenzionalnog podoposežnog transformatora. Treće poglavlje sadrži analizu i hardversku realizaciju kvantizatora sa baferovanjem. U četvrtom poglavlju je predstavljena analiza rada i hardverska realizacija enkoderskog procenjivača verovatnoća i entropijskog enkodera. Peto poglavlje sadrži rezultate simulacije hardverskog dvodimenzionalnog enkodera za jednu ravan boje. U šestom poglavlju prikazana je realizacija sistema za enkodovanje slike sa mrežnim interfejsom.

## Direktni dvodimenzionalni podopsežni transformator

**M**oderno kodovanje sa transformacijom sa talasićima sadrži četiri koraka: (1) podospežno razlaganje, (2) kvantizaciju, (3) procenu verovatnoća, i (4) entropijsko kodovanje. Dekodovanje zahteva obrnuti redosled inverznih operacija.

Koeficijenti podospežne transformacije se izračunavaju prvobitnim filtriranjem ulaznog nekomprimovanog signala i izbacivanjem svakog drugog koeficijenta i daljim filtriranjem i izbacivanjem svakog drugog koeficijenta pomoću para nisko propusnih i visokopropusnih filtara. Svaki podopseg se posebno koduje sa bitskom brzinom koja odgovara važnosti odnosnog podopsega. Ovo dovodi do zadovoljavajuće rekonstrukcije signala i minimizacije artefakata kompesije.

Kao što je već navedeno, koncept podopsežne transformacije je prvo uveden za kodovanje govora. Nesavršeni filter za rekonstrukciju sa linearnom fazom je dvoopsežni QMF (Quadrature Mirror Filter) [27]. Grupe filtera za savršenu rekonstrukciju za jednodimenzionalnu (1-D) podopsežnu transformaciju su istraživane u [28]-[34]. Tehnika projektovanja koja dovodi do numerički savršenih grupa filtera za rekonstrukciju razvijena je u [35]. Međutim, takvi filteri su relativno dugački i odatle nepogodni za primene u kodovanju slike. Teorija jednodimenzionalne podopsežne transformacije je proširena na dvodimenzionalni slučaj u [36]-[38], [39] koji koristi 2-D razdvojene QMF grupe, [40]-[42].

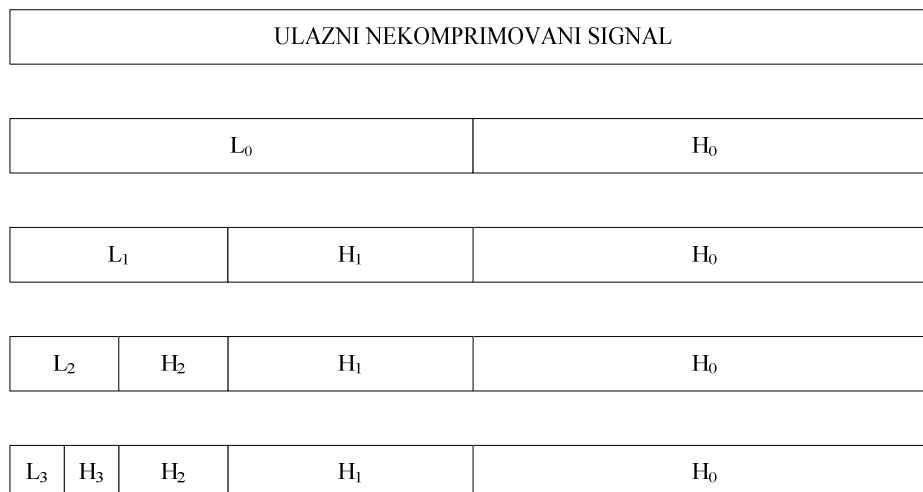
Algoritmi za podopsežnu kompresiju mogu biti podeljeni u tri osnovne grupe: jednoprolazni, dvoprolazni i višeprolazni. Jednoprolazni algoritmi koduju i dekoduju signal korišćenjem jednog pristupa svakom transformacionom koeficijentu u memoriji, kakav je pristup zasnovan na linijama u kombinaciji sa lifting transformacijom sa talasićima, u kome se ulazni nekomprimovani signal čita blok po blok ili liniju po liniju [43]-[49], kako bi se minimizirala veličina potrebne memorije. Jednoprolazni algoritmi su obično ograničeni na prethodne statističke modele sa fiksiranim parametrima, što tipično dovodi do manjeg stepena kompresije od onoga koji se postiže sa drugim metodama.

Dvoprolazni algoritmi enkoduju i dekoduju signal korišćenjem dva pristupa svakom transformacionom koeficijentu u memoriji. Stoga, ovi algoritmi mogu da koriste prethodni statistički model sa promenljivim parametrima, što dovodi do višeg stepena kompresije nego u slučaju sa jednoprolaznim algoritmima. Međutim, potrebno je da se svi transformacioni koeficijenti čuvaju u memoriji, kako bi se sproveo drugi prolazak, što zahteva povećanje veličine memorije reda veličine ulaznog nekomprimovanog signala.

## Direktna jednodimenzionalna podopsežna transformacija

**Slika 2.1** ilustruje rezultate u toku tri nivoa direktne jednodimenzionalne podopsežne transformacije (D1DST), koja se prema postojećem stanju tehnike izvršava

niskopropusnim filtriranjem ulaznih podataka i dvostrukim smanjenjem učestanosti odabiranja, da bi se proizveo podopseg  $L$ , i visokopropusnim filtriranjem ulaznih podataka i dvostrukim smanjenjem učestanosti odabiranja, da bi se proizveo podopseg  $H$ .

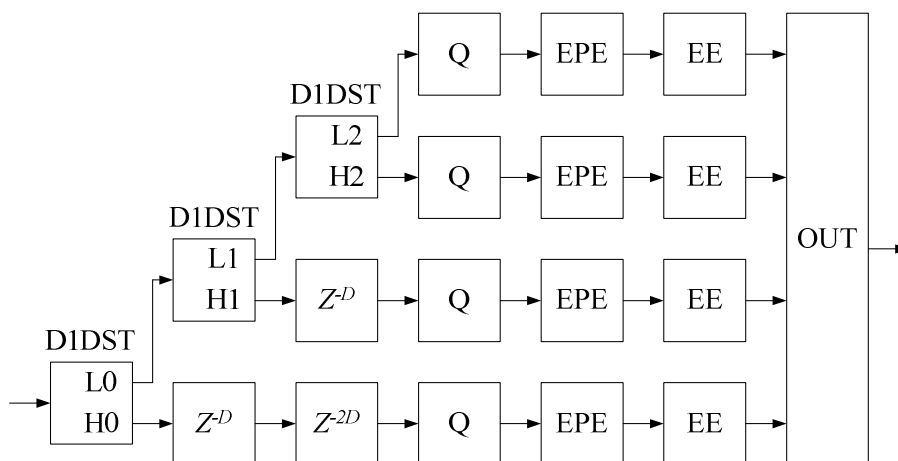


**Slika 2.1** Rezultati jednodimenzionalne podopsežne transformacije

Blok dijagram jednodimenzionog enkodera sa  $N = 3$  nivoa dekompozicije prema postojećem stanju tehnike prikazan je na **Slici 2.2**. Višenivooska D1DST izvršava se primenom jednonivooske D1DST na ulazne nekomprimovane podatke u nivou 0, a zatim narednim izvršavanjem jednonivooske D1DST na podopsegu  $L_i$ , proizvedenom kao rezultat prethodne jednonivooske D1DST. Podopsezi nivoa 0 su  $L_0$  i  $H_0$ . Podopsezi  $L_1$  i  $H_1$  nivoa 1 su proizvedeni primenom filtra propusnika niskih učestanosti i filtra propusnika visokih učestanosti na podopseg  $L_0$ , respektivno. Podopsezi  $L_2$  i  $H_2$  nivoa 2 su proizvedeni primenom filtra propusnika niskih učestanosti i filtra propusnika visokih učestanosti na podopseg  $L_1$ , respektivno, itd.

Izlazi svih D1DST blokova su transformacioni koeficijenti koji se mogu kvantizovati u kvantizatorima  $Q$  u slučaju kompresije sa gubicima, ili jednostavno propustiti do enkoderskih procenjivača verovatnoća  $EPE$  u slučaju kompresije bez gubitaka. Izlazi enkoderskih procenjivača verovatnoća su verovatnoće simbola u okviru odgovarajućeg konteksta, kako bi se proizveo izlazni komprimovani signal u izlaznom baferu  $OUT$  pre prosleđivanja.

Enkoderi prema postojećem stanju tehnike mogu da primene D1DST korišćenjem konvolucionog filtriranja sa filtrom sa konačnim impulsnim odzivom (FIR) neparne dužine  $L = 2 \cdot D + 1$  i simetričnim produženjem ulaznih nekomprimovanih podataka na granicama. Odbirci se primaju serijski sa periodom  $T_p$ . Posle kašnjenja od  $D \cdot T_p$ , ukupno  $L = D + 1 + D$  odbiraka je primljeno zbog produženja, tako da filter započinje generisanje koeficijenata visokopropusnog podopsega  $H_0$  sa periodom  $2 \cdot T_p$ , kao i generisanje koeficijenata niskopropusnog podopsega  $L_0$  sa periodom  $2 \cdot T_p$ , usled dvostrukog smanjenja učestanosti odabiranja.



**Slika 2.2** Jednodimenzionalni enkoder prema postojećem stanju tehnike

Nažalost, FIR filtri sa linearnom fazom su računarski skupi u smislu zahtevanog procesorskog vremena i kapaciteta memorije. Ovi nedostaci mogu biti eliminisani: a) različitim filtrima sa dizanjem (lifting), naizmeničnim ažuriranjem vrednosti neparnih odbiraka signala težinskom sumom parnih odbiraka signala i ažuriranjem vrednosti parnih odbiraka signala težinskom sumom neparnih odbiraka signala [43], [44]; b) posebnim nestacionarnim filtrom koji je prikazan u [50]. Memorija potrebna za filtriranje zavisi od  $D$  i veličine pohranjenog transformacionog koeficijenta, prema **Tabeli 2.1**.

Kašnjenje u nivou 0 D1DST je  $D \cdot T_p$ . Period između generisanja uzastopnih transformacionih koeficijenata u svakom od podopsega  $L_0$  i  $H_0$  je  $2 \cdot T_p$ .

Kašnjenje u nivou 1 D1DST je  $2 \cdot D \cdot T_p$ . Ovo kašnjenje je kompenzovano od strane sinhronizacione memorije  $z^{-D}$  kapaciteta  $D$  za podopseg  $H_0$ , usled perioda  $2 \cdot T_p$  između generisanja uzastopnih transformacionih koeficijenata. Period između generisanja uzastopnih transformacionih koeficijenata u svakom od podopsega  $L_1$  i  $H_1$  je  $4 \cdot T_p$ .

Kašnjenje u nivou 2 D1DST je  $4 \cdot D \cdot T_p$ . Ovo kašnjenje je kompenzovano od strane sinhronizacione memorije  $z^{-2D}$  kapaciteta  $2 \cdot D$  za podopseg  $H_0$ , usled perioda  $2 \cdot T_p$  između generisanja uzastopnih transformacionih koeficijenata, kao i sinhronizacione memorije  $z^{-D}$  kapaciteta  $D$  za podopseg  $H_1$ , usled perioda  $4 \cdot T_p$  između generisanja uzastopnih transformacionih koeficijenata. Period između generisanja uzastopnih transformacionih koeficijenata u svakom od podopsega  $L_2$  i  $H_2$  je  $8 \cdot T_p$ .

**Tabela 2.1**  
Veličina filterske memorije u okviru D1DST

Postupak filtriranja	Kapacitet filterske memorije $F$ po D1DST nivou	Ukupan kapacitet filterske memorije za $N$ D1DST nivoa
Konvolucija	$2 \cdot D + 1$	$(2 \cdot D + 1) \cdot N$
Dizanje [45]	$D + 2$	$(D + 2) \cdot N$
Dizanje [43], [44]	$D + 1$	$(D + 1) \cdot N$
Nestacionarni filtri [50]	$D$	$D \cdot N$

Zahvaljujući kompenzaciji kašnjenja, svi transformacioni koeficijenti postaju istovremeno raspoloživi za kvantizaciju u kvantizatorima, enkodersku procenu verovatnoće u enkoderskim procenjivačima verovatnoća i entropijsko enkodovanje u entropijskim enkoderima. Međutim, u stvarnoj realizaciji, kvantizacija može biti izvedena pre smeštanja u sinhronizacione memorije, da bi se smanjio potreban ukupan kapacitet sinhronizacione memorije, na primer ako su 32-bitni transformacioni koeficijenti u pokretnom zarezu rezultat nepovratne DWT upotrebom (9/7) filtra sa 9-koeficijenata/7-koeficijenata [51], koji je primenjen u JPEG2000 standardu za kompresiju sa gubicima.

Kapaciteti sinhronizacionih memorija u jedinicama transformacionih koeficijenata za enkoder sa  $N = 3$  D1DST nivoa dati su u **Tabeli 2.2**.

Kapaciteti sinhronizacionih memorija u jedinicama transformacionih koeficijenata za enkoder unutar komunikacionog sistema koji koristi  $N$  D1DST nivoa su dati u **Tabeli 2.3**. Kako se povećava broj  $N$  D1DST nivoa, kapacitet sinhronizacione memorije svih D1DST nivoa raste mnogo brže nego kapacitet filterske memorije.

**Tabela 2.2**

Veličina sinhronizacione memorije za enkoder sa  $N = 3$  nivoa D1DST

D1DST nivo	Period ulaznih podataka	Kašnjenje D1DST nivoa	Period izlaznih podataka	Kapacitet sinhronizacione memorije D1DST nivoa 1	Kapacitet sinhronizacione memorije D1DST nivoa 2	Ukupan kapacitet sinhronizacione memorije
0	$T_p$	$D \cdot T_p$	$2 \cdot T_p$	$D$	$2 \cdot D$	$D + 2 \cdot D = 3 \cdot D$
1	$2 \cdot T_p$	$2 \cdot D \cdot T_p$	$4 \cdot T_p$	-	$D$	$D$
2	$4 \cdot T_p$	$4 \cdot D \cdot T_p$	$8 \cdot T_p$	-	-	0
Svi	-	-	-	-	-	$3 \cdot D + D = 4 \cdot D$

Postoji stalna potreba za smanjenje kompleksnosti podopsežnih transformatora, na primer eliminacijom množača u protočnoj arhitekturi sa presavijenim talasićima (*wavelet*) bez množača baziranoj na dizanju za 9/7 i 5/3 filtre [52] i arhitekturi sa bankama filtara bez množača baziranoj na distribuiranoj aritmetici bez tabela za pretraživanje sa dualnim bankama filtara, koje dozvoljavaju korišćenje istog hardvera za visokopropusne i niskopropusne filtre [53].

Na **Slici 2.3** prikazan je blok dijagram jednodimenzionalnog enkodera bez sinhronizacione memorije i sa  $N = 3$  nivoa dekompozicije, koji je objavljen u [54].

Umesto baferisanja komprimovanih transformacionih koeficijenata u sinhronizacionoj memoriji enkodera, ulazni nekomprimovani podaci se direktno podopsežno transformišu u D1DST, transformacioni koeficijenti se kvantizuju u kvantizatorima, verovatnoće transformacionih koeficijenata unutar specificiranih konteksta se procenjuju u enkoderskim procenjivačima verovatnoća, kvantizovani transformacioni koeficijenti se entropijski enkoduju u entropijskim enkoderima i na kraju se u njihovom

## Direktni dvodimenzionalni podopsežni transformator

komprimovanom obliku smeštaju u izlazni komprimovani bafer *OUT*, koji privremeno služi kao sinhronizaciona memorija, iz koga će biti preneti.

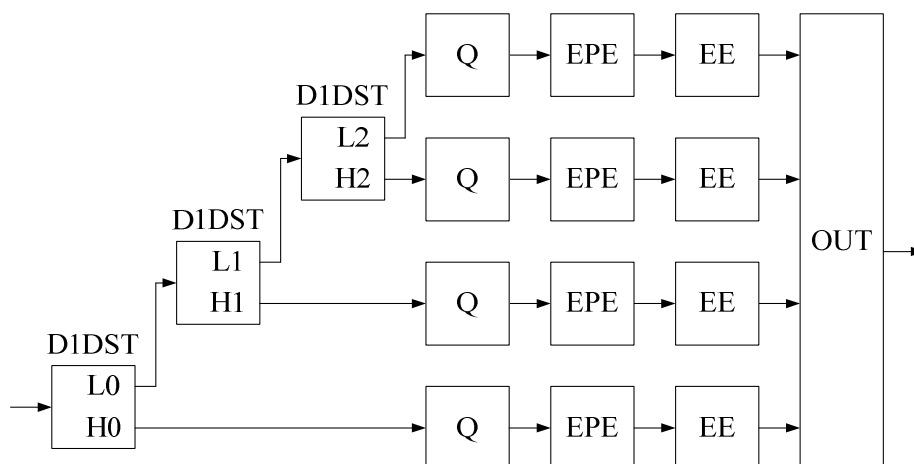
Direktni jednodimenzioni podopsežni transformator prema postojećem stanju tehnike obezbeđuje niskofrekventno filtriranje i dvostruko smanjenje učestanosti odbiraka u gornjoj grani sa **Slike 2.4.** kao i visokofrekventno filtriranje i dvostruko smanjenje učestanosti odbiraka u donjoj grani sa **Slike 2.4.** Svaki drugi odbirak u obe grane se odbacuje tokom decimacije, uprkos memorijskim i procesorskim resursima koji su iskorišćeni za njihovo generisanje.

**Tabela 2.3**  
Veličina sinhronizacione memorije za enkoder

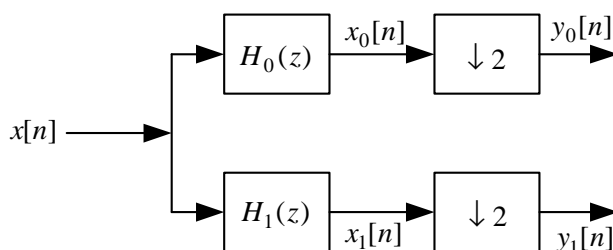
D1DST nivo	Kapacitet sinhronizacione memorije po 1DST nivou [koeficijent]
0	$D + 2 \cdot D + 2^2 \cdot D + \dots + 2^{N-2} \cdot D = \sum_{k=0}^{N-2} 2^k \cdot D = (2^{N-1} - 1) \cdot D$
1	$D + 2 \cdot D + 2^2 \cdot D + \dots + 2^{N-3} \cdot D = \sum_{k=0}^{N-3} 2^k \cdot D = (2^{N-2} - 1) \cdot D$
2	$D + 2 \cdot D + 2^2 \cdot D + \dots + 2^{N-4} \cdot D = \sum_{k=0}^{N-4} 2^k \cdot D = (2^{N-3} - 1) \cdot D$
$n$	$D + 2 \cdot D + 2^2 \cdot D + \dots + 2^{N-n-2} \cdot D = \sum_{k=0}^{N-n-2} 2^k \cdot D = (2^{N-n-1} - 1) \cdot D$
$N - 3$	$D + 2 \cdot D = 3 \cdot D$
$N - 2$	$D$
Svi nivoi	$\sum_{n=0}^{N-2} (2^{N-n-1} - 1) \cdot D = (2^N - N - 1) \cdot D$

D1DST prikazan u [50] koristi neiskorišćene resurse za naizmenično generisanje niskofrekventnih i visokofrekventnih koeficijenata, odnosno isti nestacionarni filter se koristi kao niskofrekventni i visokofrekventni filter u naizmeničnim ciklusima.



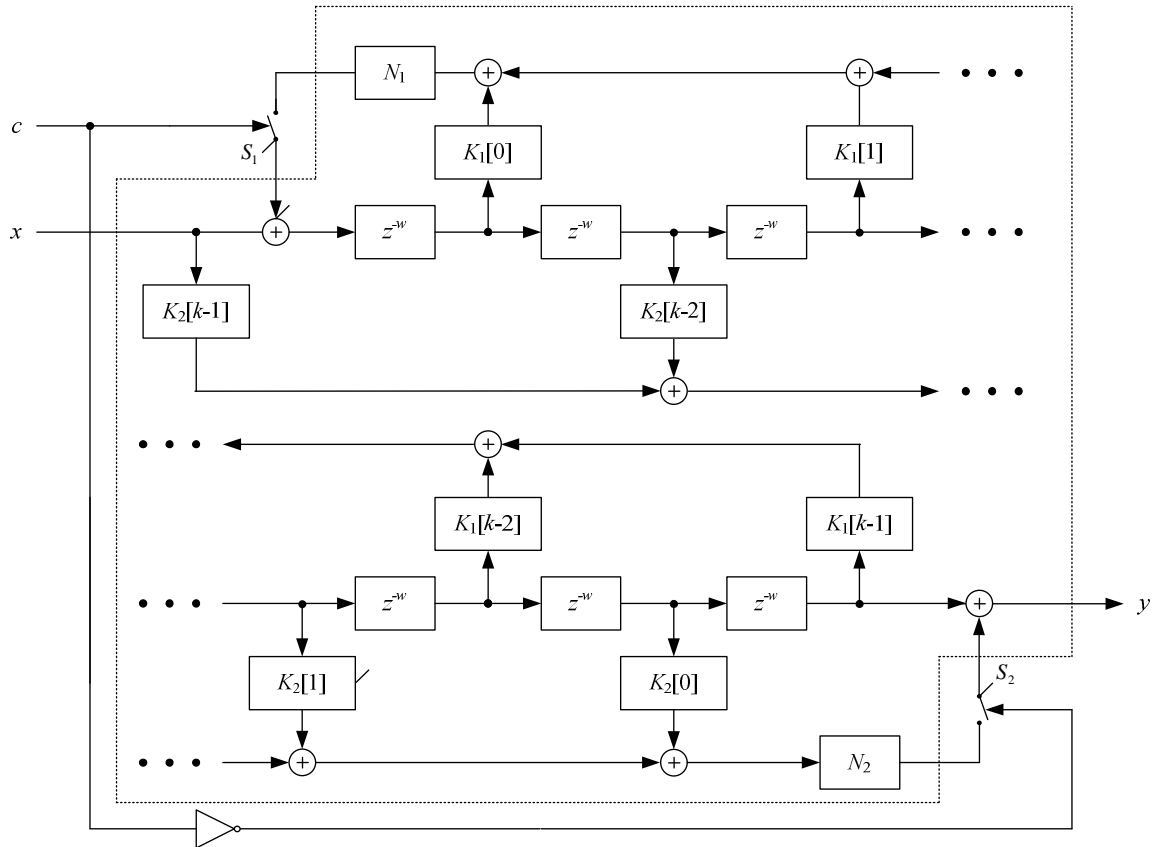


**Slika 2.3** Jednodimenzionalni enkoder sa minimalnim zahtevanim resursima



**Slika 2.4** Jednodimenzioni podpsežni transformator prema postojećem stanju tehnike

Na **Slici 2.5** prikazan je blok dijagram opšte nestacionarne filterske ćelije (NSFC) iz [55], koja je osnova nestacionarnog filtra iz [50]. NSFC obezbeđuje oktavno podopsežno razlaganje, pri čemu očuvava linearnu fazu, što je suprotno od stacionarnih filtara sa beskonačnim impulsnim odzivom (IIR) prema postojećem stanju tehnike. Dodatno, realizacija nestacionarnih filtara je dvostruko jednostavnija od realizacije ekvivalentnog FIR filtra. Pored toga, nestacionarni filtri obezbeđuju naizmenično niskopropusno i visokopropusno filtriran rezultat na svom izlazu, sa parnim i neparnim indeksima odbiraka, respektivno, što dodatno smanjuje složenost realizacije. Ukupan broj elemenata za kašnjenje je obično oko dva puta manji nego kod dva FIR filtra koji zajednički obavljaju isti zadatak (videti **Tabelu 2.8** napred). Konačno, direktni nestacionarni filtar ne koristi množače jer su filterski koeficijenti stepena dvojke.



Slika 2.5 Opšta nestacionarna filterska ćelija

Opšta NSFC  $F_{N_1, N_2, \vec{K}_1, \vec{K}_2}(x, c)$  ima dva NSFC ulaza,  $x$  i  $c$ , kao i jedan NSFC izlaz  $y$ . Pri tome  $x$  i  $y$  predstavljaju ulaz i izlaz filterskog sklopa, čijom nestacionarnošću upravlja ulaz  $c$  preko dva prekidača  $S_1$  i  $S_2$ . Ulazni odbirci se primaju serijski, po jedan odbirak svakog perioda na ulazu  $x$ . Ulaz  $c$  je  $c=0$  za odbirke sa parnim indeksima i  $c=1$  za odbirke sa neparnim indeksima. Izlazni odbirci sa parnim indeksima su niskopropusno filtrirani i sa dvostruko smanjenom učestanošću odabiranja. Izlazni odbirci sa neparnim indeksima su visokopropusno filtrirani i sa dvostruko smanjenom učestanošću odabiranja. Specifične realizacije zavise od izabranih parametara  $N_1$ ,  $N_2$ ,  $\vec{K}_1$  i  $\vec{K}_2$ , gde su  $N_1$  i  $N_2$  funkcije,  $\vec{K}_1$  je konstantni vektor sa elementima  $K_1[0], K_1[1], \dots, K_1[k-2], K_1[k-1]$ , i  $\vec{K}_2$  je konstantni vektor sa elementima  $K_2[0], K_2[1], \dots, K_2[k-2], K_2[k-1]$ .

Objašnjenje rada opšte NSFC biće urađeno na primeru filtera sa neparnim brojem elemenata za kašnjenje  $z^{-w}$  kojih ukupno ima  $m-1$ ,  $m = 2 \cdot k$ . Izlazi parnih elemenata za kašnjenje  $z^{-w}$  se usmeravaju kroz množače  $K_1[0], K_1[1], \dots, K_1[k-1]$ , koji se zamenjuju pomeračima. Izlazi množača  $K_1[0], K_1[1], \dots, K_1[k-1]$  se zajedno sabiraju korišćenjem sabirača u gornjoj grani, zbir se transformiše u  $N_1$ , vodi se povratnom vezom i konačno se sabira sa ulazom  $x$  za zatvoren prekidač  $S_1$ , tj. pri  $c=1$ . Ulaz  $x$  NSFC i izlazi neparnih elemenata za kašnjenje  $z^{-w}$  se usmeravaju kroz množače  $K_2[k-1], K_2[k-2], \dots, K_2[1], K_2[0]$ , koji se zamenjuju pomeračima. Izlazi množača  $K_2[k-1], K_2[k-2], \dots, K_2[1], K_2[0]$  se zajedno sabiraju korišćenjem sabirača iz donje grane, zbir se transformiše u  $N_2$ , vodi se vezom unapred i konačno se sabira sa izlazom poslednjeg elementa za kašnjenje  $z^{-w}$  za zatvoren prekidač  $S_2$ , tj. pri  $c=0$ , da bi se proizveo izlaz  $y$  NSFC.

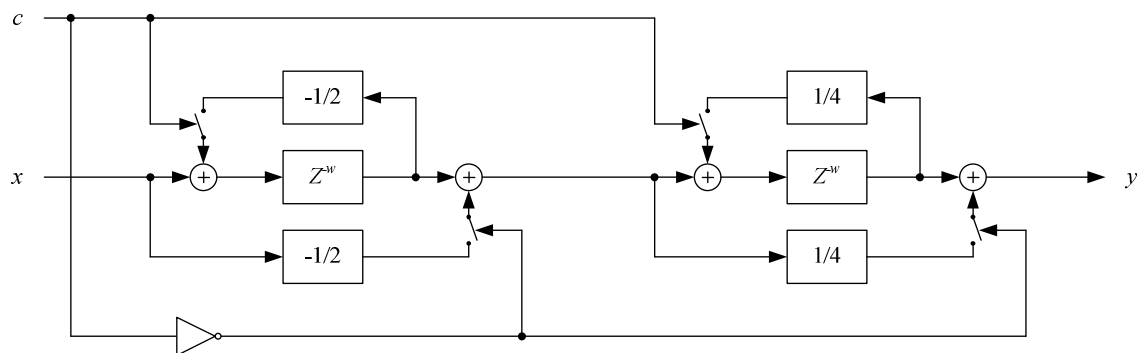
Ako je direktni NSFC definisan sa  $F_{N_1, N_2, \vec{K}_1, \vec{K}_2}(x, c)$ , inverzni NSFC je definisan sa  $F_{-N_1, -N_2, \vec{K}_1, \vec{K}_2}(x, \bar{c})$ , gde  $-N_1$  i  $-N_2$  predstavljaju funkcije sa negativnim predznakom, dok  $\bar{c}$  predstavlja invertovanu binarnu promenljivu  $c$ , koja je 1 za odbirke sa parnim indeksima i 0 za odbirke sa neparnim indeksima. Očigledno je da i direktna i inverzna NSFC imaju istu strukturu kao opšta NSFC. Serijska veza direktne i inverzne NSFC obezbeđuje funkciju prenosa  $z^{-2 \cdot w \cdot (m-1)}$ , tj. vremensko kašnjenje od  $2 \cdot w \cdot (m-1)$  odbiraka, gde je  $m-1$  broj elemenata za kašnjenje  $z^{-w}$  u svakoj od direktne i inverzne NSFC. Element za kašnjenje  $z^{-w}$  unutar jednodimenzionog filtra mora da čuva samo jedan piksel, tj.  $w=1$ .

Moguće je pokazati da su dovoljni uslovi za obezbeđenje linearne faze u NSFC  $N_1 = N_2 = const$  i  $\vec{K}_1 = \vec{K}_2$ . Međutim, pomenuti uslovi nisu potrebni. Na primer, pokazuje se da  $F_{1,1,[1],[1]}$  takođe obezbeđuje linearnu fazu.

**Slika 2.6** je blok dijagram direktnog nestacionarnog filtra napravljenog od dve redno povezane direktne NSFC prvog reda  $F_1 = F_{-1,-1,[1/2],[1/2]}(x, c)$  i  $F_2 = F_{1,1,[1/4],[1/4]}(x, c)$ .

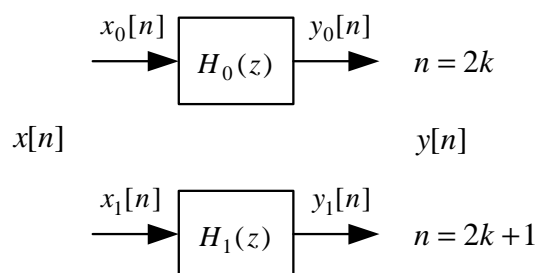
## Direktni dvodimenzionalni podopsežni transformator

Uprkos potpuno različitom pristupu, funkcija prenosa ovog filtra je odgovarajuća (5/3) filtru za analizu sa 5-koeficijentna/3-koeficijenta, obelodanjenom u [42], koji se koristi za reverzibilnu DWT u JPEG2000 standardu [56].



**Slika 2.6** Nestacionarni D1DST 5/3 filter

Na **Slici 2.7** prikazan je blok dijagram D1DST sa **Slike 2.6** u dva susedna ciklusa, Ulazni odbirci se prihvataju serijski, jedan uzorak po ciklusu.



**Slika 2.7** Nestacionarni D1DST u dva susedna ciklusa

Parni uzorci  $x_0[n]$  se filtriraju bez dvostrukog smanjenja učestanosti, čime se generišu izlazni koeficijenti  $y_0[n]$ .

$$y_0[n] = -\frac{1}{8}x[n] + \frac{1}{4}x[n-1] + \frac{3}{4}x[n-2] + \frac{1}{4}x[n-3] - \frac{1}{8}x[n-4]$$

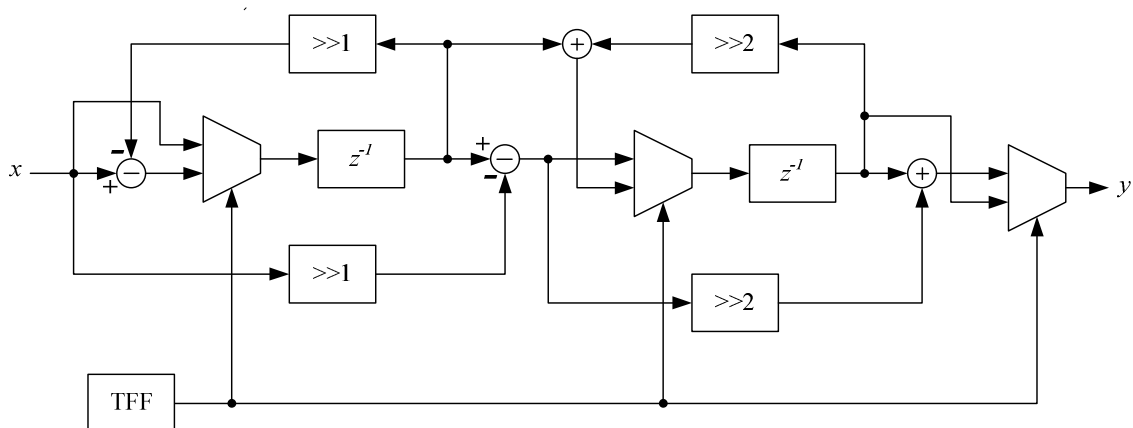
Neparni uzorci  $x_1[n]$  se filtriraju bez dvostrukog smanjenja učestanosti, čime se generišu izlazni koeficijenti  $y_1[n]$ .

$$y_1[n] = -\frac{1}{2}x[n-1] + x[n-2] - \frac{1}{2}x[n-3]$$

Množači u nestacionarnom filtru mogu biti napravljeni bilo kao pomerači ili pomerena ožičenja bitskih veznih linija (pomerene veze između izlaznih i ulaznih bitskih linija) u hardveru, ili pomeračke instrukcije ili strukture podataka za remapiranje bita (bitska polja u C programskom jeziku) u softveru.

Sva filtarska ostvarenja koriste simetrično proširenje ulaznih podataka na granicama slike. Međutim, moguće je realizovati nestacionarne koeficijente filtra blizu granica slike, da bi se proizveo isti efekat.

Na **Slici 2.8** prikazana je hardverska realizacija D1DST nestacionarnog filtra sa **Slike 2.6**. U **Tabeli 2.4** prikazano je poređenje performansi ove hardverske realizacije sa ekvivalentnim transponovanim konvolucionim i filtrom sa dizanjem prema postojećem stanju tehnike, u integrisanom kolu Altera EP4CE115F29C7 korišćenjem softverskog alata Quartus II 12.1, a za širinu podataka od 19 bita. Iz tabele se vidi da je realizacija D1DST sa nestacionarnim filtrom 60% jednostavnija nego realizacija koja koristi ekvivalentni transponovani konvolucionni filter, i 10% jednostavnija nego realizacija koja koristi ekvivalentni filter sa dizanjem. Maksimalna radna frekvencija bez ograničenja je u isto vreme između 40% i 120% viša. Radna frekvencija je ulazno/izlaznim blokovima ograničena na 250MHz.



**Slika 2.8** Hardverska realizacija nestacionarnog filtra

**Tabela 2.4**  
Performanse D1DST u Altera EP4CE115F29C7

D1DST @ 85 <sup>0</sup> C	Konvolucija	Dizanje	Nestacionarni filter
Iskorišćenje integrisanog kola [LC]	163	114	103
Maksimalna učestanost bez ograničenja [MHz]	211.86	133.92	291.63

## Direktna dvodimenzionalna podopsežna transformacija

Kako je već rečeno u Uvodu, minimalno iskorišćenje memorije prema postojećem stanju tehnike postignuto je sa enkoderima baziranim na linijama, gde se ulazna nekomprimovana slika čita liniju po liniju. Na drugoj strani, maksimalno iskorišćenje memorije prema postojećem stanju tehnike zahteva se od strane višeprolaznih kodeka koji enkoduju/dekoduju sliku na osnovu implicitno definisanog statičkog modela: EZW, SPIHT, EPWIC i JPEG2000. Uobičajen pristup za smanjenje zahtevanog kapaciteta memorije je podela velike slike na blokove i nezavisno enkodovanje svakog bloka. Svi najbolji postupci za kompresiju nepokretne slike (JPEG2000, JPEG, itd) i postupci za kompresiju pokretne slike (MPEG-4, MPEG-2, MPEG-1) prema postojećem stanju tehnike su ili zasnovani na blokovima ili imaju blokovski mod.

JPEG2000 enkoder prvo deli ulaznu nekomprimovanu sliku u nepreklapajuće blokove, zatim nezavisno rekurzivno podopsežno transformiše svaki blok pomoću direktne diskretne transformacije sa talasićima (DWT) [14]. Transformacioni koeficijenti se zatim kvantizuju i entropijski koduju, pre uobličavanja izlaznog kodiranog niza. Ulazni kodirani niz se u dekoderu prvo entropijski dekoduje, dekvantizuje i rekurzivno podopsežno transformiše u nezavisne blokove korišćenjem inverzne DWT, da bi se proizvela rekonstruisana slika. Međutim, podela na blokove proizvodi blokovske artefakte na granicama blokova. Ovaj nedostatak se delimično otklanja uokviravanjem, tj. preklapanjem susednih blokova za bar jedan piksel. Drugi ozbiljan nedostatak je opadanje kvaliteta pri većim stepenima kompresije i ograničen maksimalan prihvatljiv stepen kompresije.

JPEG2000 standard podržava dva režima filtriranja: konvoluciju i dizanje. Potrebno je prvo periodično proširiti signal na oba kraja za polovinu širine filtra [57]. Filtriranje zasnovano na konvoluciji se sastoji od izvršavanja serije množenja između koeficijenata filtera propusnika niskih učestanosti i propusnika visokih učestanosti sa odbircima proširenog 1-D signala [51]. Filtriranje zasnovano na dizanju se sastoji od niza naizmeničnih ažuriranja vrednosti odbiraka sa neparnim indeksima sa težinskim zbirom vrednosti odbiraka sa parnim indeksima, i ažuriranja vrednosti odbiraka sa parnim indeksima sa težinskim zbirom odbiraka sa neparnim indeksima [58].

JPEG standard je objašnjen u [59], [60]. Originalna slika se deli na blokove dimenzije  $8 \times 8$ , koji se odvojeno transformišu korišćenjem DCT. Posle transformacije, 64 transformaciona koeficijenta se kvantizuju različitim koracima kvantizacije da bi se uzeo u obzir različit značaj svakog transformacionog koeficijenta, korišćenjem malog koraka kvantizacije za koeficijente niskih učestanosti u odnosu na one za koeficijente visokih učestanosti. Transformacioni koeficijenti se zatim koduju korišćenjem bilo Hafmanovog bilo aritmetičkog kodiranja. Nezavisno kvantizovanje blokova prouzrokuje blokovski efekat. JPEG kompresija bez gubitaka ne koristi transformaciju, već predviđanje za odstranjivanje suvišne informacije između susednih piksela. Greška predviđanja se kodira pomoću Hafmanovog koda. Stepen kompresije je oko 2:1 za prirodne slike.

JPEG implementacija sa malim zahtevanim memorijskim resursima, opisana je u [61]. Ova implementacija ima niži kvalitet kompresije od kompresije zasnovane na podopsežnoj transformaciji. Implementacije podopsežne transformacije sa malim zahtevanim memorijskim resursima zasnivaju se na filteru sa frakcionim talasićima [62], [63] uz sistem za kodovanje slike sa malom memorijom [64], što omogućava izbor između manje veličine RAM memorije, značajno veću fleš memoriju i nekoliko puta više računskih operacija.

MPEG-4 standard za kompresiju pokretnih slika zasnovan je na objektima i omogućava kompresiju niza slika korišćenjem zajedničkog kodiranja više slika. Međutim, njegovo kodiranje unutar jedne slike predstavlja postupak kodiranja nepokretne slike, veoma

sličan JPEG. Granični ram objekta koji treba kodirati se deli na makroblokove dimenzije  $16 \times 16$ , koji sadrže četiri bloka od  $8 \times 8$  piksela za osvetljaj i dva bloka od  $8 \times 8$  piksela za boju odabiranih sa nižom učestanošću. DCT se odvojeno izvršava za svaki od blokova unutar makrobloka, koeficijenti se kvantizuju, cik-cak skaniraju i entropijski kodiraju pomoću “run-length” i Hafmanovog postupka [2], [26].

**Slika 2.9** ilustruje rezultate unutar tri nivoa direktne dvodimenzionalne podopsežne transformacije (D2DST).

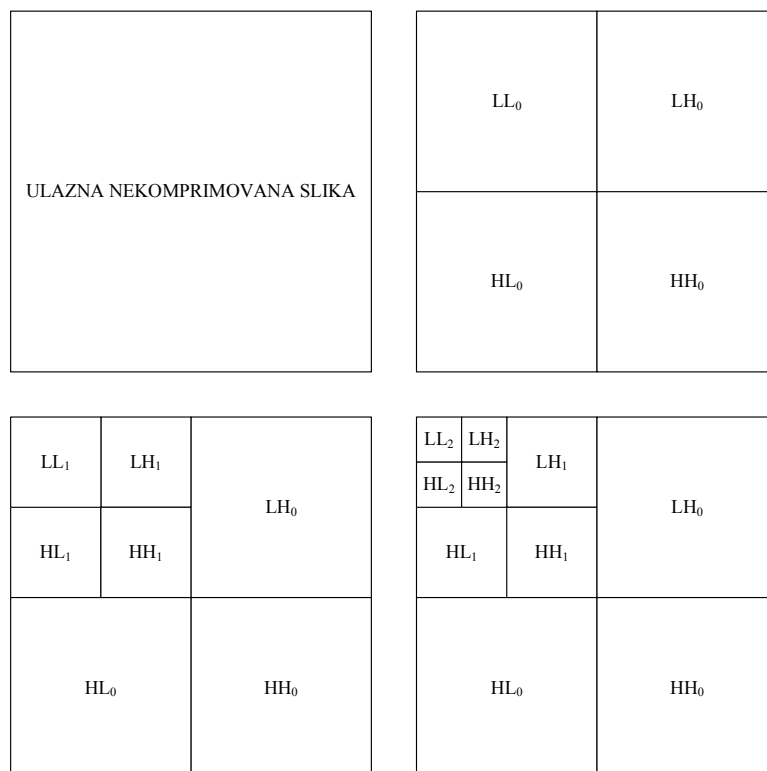
Jednonivooska D2DST prema postojećem stanju tehnike se izvršava razdvojenom primenom D1DST, prvo horizontalno duž redova a zatim vertikalno duž kolona. Rezultati jednonivooske D2DST su četiri podopsega: LL, LH, HL i HH.

Podopseg LL odgovara niskopropusnom filtriranju duž redova i niskopropusnom filtriranju duž kolona, i sadrži istovremeno informaciju o niskim učestanostima horizontalno i informaciju o niskim učestanostima vertikalno. Veći deo energije tipične slike se nalazi u ovom podopsegu.

Podopseg LH odgovara niskopropusnom filtriranju duž redova i visokopropusnom filtriranju duž kolona, i sadrži istovremeno informaciju o niskim učestanostima horizontalno i informaciju o visokim učestanostima vertikalno, tj. informaciju o horizontalnim ivicama.

Podopseg HL odgovara visokopropusnom filtriranju duž redova i niskopropusnom filtriranju duž kolona, i sadrži istovremeno informaciju o visokim učestanostima horizontalno i informaciju o niskim učestanostima vertikalno, tj. informaciju o vertikalnim ivicama.

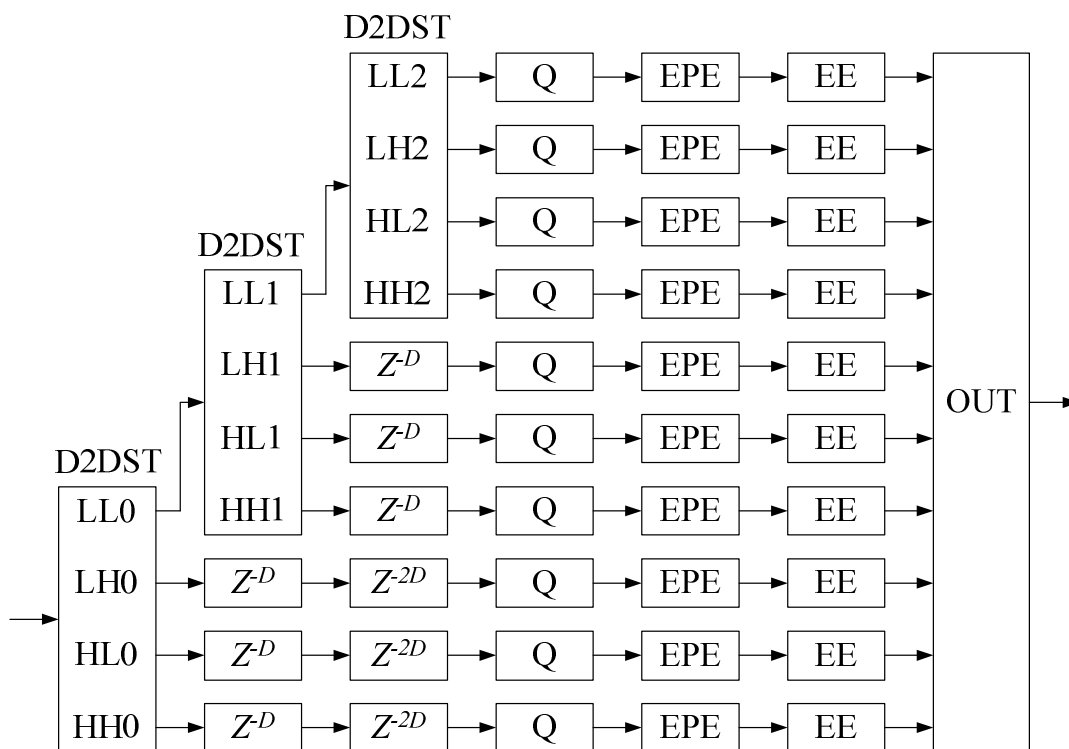




**Slika 2.9** Rezultati dvodimenzionalne podopsežne transformacije

Podopseg HH odgovara visokopropusnom filtriranju duž redova i visokopropusnom filtriranju duž kolona, i sadrži istovremeno informaciju o visokim učestanostima horizontalno i informaciju o visokim učestanostima vertikalno, tj. informaciju o dijagonalnim ivicama.

Blok dijagram dvodimenzionog enkodera sa  $N = 3$  nivoa dekompozicije prema postojećem stanju tehnike prikazan je na **Slici 2.10**. Višenivooska D2DST prema postojećem stanju tehnike se izvršava primenom jednonivooske D2DST na ulaznu nekomprimovanu sliku na nivou 0, i zatim narednim izvršavanjem jednonivooske D2DST na podopsegu LL<sub>*i*</sub>, proizvedenom kao rezultat prethodne jednonivooske D2DST. Podopsezi nivoa 0 su LL<sub>0</sub>, LH<sub>0</sub>, HL<sub>0</sub> i HH<sub>0</sub>. Podopsezi nivoa 1 LL<sub>1</sub>, LH<sub>1</sub>, HL<sub>1</sub> i HH<sub>1</sub> su proizvedeni primenom D2DST na podopseg LL<sub>0</sub>. Podopsezi nivoa 2 LL<sub>2</sub>, LH<sub>2</sub>, HL<sub>2</sub> i HH<sub>2</sub> su proizvedeni primenom D2DST na podopseg LL<sub>1</sub>, itd.



Slika 2.10 Dvodimenzionalni enkoder prema postojećem stanju tehnike

Izlazi svih D2DST blokova su transformacioni koeficijenti koji se mogu kvantizovati u kvantizatorima  $Q$  u slučaju kompresije sa gubicima, ili jednostavno propustiti do enkoderskih estimatora verovatnoća  $EPE$  u slučaju kompresije bez gubitaka. Izlazi enkoderskih procenjivača verovatnoća su verovatnoće simbola u okviru odgovarajućeg konteksta, kako bi se proizveo izlazni komprimovani signal u izlaznom baferu  $OUT$  pre prosleđivanja.

Po završetku 1DST horizontalnog filtriranja na nivou 0, kompletne linije transformisanih koeficijenata se smeštaju u memoriju, što zahteva kapacitet memorije  $W$  za svaku liniju, gde je  $W$  broj piksela u svakoj liniji, tj. širina slike. Širina podopsežne slike se dvostruko smanjuje sa svakim inkrementom 2DST nivoa. Odatle, svaki podopseg  $LL_{i+1}$  na 2DST nivou  $i+1$  zahteva memoriju za polovinu linija u odnosu na prethodni podopseg  $LL_i$  na 2DST nivou  $i$ , prema **Tabeli 2.5**, gde je  $F = 2 \cdot D + 1$  za konvoluciju,  $F = D + 1$  za dizanje, i  $F = D$  za nestacionarne filtre [50].

Ukoliko se ulazna nekomprimovana slika čita linija po linija, redosled po kome se podopsežni transformacioni koeficijenti generišu od strane direktne 2DST su skoro

suprotni redosledu koji očekuje inverzna 2DST, što zahteva dodatnu sinhronizacionu memoriju u enkoderu (kao i u dekoderu). Ukupan kapacitet sinhronizacione memorije je obično podjednako podeljen između enkodera i dekodera, mada je moguće je pridružiti kompletnu sinhronizacionu memoriju samo enkoderu ili samo dekoderu, zavisno od specifične primene. Dalje izlaganje će biti posvećeno samo simetričnoj sinhronizacionoj memoriji podjednagog kapaciteta u enkoderu i dekoderu.

**Tabela 2.5**  
Veličina filterske memorije u okviru D2DST

2DST nivo	Kapacitet filterske memorije po 2DST nivou [interna promenljiva]
0	$F \cdot W$
1	$\frac{F \cdot W}{2}$
2	$\frac{F \cdot W}{4}$
$n$	$\frac{F \cdot W}{2^n}$
$N - 1$	$\frac{F \cdot W}{2^{N-1}}$
Svi nivoi	$\sum_{n=0}^{N-1} \frac{F \cdot W}{2^n} = 2 \cdot F \cdot W \cdot \left(1 - \frac{1}{2^N}\right) < 2 \cdot F \cdot W$

Period između prijema uzastopnih linija ulazne nekomprimovane slike je  $T_L$ . Kašnjenje nivoa 0 D2DST je  $D \cdot T_L$ . Period između generisanja uzastopnih linija u podopsegu  $LL_0$  je  $2 \cdot T_L$  a širina svake linije je  $W/2$ . Kašnjenje u nivou 1 D2DST je  $2 \cdot D \cdot T_L$ . Ovo kašnjenje je kompenzovano od strane sinhronizacionih memorija  $z^{-D}$  kapaciteta  $D \cdot W/2$  za svaki od podopsega  $LH_0$ ,  $HL_0$  i  $HH_0$ , usled perioda  $2 \cdot T_L$  između generisanja uzastopnih linija i širine  $W/2$  svake linije. Period između generisanja uzastopnih linija u svakom od podopsega  $LL_1$ ,  $LH_1$ ,  $HL_1$  i  $HH_1$  je  $4 \cdot T_L$  a širina svake linije je  $W/4$ .

Kašnjenje u nivou 2 D2DST je  $4 \cdot D \cdot T_L$ . Ovo kašnjenje je kompenzovano od strane sinhronizacionih memorija  $z^{-2D}$  kapaciteta  $2 \cdot D \cdot W/2$  za svaki od podopsega  $LH_0$ ,  $HL_0$  i  $HH_0$ , usled perioda  $2 \cdot T_L$  između generisanja uzastopnih linija i širine  $W/2$  svake

linije, kao i sinhronizacionih memorija  $z^{-D}$  kapaciteta  $D \cdot W/4$  za svaki od podopsega LH<sub>1</sub>, HL<sub>1</sub> i HH<sub>1</sub>, usled perioda  $4 \cdot T_L$  između generisanja uzastopnih linija i širine  $W/4$  svake linije. Period između generisanja uzastopnih linija u svakom od podopsega LL<sub>2</sub>, LH<sub>2</sub>, HL<sub>2</sub> i HH<sub>2</sub> je  $8 \cdot T_L$ .

Zahvaljujući kompenzaciji kašnjenja, svi transformacioni koeficijenti postaju istovremeno raspoloživi za kvantizaciju u kvantizatorima, enkodersku procenu verovatnoće u enkoderskim procenjivačima verovatnoća i entropijsko enkodovanje u entropijskim enkoderima. Međutim, u stvarnom izvođenju, kvantizacija može biti izvedena pre smeštanja u sinhronizacione memorije, da bi se smanjio potreban ukupan kapacitet sinhronizacione memorije.

Kapaciteti sinhronizacionih memorija za enkoder (ili dekoder) sa  $N = 3$  2DST nivoa su dati u **Tabeli 2.6**.

**Tabela 2.6**  
Kapacitet sinhronizacione memorije za enkoder sa  $N = 3$  2DST nivoa

2DST nivo	Period ulazne linije	Kašnjenje 2DST nivoa	Period izlazne linije	Kapacitet sinhronizacione memorije 2DST nivoa 1	Kapacitet sinhronizacione memorije 2DST nivoa 2	Ukupan kapacitet sinhronizacione memorije
0	$T_L$	$D \cdot T_L$	$2 \cdot T_L$	$D \cdot W/2$	$2 \cdot D \cdot W/2$	$3 \cdot D \cdot W/2$
1	$2 \cdot T_L$	$2 \cdot D \cdot T_L$	$4 \cdot T_L$	-	$D \cdot W/4$	$D \cdot W/4$
2	$4 \cdot T_L$	$4 \cdot D \cdot T_L$	$8 \cdot T_L$	-	-	0
Svi	-	-	-	-	-	$7 \cdot D \cdot W/4$

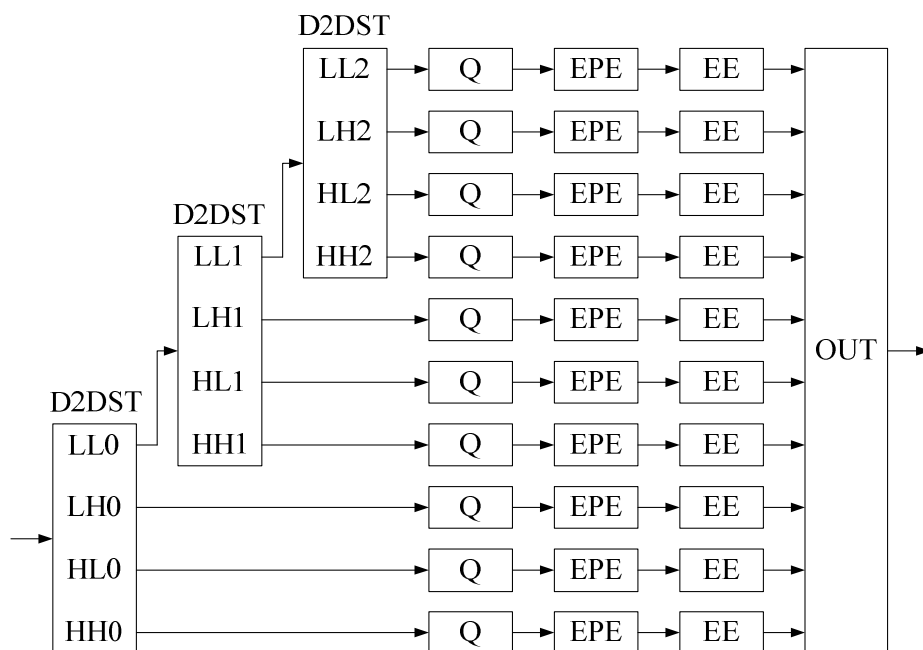
Kapacitet sinhronizacionih memorija u jedinicama transformacionih koeficijenata za enkoder sa  $N$  2DST nivoa je dat u **Tabeli 2.7**. Ukupan kapacitet sinhronizacione memorije se izvodi množenjem poslednjeg rezultata iz **Tabele 2.7**, sa brojem bajtova jednog transformacionog koeficijenta, odnosno brojem bajtova jednog kvantizovanog koeficijenta, ako se kvantizacija obavlja pre biferisanja transformacionih koeficijenata.

**Tabela 2.7**  
Kapacitet sinhronizacione memorije za enkoder

2DST nivo	Ukupan kapacitet sinhronizacione memorije po podopsegu [koeficijent] Kapacitet sinhronizacione memorije za 3 podopsega (LH, HL i HH) je 3 puta veći
0	$(D + 2 \cdot D + 2^2 \cdot D + \dots + 2^{N-2} \cdot D) \cdot \frac{W}{2} = \frac{W}{2} \cdot \sum_{k=0}^{N-2} 2^k \cdot D = (2^{N-1} - 1) \cdot D \cdot \frac{W}{2}$
1	$(D + 2 \cdot D + 2^2 \cdot D + \dots + 2^{N-3} \cdot D) \cdot \frac{W}{4} = \frac{W}{4} \cdot \sum_{k=0}^{N-3} 2^k \cdot D = (2^{N-2} - 1) \cdot D \cdot \frac{W}{4}$
2	$(D + 2 \cdot D + 2^2 \cdot D + \dots + 2^{N-4} \cdot D) \cdot \frac{W}{8} = \frac{W}{8} \cdot \sum_{k=0}^{N-4} 2^k \cdot D = (2^{N-3} - 1) \cdot D \cdot \frac{W}{8}$
$n$	$(D + 2 \cdot D + 2^2 \cdot D + \dots + 2^{N-n-2} \cdot D) \cdot \frac{W}{2^{n+1}} = \frac{W}{2^{n+1}} \cdot \sum_{k=0}^{N-n-2} 2^k \cdot D = (2^{N-n-1} - 1) \cdot D \cdot \frac{W}{2^{n+1}}$
$N - 3$	$(D + 2 \cdot D) \cdot \frac{W}{2^{N-2}} = 3 \cdot D \cdot \frac{W}{2^{N-2}}$
$N - 2$	$D \cdot \frac{W}{2^{N-1}}$
Svi nivoi	$\sum_{n=0}^{N-2} (2^{N-n-1} - 1) \cdot D \cdot \frac{W}{2^{n+1}} = \frac{1}{3} \cdot \left( 2^N - 3 + \frac{1}{2^{N-1}} \right) \cdot D \cdot W$
Svi nivoi i svi podopsezi	$3 \cdot \sum_{n=0}^{N-2} (2^{N-n-1} - 1) \cdot D \cdot \frac{W}{2^{n+1}} = \left( 2^N - 3 + \frac{1}{2^{N-1}} \right) \cdot D \cdot W$

Na **Slici 2.11** prikazan je blok dijagram dvodimenzionalnog enkodera bez sinhronizacione memorije i sa  $N = 3$  nivoa dekompozicije, koji je objavljen u [65].

Umesto baferisanja komprimovanih transformacionih koeficijenata u sinhronizacionoj memoriji enkodera, ulazna nekomprimovana slika se direktno podopsežno transformiše u D2DST, korišćenjem direktnih nestacionarnih filtara, transformacioni koeficijenti se kvantizuju u kvantizatorima, verovatnoće transformacionih koeficijenata unutar specificiranih konteksta se procenjuju u enkoderskim procenjivačima verovatnoća, kvantizovani transformacioni koeficijenti se entropijski enkoduju u entropijskim enkoderima i na kraju u njihovom komprimovanom obliku se smeštaju u izlazni komprimovani bafer *OUT*, iz koga će izlazna komprimovana slika biti preneti.



**Slika 2.11** Dvodimenzionalni enkoder sa minimalnim zahtevanim resursima

Kako se memorija izlazne komprimovane slike *OUT* ne smatra delom sistema za kompresiju, dvodimenzionalni enkoder iz [65] ne zahteva sinhronizacionu memoriju, prema **Tabeli 2.8**. Pored toga, filtriranje kolona započinje čim je filtriran dovoljan broj horizontalnih linija (**Tabela 2.1**), tako da su potrebne svega  $D$  filtrirane horizontalne linije da bi započelo filtriranje kolona.

Primećuje se da se enkoder sa **Slike 2.11** ne može predstaviti u obliku blokova sa **Slike 1.2**, usled jako distribuiranih kvantizatora, enkoderskih procenjivača verovatnoća i entropijskih enkodera.

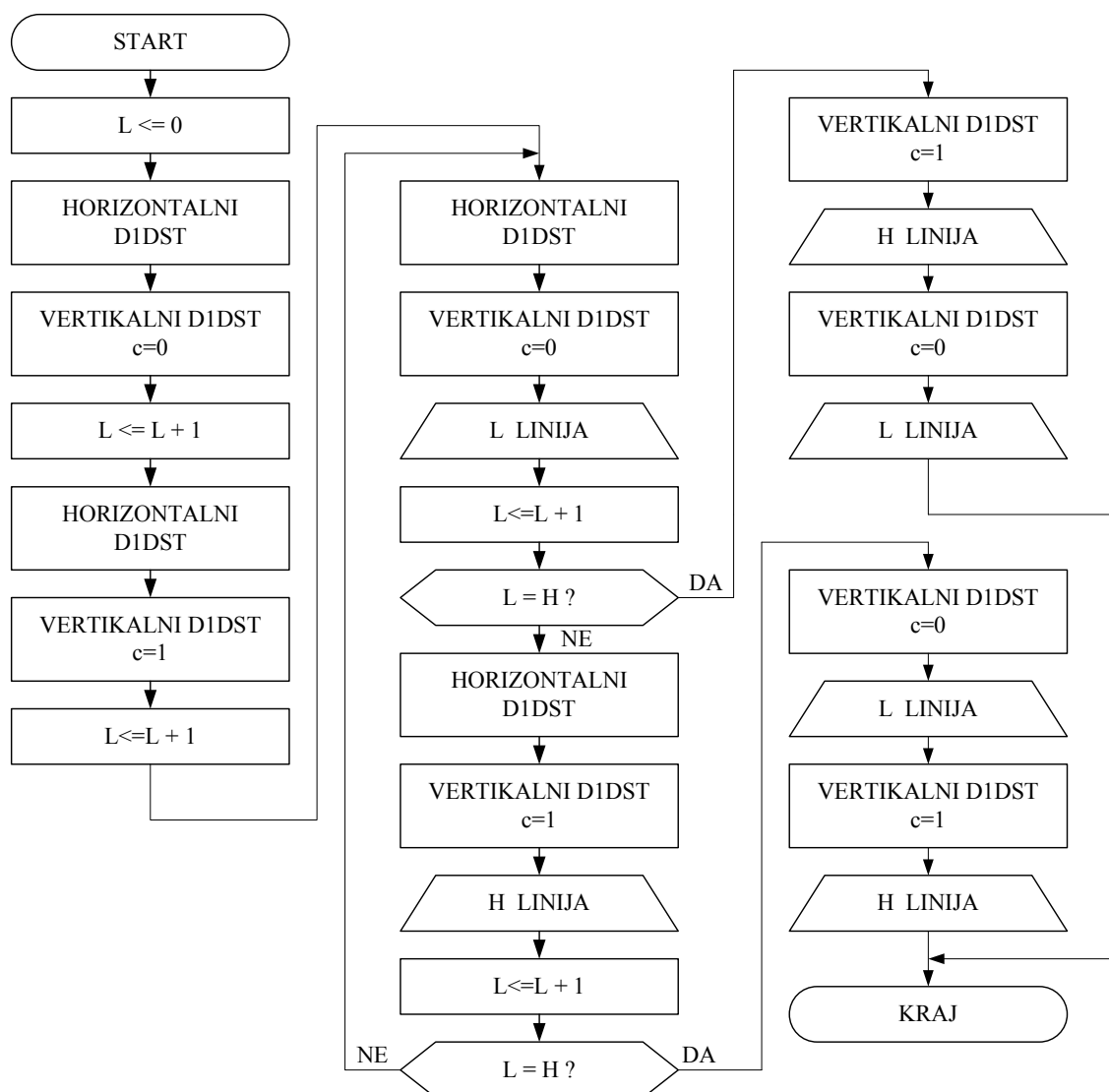
Većina postupaka za kompresiju koristi isti D1DST filter i za redove i za kolone, jer bi sve pravce u slici trebalo jednako tretirati. U načelu, kako horizontalni filter ima mnogo manji odnos memorije i kvaliteta, može se koristiti bolji i složeniji horizontalni filter, i nešto lošiji i manje složen vertikalni filter, praktično bez povećanja kapaciteta memorije.

**Tabela 2.8**  
Ukupni kapaciteti filterske i sinhronizacione memorije

Postupak	Ukupan kapacitet filterske memorije [interna promenljiva]	Ukupan kapacitet sinhronizacione memorije [koeficijent]
Konvolucija	$2 \cdot (2 \cdot D + 1) \cdot W \cdot \left(1 - \frac{1}{2^N}\right)$	$D \cdot W \cdot \left(2^N - 3 + \frac{1}{2^{N-1}}\right)$
Dizanje [45]	$2 \cdot (D + 2) \cdot W \cdot \left(1 - \frac{1}{2^N}\right)$	$D \cdot W \cdot \left(2^N - 3 + \frac{1}{2^{N-1}}\right)$
Nestacionarni filtri [50]	$2 \cdot D \cdot W \cdot \left(1 - \frac{1}{2^N}\right)$	0

**Slika 2.12** je dijagram toka jednog nivoa 2DST koja koristi nestacionarni vertikalni filter dužine  $2 \cdot D + 1 = 5$ , sa **Slike 2.6**, sa obradom linija po linija. Element za kašnjenje  $z^{-w}$  za vertikalni filter mora da čuva kompletnu horizontalnu liniju sa  $W$  piksela, tako da je  $w = W$ .

Horizontalna D1DST se obavlja naizmeničnom upotrebom  $c = 0$  za piksele sa parnim indeksima i  $c = 1$  za piksele sa neparnim indeksima. Nakon horizontalnog filtriranja vrši se vertikalna D1DST naizmeničnom upotrebom  $c = 0$  za linije sa parnim indeksima i  $c = 1$  za linije sa neparnim indeksima. Prva dva vertikalna filtriranja ne proizvode izlazne linije, usled inherentnog kašnjenja vertikalnog filtra. Počevši od treće linije, u petlji, svako vertikalno filtriranje proizvodi jednu izlaznu liniju, L (koja obuhvata LL i LH transformacione koeficijente) za  $c = 0$  ili H (koja obuhvata HL i HH transformacione koeficijente) za  $c = 1$ . Nakon što se učitaju i horizontalno i vertikalno filtriraju sve linije na ulazu, izvršavaju se još dva vertikalna filtriranja kako bi se ispraznila linija za kašnjenje u vertikalnom filtru. U zavisnosti od parnosti visine slike, poslednja generisana linija sadržiće visokopropusno ili niskopropusno transformisane koeficijente.



**Slika 2.12** Dijagram toka jednog nivoa D2DST sa linijskom obradom

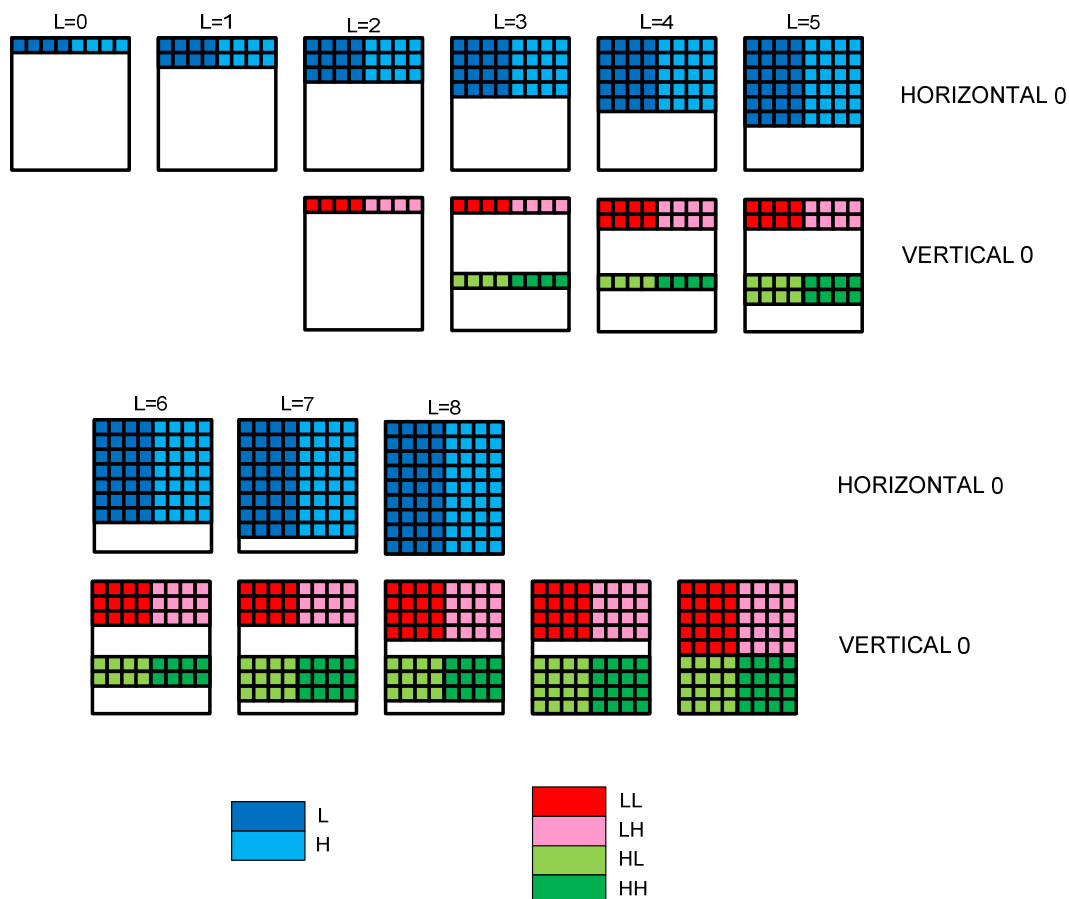
**Slika 2.13** ilustruje redosled generisanja horizontalno i vertikalno filtriranih linija za jednonivoosku D2DST za sliku dimenzija  $W = 8$ ,  $H = 9$ . Kako je prikazano, prva podopsežno transformisana linija generiše se nakon što su učitane i horizontalno filtrirane tri ulazne linije. Nakon što je učitana poslednja ulazna linija generišu se ukupno tri izlazne podopsežno transformisane linije.

Linijski metod obrade nije pogodan za rad u stvarnom vremenu, jer se vertikalno filtriranje izvršava tek kada je učitana i filtrirana cela horizontalna linija, što važi za sve nivoje podopsežne transformacije. Ovo se može uočiti na **Slici 2.14** na kojoj je prikazan



## Direktni dvodimenzionalni podopsežni transformator

početni deo dijagrama toka za višenivosku D2DST sa  $N = 2$  nivoa sa linijskom obradom, kao i na **Slici 2.15** koja ilustruje redosled generisanja horizontalno i vertikalno filtriranih linija za višenivoosku D2DST za sliku dimenzija  $W = 8$ ,  $H = 9$ , sa  $N = 2$  nivoa sa linijskom obradom.



**Slika 2.13** Redosled generisanja horizontalno i vertikalno filtriranih linija na nivou 0 D2DST sa linijskom obradom za sliku veličine  $8 \times 9$

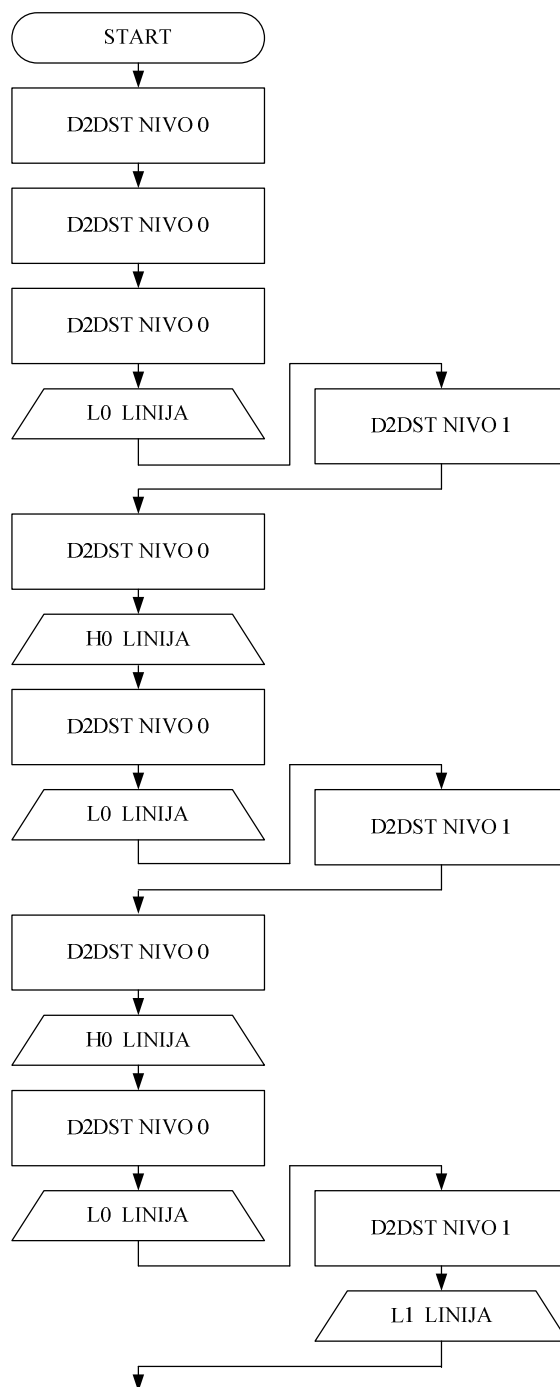
Kao što se vidi sa **Slikâ 2.12 i 2.13**, ukoliko se koriste po jedan horizontalni i vertikalni D1DST u ovoj metodi, potrebno je izvršiti D2DST na svim nivoima pre nego što se učitata sledeća ulazna linija. Ovo uvodi ograničenje na trajanje pauze između ulaznih linija na

$$\text{HorizontalBlanking} \geq W + \frac{W}{2} + \frac{W}{4} + \frac{W}{8} + \dots = W \sum_{i=0}^{N-1} \frac{1}{2^i} = W \left( 2 - \frac{1}{2^N} \right) \leq 2W .$$

## Direktni dvodimenzionalni podposežni transformator

Takođe, zbog pražnjenja linija za kašnjenje na svim nivoima potrebno je da pauza između dve susedne slike traje najmanje

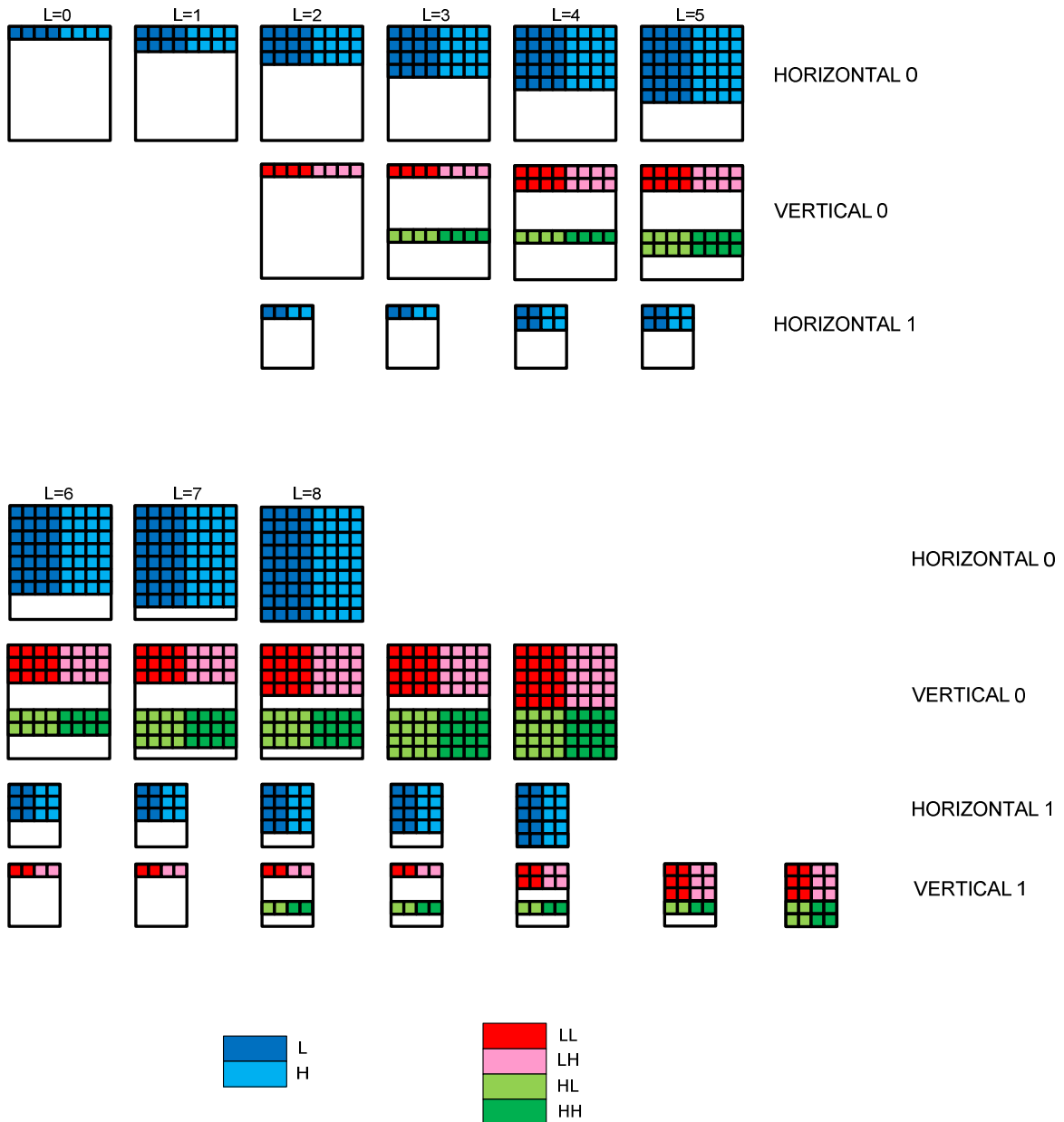
$$VerticalBlanking \geq 2 \left( W + \frac{W}{2} + \frac{W}{4} + \dots \right) = 2W \sum_{i=0}^{N-1} \frac{1}{2^i} = 2W \left( 2 - \frac{1}{2^N} \right) \leq 4W .$$



**Slika 2.14** Početni deo dijagrama toka višenivooskog D2DST sa linijskom obradom i  $N = 2$  nivoa

## Direktni dvodimenzionalni podopsežni transformator

U praksi je pauza između susednih slika sigurno veća od trajanja četiri ulazne linije. Međutim, pauza između susednih linija je sigurno kraća od trajanja jedne ulazne linije. Pored toga, u linijskoj obradi je potrebno čuvati cele linije LL podopsega u cilju horizontalnog filtriranja na sledećem nivou što povećava zahteve za memorijskim resursima.



**Slika 2.15** Redosled generisanja horizontalno i vertikalno filtriranih linija kod višenivooske D2DST sa linijskom obradom i  $N = 2$  nivoa za sliku veličine  $8 \times 9$

Ukoliko se u višenivooskoj D2DST koriste po dva horizontalna i vertikalna filtra, jedan od obe vrste za D2DST na nivou 0 i još po jedan od obe vrste za D2DST na svim ostalim nivoima (**Slika 2.16**), moguće je smanjiti minimalno trajanje pauze između dve susedne ulazne linije tako da

$$\text{HorizontalBlanking} \rightarrow 0$$

Međutim, preostaje problem povećanog iskorišćenja memorijskih resursa.

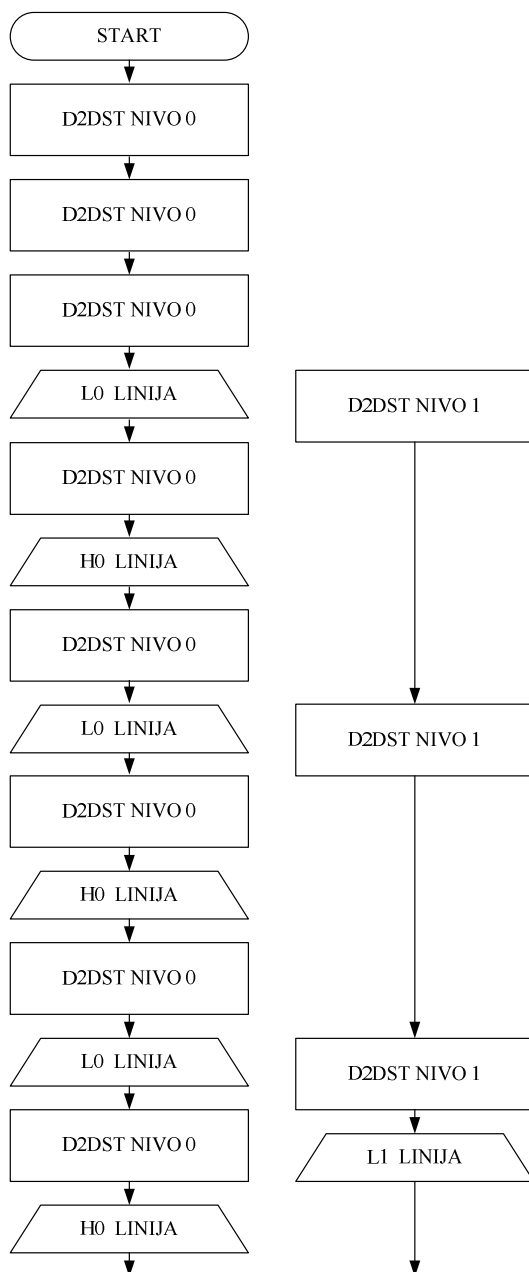
Ukoliko se iskoristi mogućnost da horizontalno filtriranje može da započne nakon što su dobavljena prva tri koeficijenta ulazne linije, moguće je smanjiti potrebu za memorijskim resursima. U tu svrhu, posmatrajmo **Slike 2.17 a)-d)** na kojima su ilustrovani trenuci generisanja koeficijenata na izlazima horizontalnog i vertikalnog filtra na različitim nivoima u toku nekoliko različitih ulaznih linija, a za širinu slike  $W = 14$  gde su korišćeni nestacionarni D1DST filtri [50]. Broj nivoa podopsežne dekompozicije je  $N = 2$ .

Tokom učitavanja trećeg piksela prve ulazne linije (**Slika 2.17 a)**), horizontalni filter generiše prvi niskopropusno filtrirani koeficijent u prvoj liniji horizontalno filtriranih koeficijenata na nivou 0. Tokom učitavanja četvrtog piksela prve ulazne linije, horizontalni filter generiše prvi visokopropusno filtrirani koeficijent u prvoj liniji horizontalno filtriranih koeficijenata na nivou 0. Tokom učitavanja petog piksela prve ulazne linije, horizontalni filter generiše drugi niskopropusno filtrirani koeficijent u prvoj liniji horizontalno filtriranih koeficijenata na nivou 0, itd.

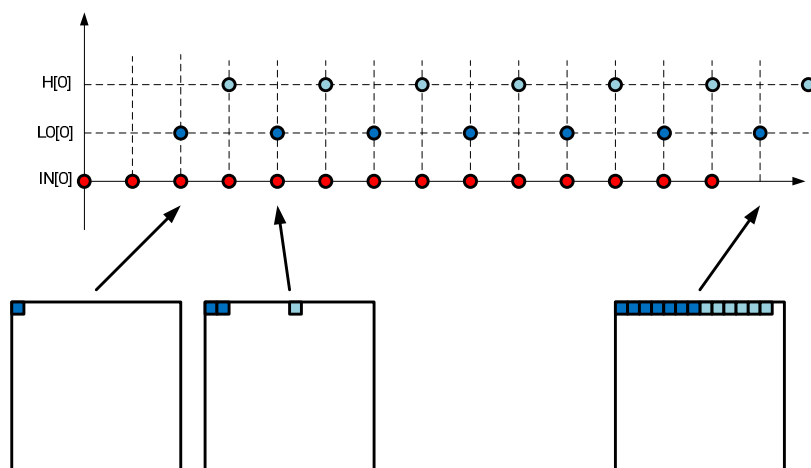
Tokom učitavanja trećeg piksela treće ulazne linije (**Slika 2.17 b)**), horizontalni filter generiše prvi niskopropusno filtrirani koeficijent u trećoj liniji horizontalno filtriranih koeficijenata na nivou 0. Vertikalni filter generiše prvi LL transformacioni koeficijent u prvoj liniji na nivou 0. Tokom učitavanja četvrtog piksela treće ulazne linije, horizontalni filter generiše prvi visokopropusno filtrirani koeficijent u trećoj liniji horizontalno filtriranih koeficijenata na nivou 0. Vertikalni filter generiše prvi LH transformacioni koeficijent u prvoj liniji na nivou 0. Tokom učitavanja sedmog piksela treće ulazne linije, horizontalni filter generiše treći niskopropusno filtrirani koeficijent u

## Direktni dvodimenzionalni podopsežni transformator

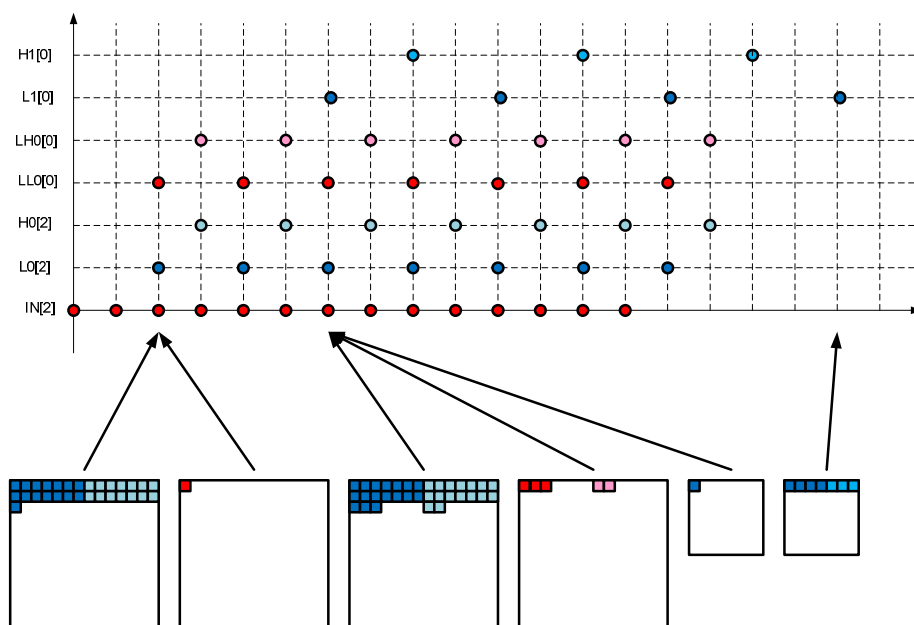
trećoj liniji horizontalno filtriranih koeficijenata na nivou 0. Vertikalni filtar generiše treći LL transformacioni koeficijent u prvoj liniji na nivou 0. Kako su na ulazu horizontalnog filtra dostupna tri LL transformaciona koeficijenta iz prve linije sa nivoua 0, horizontalni filtar generiše prvi niskopropusno transformisani koeficijent u prvoj liniji horizontalno filtriranih koeficijenata na nivou 1, itd.



**Slika 2.16** Početni deo dijagrama toka višenivooskog D2DST sa linijskom obradom i  $N = 2$  nivoua, korišćenjem jednog para filtara za nivou 0 i drugog para filtara za sve ostale nivoe



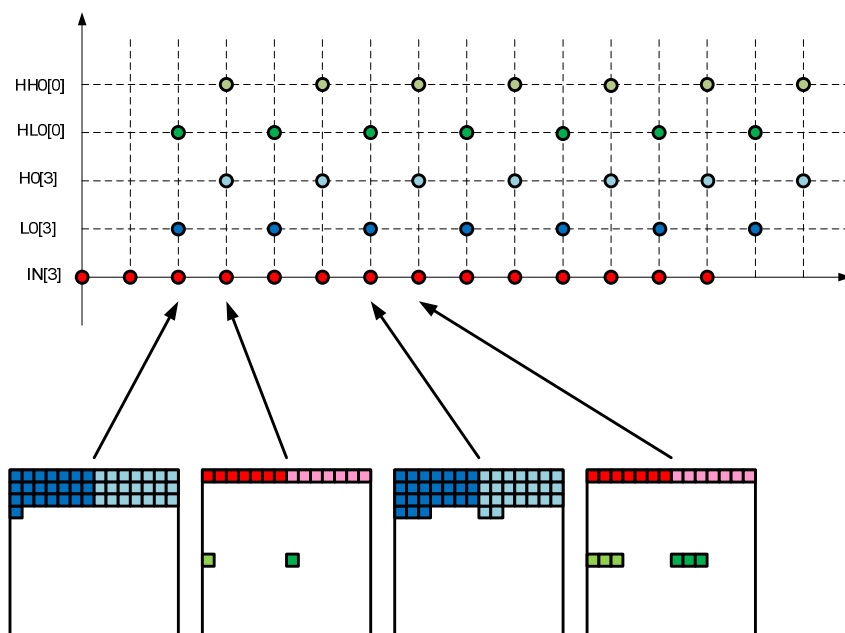
Slika 2.17 a) Filtriranje tokom učitavanja prve ulazne linije korišćenjem nestacionarnih DIDST filtara



Slika 2.17 b) Filtriranje tokom učitavanja treće ulazne linije korišćenjem nestacionarnih DIDST filtara

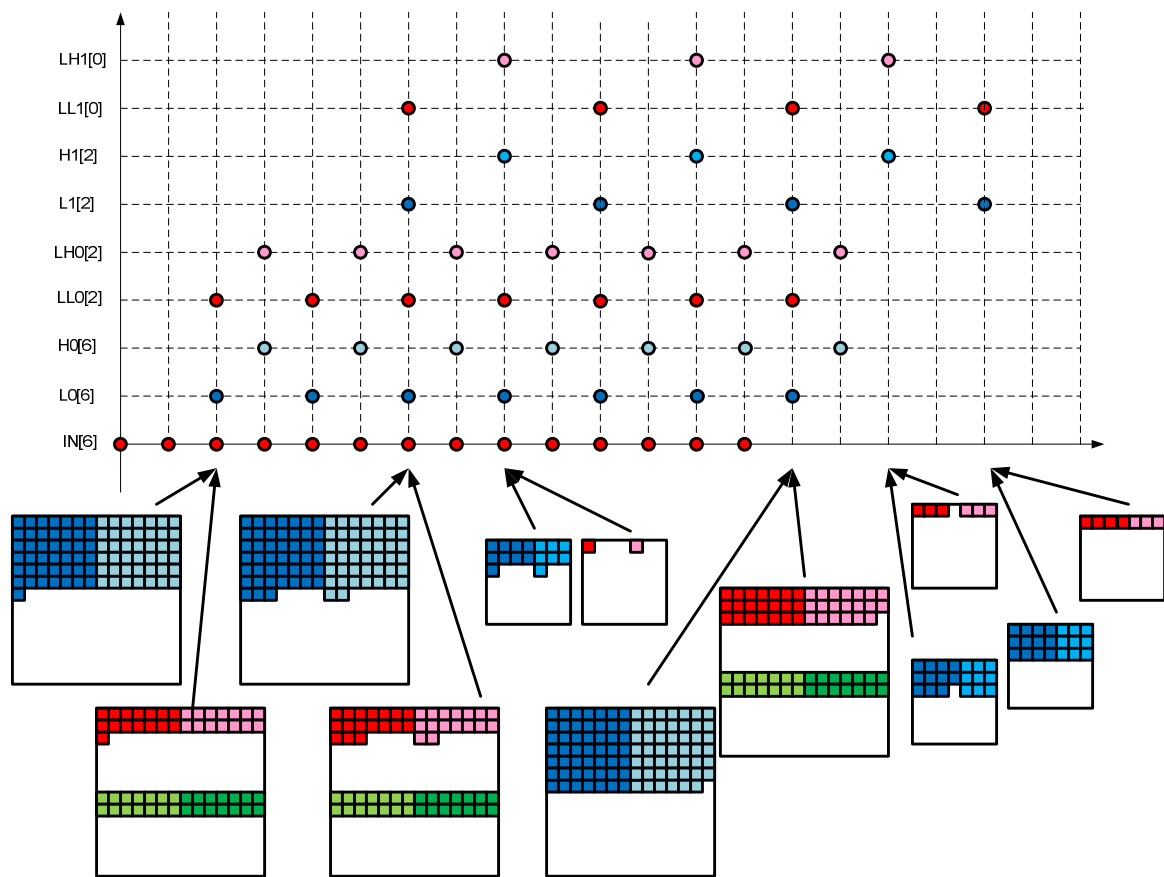
Tokom učitavanja trećeg piksela četvrte ulazne linije (Slika 2.17 c)), horizontalni filter generiše prvi niskopropusno filtrirani koeficijent u četvrtoj liniji horizontalno filtriranih koeficijenata na nivou 0. Vertikalni filter generiše prvi HL transformacioni koeficijent u prvoj liniji na nivou 0. Tokom učitavanja četvrtog piksela četvrte ulazne linije, horizontalni filter generiše prvi visokopropusno filtrirani koeficijent u četvrtoj liniji horizontalno filtriranih koeficijenata na nivou 0. Vertikalni filter generiše prvi HH transformacioni koeficijent na nivou 0. Tokom učitavanja petog piksela četvrte ulazne

linije, horizontalni filter generiše drugi niskopropusno filtrirani koeficijent u četvrtoj liniji horizontalno filtriranih koeficijenata na nivou 0. Vertikalni filter generiše drugi HL transformacioni koeficijent u prvoj liniji na nivou 0, itd.



**Slika 2.17 c)** Filtriranje tokom učitavanja četvrte ulazne linije korišćenjem nestacionarnih DIDST filtera

Tokom učitavanja trećeg piksela sedme ulazne linije (**Slika 2.17 d**), horizontalni filter generiše prvi niskopropusno filtrirani koeficijent u sedmoj liniji horizontalno filtriranih koeficijenata na nivou 0. Vertikalni filter generiše prvi LL transformacioni koeficijent u trećoj liniji na nivou 0. Tokom učitavanja četvrtog piksela sedme ulazne linije, horizontalni filter generiše prvi visokopropusno filtrirani koeficijent u sedmoj liniji horizontalno filtriranih koeficijenata na nivou 0. Vertikalni filter generiše prvi LH transformacioni koeficijent u trećoj liniji na nivou 0. Tokom učitavanja sedmog piksela sedme ulazne linije, horizontalni filter generiše treći niskopropusno filtrirani koeficijent u sedmoj liniji horizontalno filtriranih koeficijenata na nivou 0. Vertikalni filter generiše treći LL transformacioni koeficijent u drugoj liniji na nivou 0. Kako su na ulazu horizontalnog filtra dostupna tri LL transformaciona koeficijenta iz treće linije sa nivoa 0, horizontalni filter generiše prvi niskopropusno transformisani koeficijent u trećoj liniji horizontalno filtriranih koeficijenata na nivou 1. Vertikalni filter generiše prvi LL transformacioni koeficijent u prvoj liniji na nivou 1, itd.



**Slika 2.17 d)** Filtriranje tokom učitavanja sedme ulazne linije korišćenjem nestacionarnih DIDST filtara

Obradu opisanu na **Slikama 2.17 a)-d)** zvaćemo *obrada piksel po piksel*.

Relativni odnos trenutka generisanja transformisanih koeficijenta sa indeksom  $p$  u liniji sa indeksom  $l$  u odnosu na ulaznu sliku je

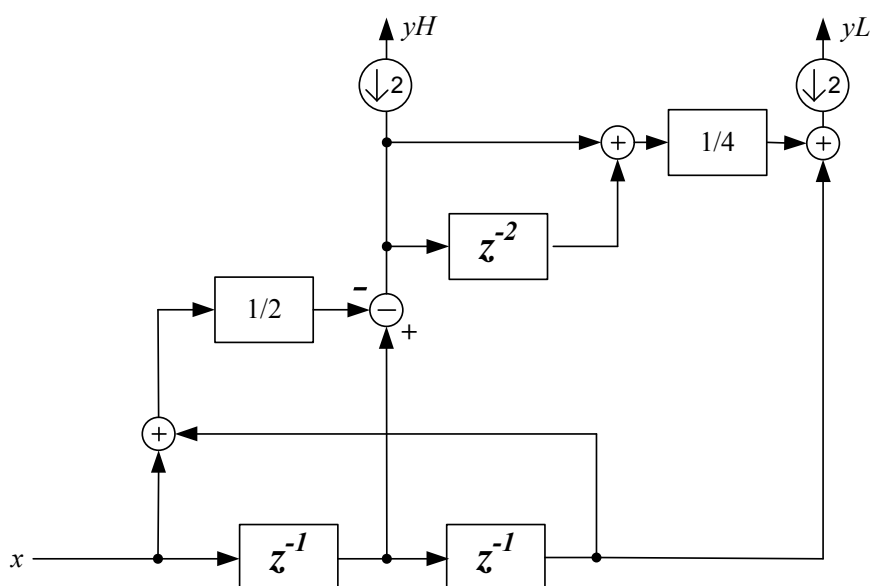
$$\begin{aligned}
 L_n[l, p] &\leq IN \left[ 2^n l + 2^{n+1} - 2, 2^{n+1} p + 2^{n+2} - 2 \right] \\
 H_n[l, p] &\leq IN \left[ 2^n l + 2^{n+1} - 2, 2^{n+1} p + 3 \cdot 2^n + 2^{n+1} - 2 \right] \\
 LL_n[l, p] &\leq IN \left[ 2^{n+1} l + 2^{n+2} - 2, 2^{n+1} p + 2^{n+2} - 2 \right] \\
 LH_n[l, p] &\leq IN \left[ 2^{n+1} l + 2^{n+2} - 2, 2^{n+1} p + 3 \cdot 2^n + 2^{n+1} - 2 \right] \\
 HL_n[l, p] &\leq IN \left[ 2^{n+1} l + 3 \cdot 2^n + 2^{n+1} - 2, 2^{n+1} p + 2^{n+2} - 2 \right] \\
 HH_n[l, p] &\leq IN \left[ 2^{n+1} l + 3 \cdot 2^n + 2^{n+1} - 2, 2^{n+1} p + 3 \cdot 2^n + 2^{n+1} - 2 \right].
 \end{aligned}$$



Na osnovu **Slika 2.17 a)-d)** uočavaju se dve stvari kod ovako koncipirane obrade piksel po piksel:

- na kraju ulazne linije postoje prazni vremenski slotovi tako da se generisanje završnih transformacionih koeficijenata može ubrzati, to jest izvršiti ranije
- postoje vremenski trenuci kada je potrebno istovremeno izvršiti dva horizontalna filtriranja ili dva vertikalna filtriranja, odnosno po jedan vertikalni i horizontalni filter nisu dovoljni.

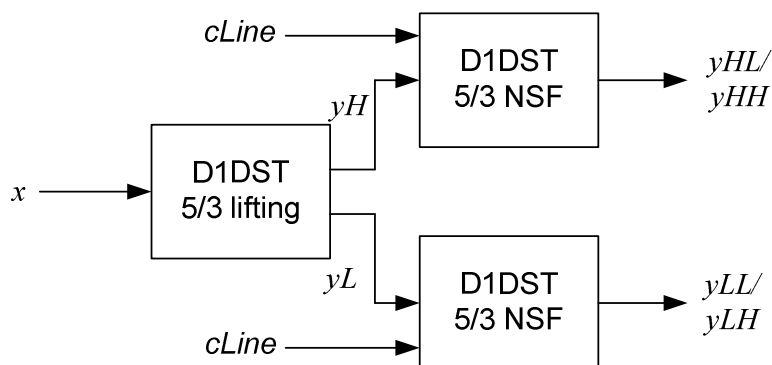
Rešenje potonjeg problema bilo bi korišćenje po dva horizontalna i vertikalna filtra, po jednog za D2DST na nivou 0, i još po jednog za D2DST za sve ostale nivoe. Ovakvo rešenje bi značajno usložnilo hardversku realizaciju. Umesto toga u ovom radu pristupilo se realizaciji obrade piksel po piksel pomoću jednog horizontalnog 5/3 filtra sa dizanjem [66] prikazanog na **Slici 2.18**, koji u svakom drugom taktu na svojim izlazima istovremeno proizvodi niskopropusno i visokopropusno D1DST transformisane koeficijente, i dva vertikalna nestacionarna 5/3 filtra sa **Slike 2.6**. Principijska blok šema realizacije filtriranja prikazana je na **Slici 2.19**.



**Slika 2.18** Realizacija 5/3 filtra sa dizanjem

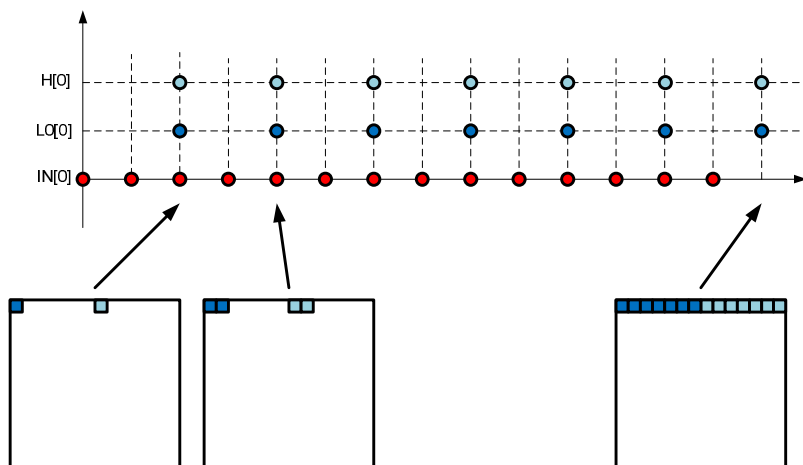
Na **Slikama 2.20 a)-d)** ilustrovani su trenuci generisanja koeficijenata na izlazima horizontalnog i vertikalnog filtra na različitim nivoima u toku nekoliko različitih ulaznih

linija, a za širinu slike  $W = 14$ , gde su korišćeni jedan horizontalni filter sa **Slike 2.18** i dva vertikalna filtra sa **Slike 2.6**, povezani na način koji je prikazan na **Slici 2.19**. Obrada je piksel po piksel, a broj nivoa  $N = 2$ .



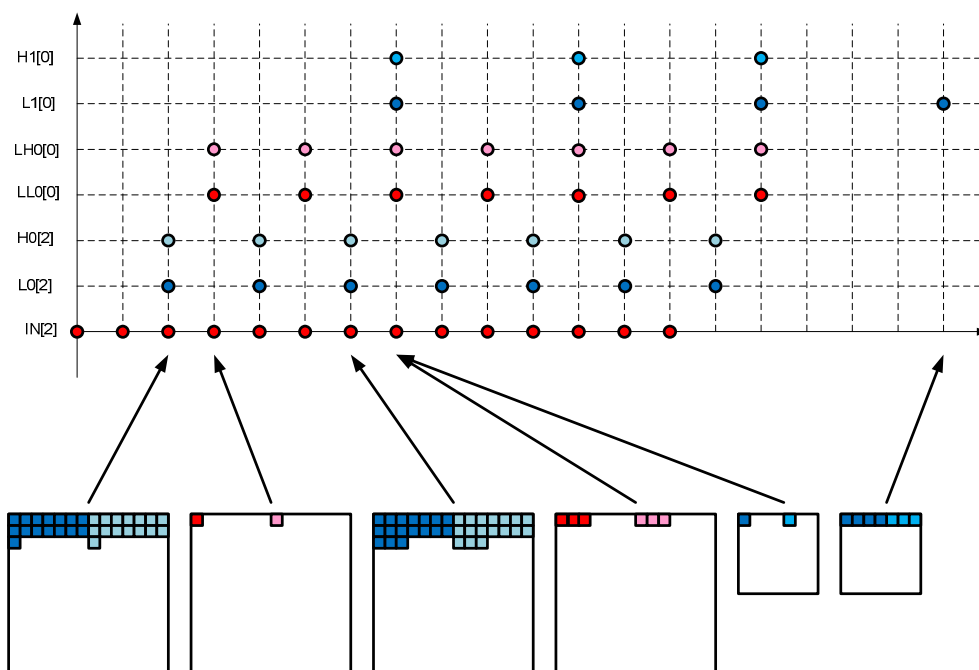
**Slika 2.19** Povezivanje horizontalnog filtra sa dizanjem i vertikalnih nestacionarnih filtara

Tokom učitavanja trećeg piksela prve ulazne linije (**Slika 2.20 a**)), horizontalni filter istovremeno generiše prvi niskopropusno i prvi visokopropusno filtrirani koeficijent u prvoj liniji horizontalno filtriranih koeficijenata na nivou 0. Tokom učitavanja petog piksela prve ulazne linije, horizontalni filter istovremeno generiše drugi niskopropusno i drugi visokopropusno filtrirani koeficijent u prvoj liniji horizontalno filtriranih koeficijenata na nivou 0, itd.



**Slika 2.20 a)** Filtriranje tokom učitavanja prve ulazne linije korišćenjem jednog horizontalnog D1DST filtra sa dizanjem i dva vertikalna D1DST nestacionarna filtra

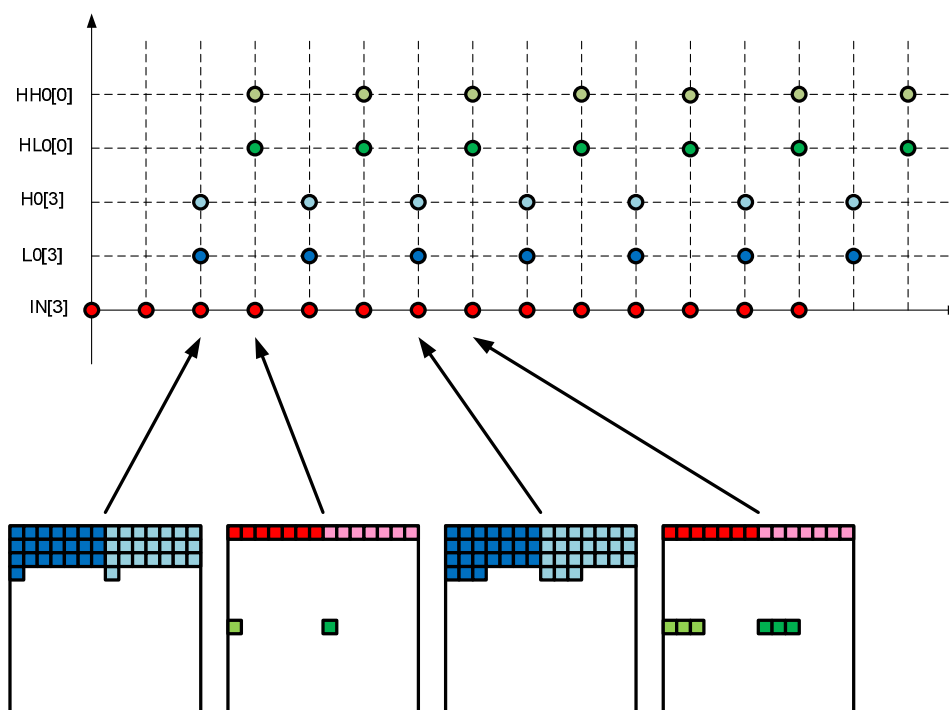
Tokom učitavanja trećeg piksela treće ulazne linije (**Slika 2.20 b**)), horizontalni filter istovremeno generiše prvi niskopropusno i prvi visokopropusno filtrirani koeficijent u trećoj liniji horizontalno filtriranih koeficijenata na nivou 0. U sledećem ciklusu prvi vertikalni filter generiše prvi LL transformacioni koeficijent u prvoj liniji na nivou 0, dok drugi vertikalni filter generiše prvi LH transformacioni koeficijent u prvoj liniji na nivou 0. Tokom učitavanja sedmog piksela treće ulazne linije, horizontalni filter istovremeno generiše treći niskopropusno i treći visokopropusno filtrirani koeficijent u trećoj liniji horizontalno filtriranih koeficijenata na nivou 0. U sledećem ciklusu prvi vertikalni filter generiše treći LL transformacioni koeficijent u prvoj liniji na nivou 0, dok drugi vertikalni filter generiše treći LH transformacioni koeficijent u prvoj liniji na nivou 0. Kako su na ulazu horizontalnog filtra dostupna tri LL transformaciona koeficijenta iz prve linije sa nivoa 0, horizontalni filter istovremeno generiše prvi niskopropusno i prvi visokopropusno transformisani koeficijent u prvoj liniji horizontalno filtriranih koeficijenata na nivou 1, itd.



**Slika 2.20 b)** Filtriranje tokom učitavanja treće ulazne linije korišćenjem jednog horizontalnog DIDST filtra sa dizanjem i dva vertikalna DIDST nestacionarna filtra

Tokom učitavanja trećeg piksela četvrte ulazne linije (**Slika 2.20 c**)), horizontalni filter istovremeno generiše prvi niskopropusno i prvi visokopropusno filtrirani koeficijent u

četvrtoj liniji horizontalno filtriranih koeficijenata na nivou 0. U sledećem ciklusu prvi vertikalni filter generiše prvi HL transformacioni koeficijent u prvoj liniji na nivou 0, dok drugi vertikalni filter generiše prvi HH transformacioni koeficijent u prvoj liniji na nivou 0. Tokom učitavanja petog piksela četvrte ulazne linije, horizontalni filter istovremeno generiše drugi niskopropusno i drugi visokopropusno filtrirani koeficijent u četvrtoj liniji horizontalno filtriranih koeficijenata na nivou 0. U sledećem ciklusu prvi vertikalni filter generiše drugi HL transformacioni koeficijent u prvoj liniji na nivou 0, dok drugi vertikalni filter generiše drugi HH transformacioni koeficijent u prvoj liniji na nivou 0, itd.

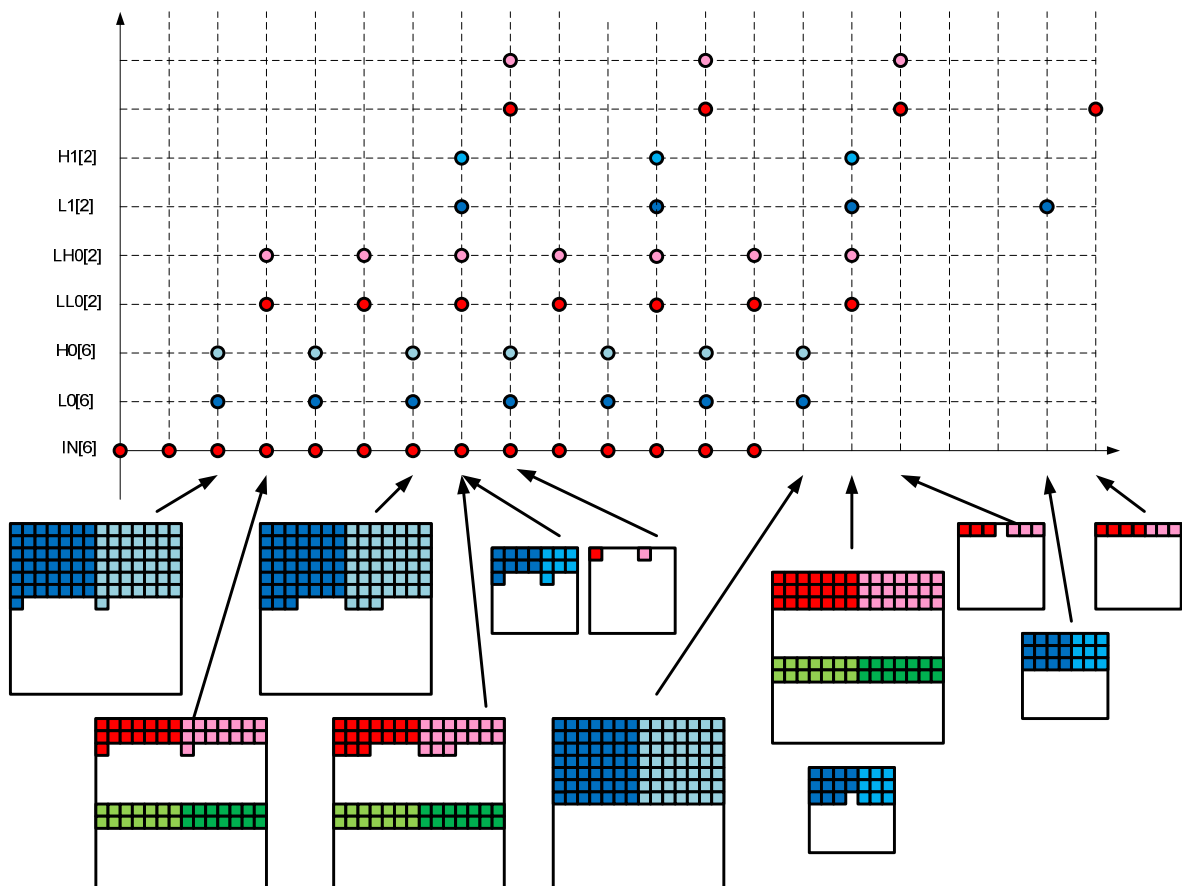


**Slika 2.20 c)** Filtriranje tokom učitavanja četvrte ulazne linije korišćenjem korišćenjem jednog horizontalnog DIDST filtra sa dizanjem i dva vertikalna DIDST nestacionarna filtra

Tokom učitavanja trećeg piksela sedme ulazne linije (**Slika 2.20 d**), horizontalni filter istovremeno generiše prvi niskopropusno i prvi visokopropusno filtrirani koeficijent u sedmoj liniji horizontalno filtriranih koeficijenata na nivou 0. U sledećem ciklusu prvi vertikalni filter generiše prvi LL transformacioni koeficijent u trećoj liniji na nivou 0, dok drugi vertikalni filter generiše prvi LH transformacioni koeficijent u trećoj liniji na nivou 0. Tokom učitavanja sedmog piksela sedme ulazne linije, horizontalni filter istovremeno generiše treći niskopropusno i treći visokopropusno filtrirani koeficijent u

## Direktni dvodimenzionalni podopsežni transformator

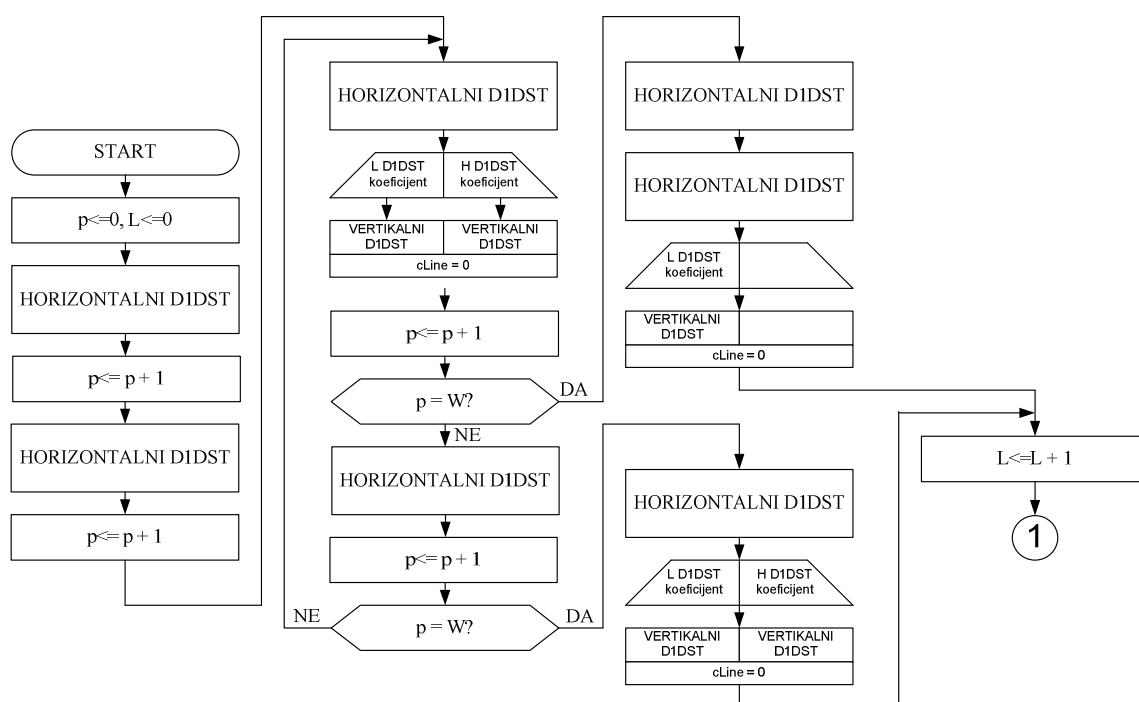
sedmoj liniji horizontalno filtriranih koeficijenata na nivou 0. U sledećem ciklusu prvi vertikalni filter generiše treći LL transformacioni koeficijent u trećoj liniji na nivou 0, dok drugi vertikalni filter generiše treći LH transformacioni koeficijent u trećoj liniji na nivou 0. Kako su na ulazu horizontalnog filtra dostupna tri LL transformaciona koeficijenta iz treće linije sa nivoua 0, horizontalni filter istovremeno generiše prvi niskopropusno i prvi visokopropusno transformisani koeficijent u trećoj liniji horizontalno filtriranih koeficijenata na nivou 1. U sledećem ciklusu prvi vertikalni filter generiše prvi LL transformacioni koeficijent u prvoj liniji na nivou 1, dok drugi vertikalni filter generiše prvi LH transformacioni koeficijent u prvoj liniji na nivou 1.



**Slika 2.20 d)** Filtriranje tokom učitavanja sedme ulazne linije korišćenjem jednog horizontalnog DIDST filtra sa dizanjem i dva vertikalna DIDST nestacionarna filtra

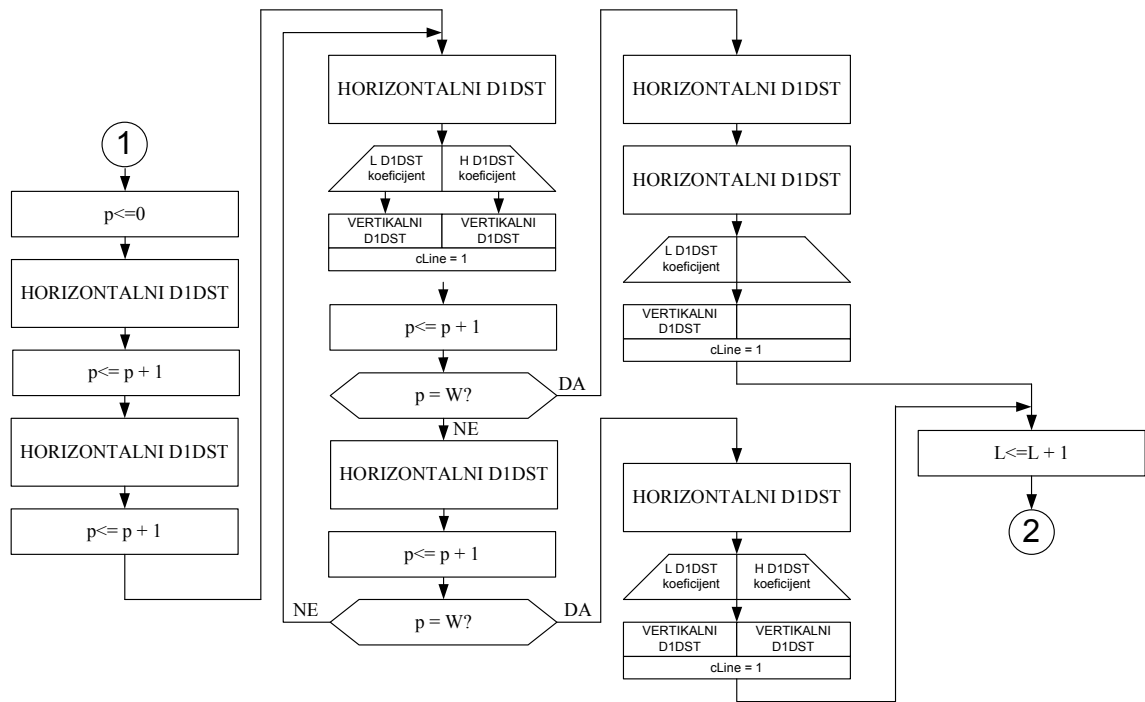
Generalizovani dijagram toka jednonivooske D2DST za vreme obrade jedne ulazne slike, za neparne i parne kombinacije dimenzija slike (širina  $W$  i visina  $H$ ) prikazan je na **Slikama 2.21 a)-f)**. Niskopropusno i visokopropusno filtrirani koeficijenti generišu se za vreme trajanja svake linije ulazne slike počevši od trećeg ulaznog piksela (**Slike 2.21**

a)-d)). LL i LH transformacioni koeficijenti generišu se počevši od trećeg piksela u svakoj neparnoj liniji počevši od treće linije ulazne slike (Slika 2.21 c)). HL i HH transformacioni koeficijenti generišu se počevši od trećeg piksela u svakoj parnoj liniji počevši od četvrte linije ulazne slike (Slika 2.21 d)). Na kraju slike je potrebno isprazniti linije za kašnjenje u vertikalnim filtrima, i na taj način generisati poslednje dve linije D2DST transformacionih koeficijenata, procedura koja se malo razlikuje u zavisnosti da li je visina slike neparan (Slika 2.21 e)) ili paran broj (Slika 2.21 f)).

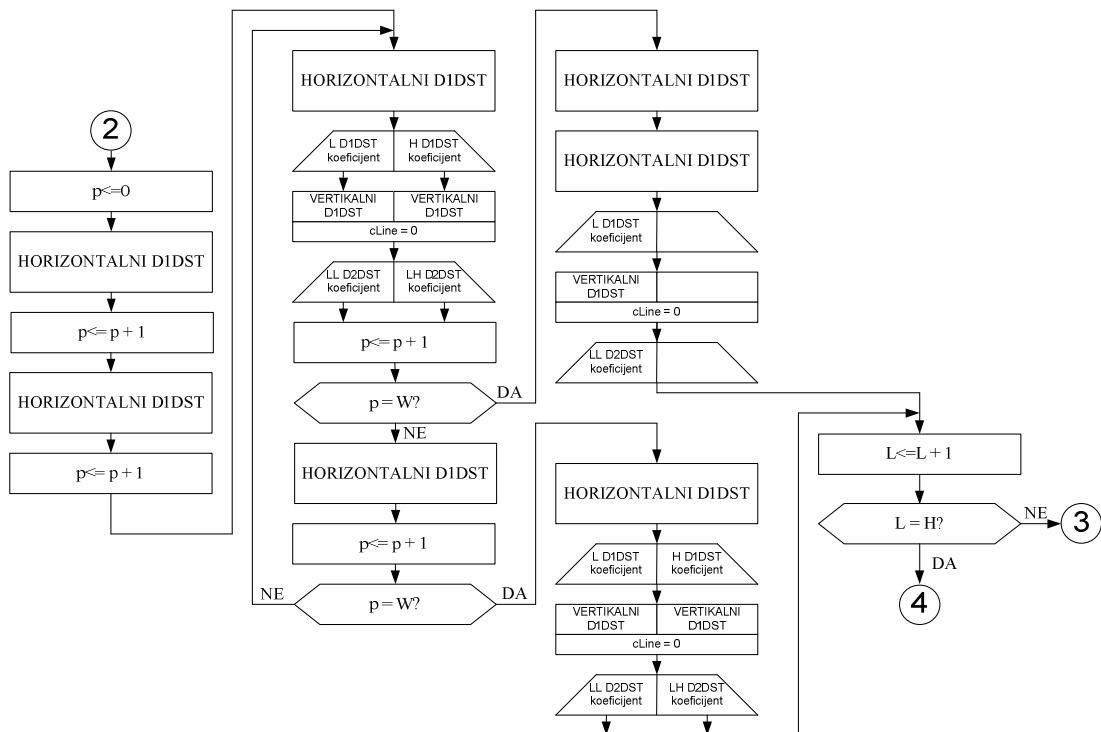


Slika 2.21 a) Dijagram toka jednonivooske D2DST sa obradom piksel po piksel, korišćenjem jednog horizontalnog DIDST filtra sa dizanjem i dva vertikalna DIDST nestacionarna filtra, za vreme učitavanja prve linije ulazne slike

# Direktni dvodimenzionalni podopsežni transformator

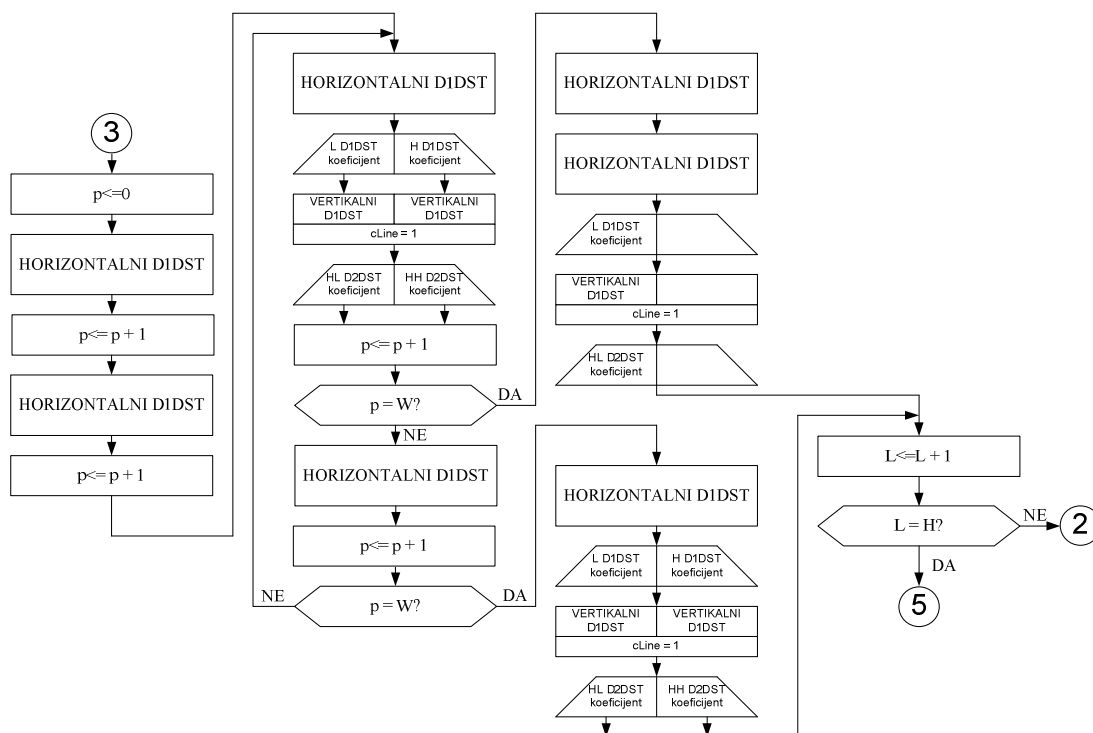


**Slika 2.21 b)** Dijagram toka jednonivooske D2DST sa obradom piksel po piksel, korišćenjem jednog horizontalnog D1DST filtra sa dizanjem i dva vertikalna D1DST nestacionarna filtra, za vreme učitavanja druge linije ulazne slike

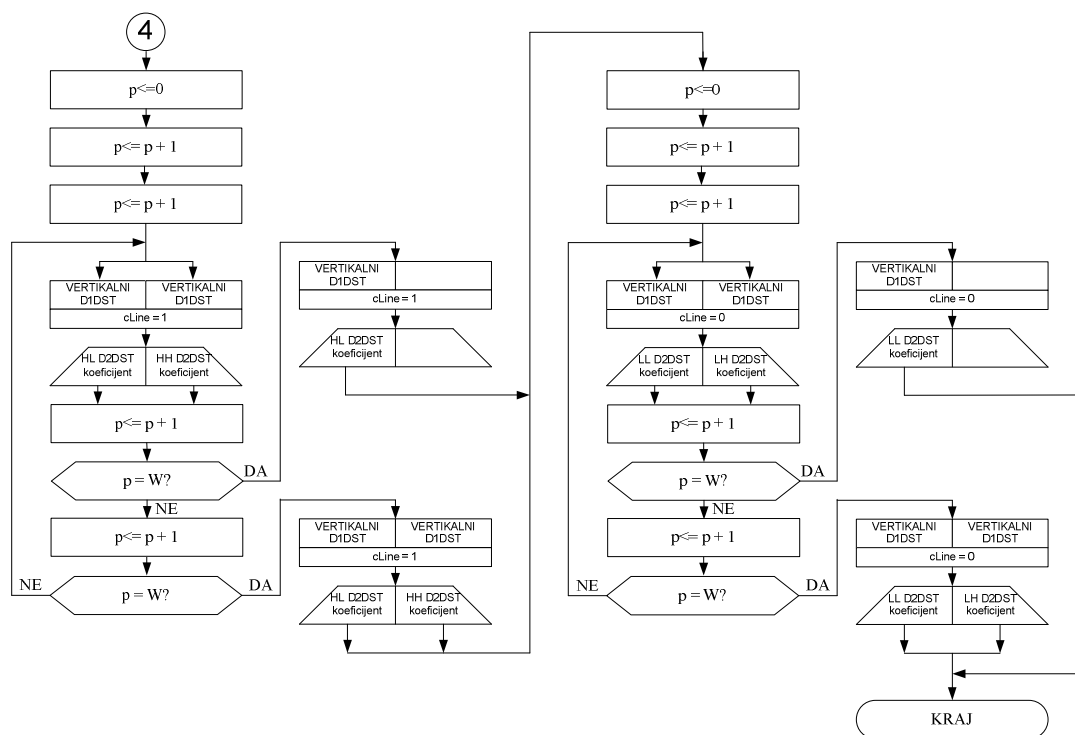


**Slika 2.21 c)** Dijagram toka jednonivooske D2DST sa obradom piksel po piksel, korišćenjem jednog horizontalnog D1DST filtra sa dizanjem i dva vertikalna D1DST nestacionarna filtra, za vreme učitavanja neparne linije ulazne slike, počevši od treće linije

# Direktni dvodimenzionalni podposežni transformator

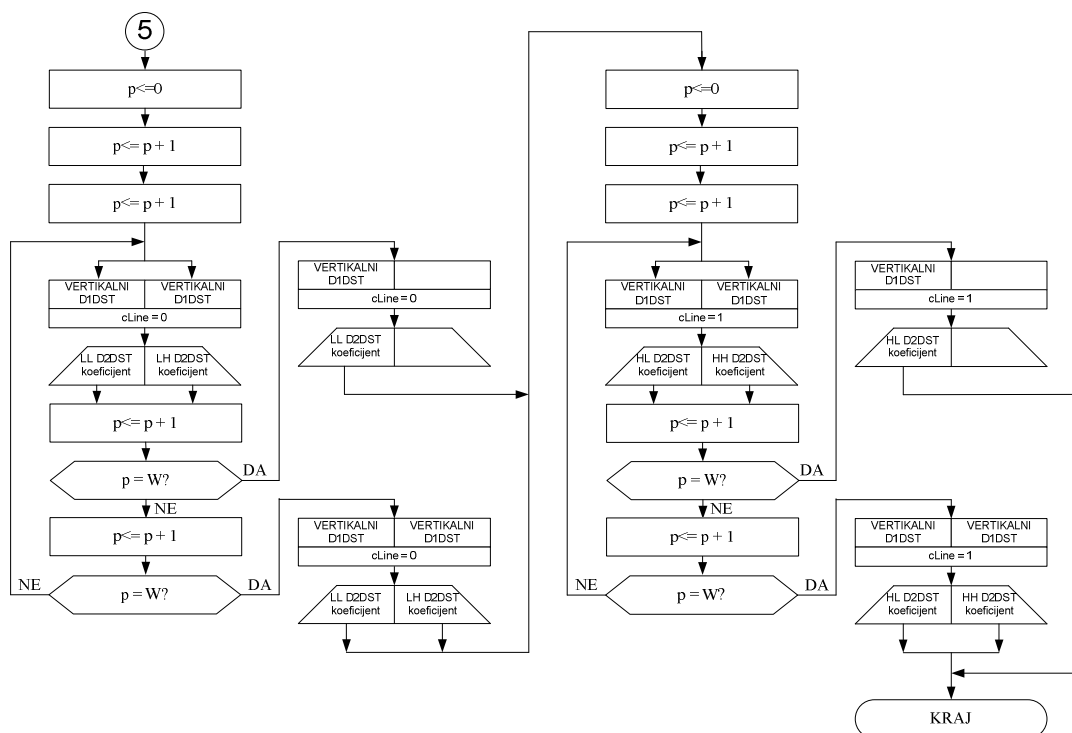


Slika 2.21 d) Dijagram toka jednonivooske D2DST sa obradom piksel po piksel, korišćenjem jednog horizontalnog D1DST filtra sa dizanjem i dva vertikalna D1DST nestacionarna filtra, za vreme učitavanja parne linije ulazne slike, počevši od četvrte linije



Slika 2.21 e) Dijagram toka jednonivooske D2DST sa obradom piksel po piksel, korišćenjem jednog horizontalnog D1DST filtra sa dizanjem i dva vertikalna D1DST nestacionarna filtra, pražnjenje linija za kašnjenje u vertikalnim filtrima, neparna visina slike





**Slika 2.21 f)** Dijagram toka jednonivooske D2DST sa obradom piksel po piksel, korišćenjem jednog horizontalnog DIDST filtra sa dizanjem i dva vertikalna DIDST nestacionarna filtra, pražnjenje linija za kašnjenje u vertikalnim filtrima, parna visina slike

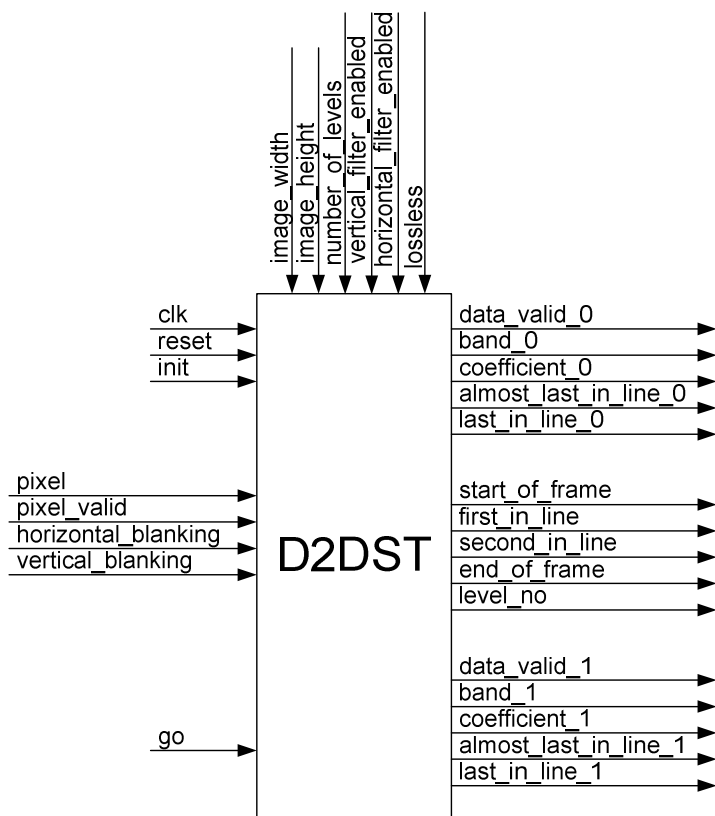
## Hardverska realizacija direktnog dvodimenzionalnog podopsežnog transformatora

Najviši hijerarhijski nivo hardverske realizacije dvodimenzionalnog podopsežnog transformatora za jednu od YUV ravni boje prikazan je na **Slici 2.22**.

Značenja signala sa **Slike 2.22** data su u **Tabeli 2.9**. U nastavku će biti detaljnije objašnjena funkcionalnost pojedinih signala.

Ulazni signal *number\_of\_levels* signalizira koliko nivoa podopsežne transformacije je potrebno izvršiti. Arhitektura podržava maksimalno 10 nivoa dekompozicije. Ograničenje je postavljeno jer bi sa većim brojem nivoa dekompozicije došlo do prekoračenja u 32-bitnoj aritmetici koja se koristi.

## Direktni dvodimenzionalni podopsežni transformator



**Slika 2.22** Najviši hijerarhijski nivo hardverske realizacije dvodimenzionalnog podopsežnog transformatora

**Tabela 2.9**  
Signali na najvišem hijerarhijskom nivou D2DST

Ime signala	Širina signala	Smer signala	Funkcija signala
pixel	8 bita	ulaz	piksel ulazne slike
pixel_valid	1 bit	ulaz	kvalifikator za piksel ulazne slike
horizontal_blanking	1 bit	ulaz	aktivan za vreme pauze između dve susedne linije
vertical_blanking	1 bit	ulaz	aktivan za vreme pauze između dve susedne ulazne slike
go	1 bit	ulaz	Ukoliko je aktivan na početku slike izvršava se D2DST, ukoliko nije D2DST se ne izvršava
image_width	16 bita	ulaz	širina ulazne slike
image_height	16 bita	ulaz	visina ulazne slike
number_of_levels	4 bita	ulaz	definiše ukupan broj nivoa dekompozicije
vertical_filter_enabled	10 bita	ulaz	signalizira da li ima potrebe da se vrši vertikalno filtriranje na datom nivou
horizontal_filter_enabled	10 bita	ulaz	signalizira da li ima potrebe da se vrši horizontalno filtriranje na datom nivou

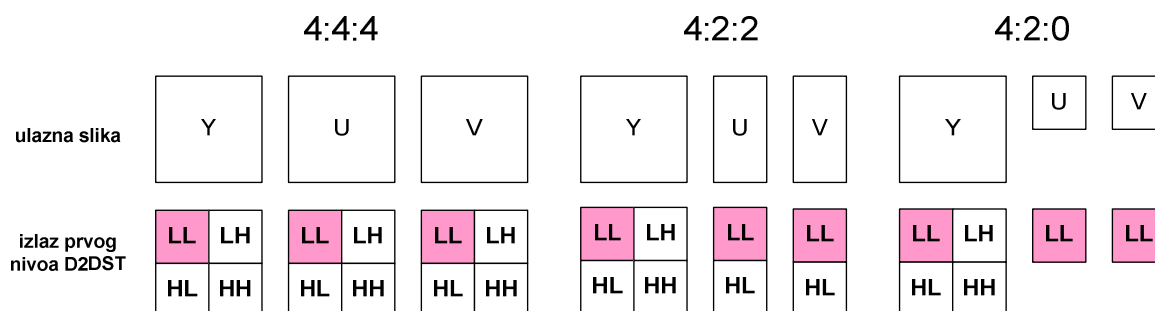
## Direktni dvodimenzionalni podopsežni transformator

lossless	1 bit	ulaz	signalizira da li je kompresija bez ili sa gubicima
coefficient_0	32 bita	izlaz	LL ili HL transformacioni koeficijent, izlaz prvog vertikalnog filtra
coefficient_1	32 bita	izlaz	LH ili HH transformacioni koeficijent, izlaz drugog vertikalnog filtra
level_no	4 bita	izlaz	označava redni broj nivoa dekompozicije kome pripadaju coefficient_0 i/ili coefficient_1
data_valid_0	1 bit	izlaz	kvalifikator za izlaz coefficient_0
data_valid_1	1 bit	izlaz	kvalifikator za izlaz coefficient_1
band_0	2 bita	izlaz	podopseg kome pripada coefficient_0
band_1	2 bita	izlaz	podopseg kome pripada coefficient_1
start_of_frame	1 bit	izlaz	ako je aktivan, signalizira da su coefficient_0 i/ili coefficient_1 prvi koeficijenti na datom nivou dekompozicije za datu ulaznu sliku
end_of_frame	1 bit	izlaz	ako je aktivan, signalizira da su coefficient_0 i/ili coefficient_1 poslednji za datu ulaznu sliku
first_in_line	1 bit	izlaz	signalizira da su coefficient_0 i/ili coefficient_1 prvi u tekućoj liniji transformacionih koeficijenata
second_in_line	1 bit	izlaz	signalizira da su coefficient_0 i/ili coefficient_1 drugi u tekućoj liniji transformacionih koeficijenata
almost_last_in_line_0	1 bit	izlaz	signalizira da je coefficient_0 preposlednji u tekućoj liniji transformacionih koeficijenata
almost_last_in_line_1	1 bit	izlaz	signalizira da je coefficient_1 preposlednji u tekućoj liniji transformacionih koeficijenata
last_in_line_0	1 bit	izlaz	signalizira da je coefficient_0 poslednji u tekućoj liniji transformacionih koeficijenata
last_in_line_1	1 bit	izlaz	signalizira da je coefficient_1 poslednji u tekućoj liniji transformacionih koeficijenata

Ulazni signali *horizontal\_filter\_enabled* i *vertical\_filter\_enabled* koriste se zbog različitih scenarija D2DST u zavisnosti od formata ulazne YUV slike. Ukoliko je ulazna YUV slika u formatu 4:4:4, sve tri ravni boje imaju iste dimenzije i biće na svim nivoima uključeno i horizontalno i vertikalno filtriranje. Ukoliko je ulazna slika u formatu 4:2:2, U i V ravni boje imaju duplo manju širinu od Y ravni boje, tako da nema

## Direktni dvodimenzionalni podopsežni transformator

potrebe da se izvrši horizontalno filtriranje na prvom nivou dekompozicije – U i V komponente slike se tretiraju kao da su već niskopropusno horizontalno filtrirane. Ukoliko je ulazna slika u formatu 4:2:0, U i V ravni boje imaju duplo manje i širinu i visinu od Y ravni boje, tako da nema potrebe da se izvrše ni horizontalno ni vertikalno filtriranje na prvom nivou dekompozicije – U i V komponente slike se tretiraju kao da su već prošle kroz jedan nivo podopsežne transformacije (**Slika 2.23**).



**Slika 2.23** Razlika između obrada na prvom nivou dekompozicije na Y, i U i V ravnima boje za tri najčešća formata ulazne slike

Ulazni signal **lossless** se u D2DST koristi kao signalizacija da li je potrebno pre prvog nivoa dekompozicije izvršiti konverziju ulaznog piksela iz 8-bitnog neoznačenog formata u 18.14 format sa nepokretnom tačkom.

Blok šema realizacije dvodimenzionalnog podopsežnog transformatora sa **Slike 2.22** prikazana je na **Slici 2.24**.

Blok D2DST FILTER sa **Slike 2.24** predstavlja horizontalni i oba vertikalna filtra, kako je prikazano na **Slici 2.25**.

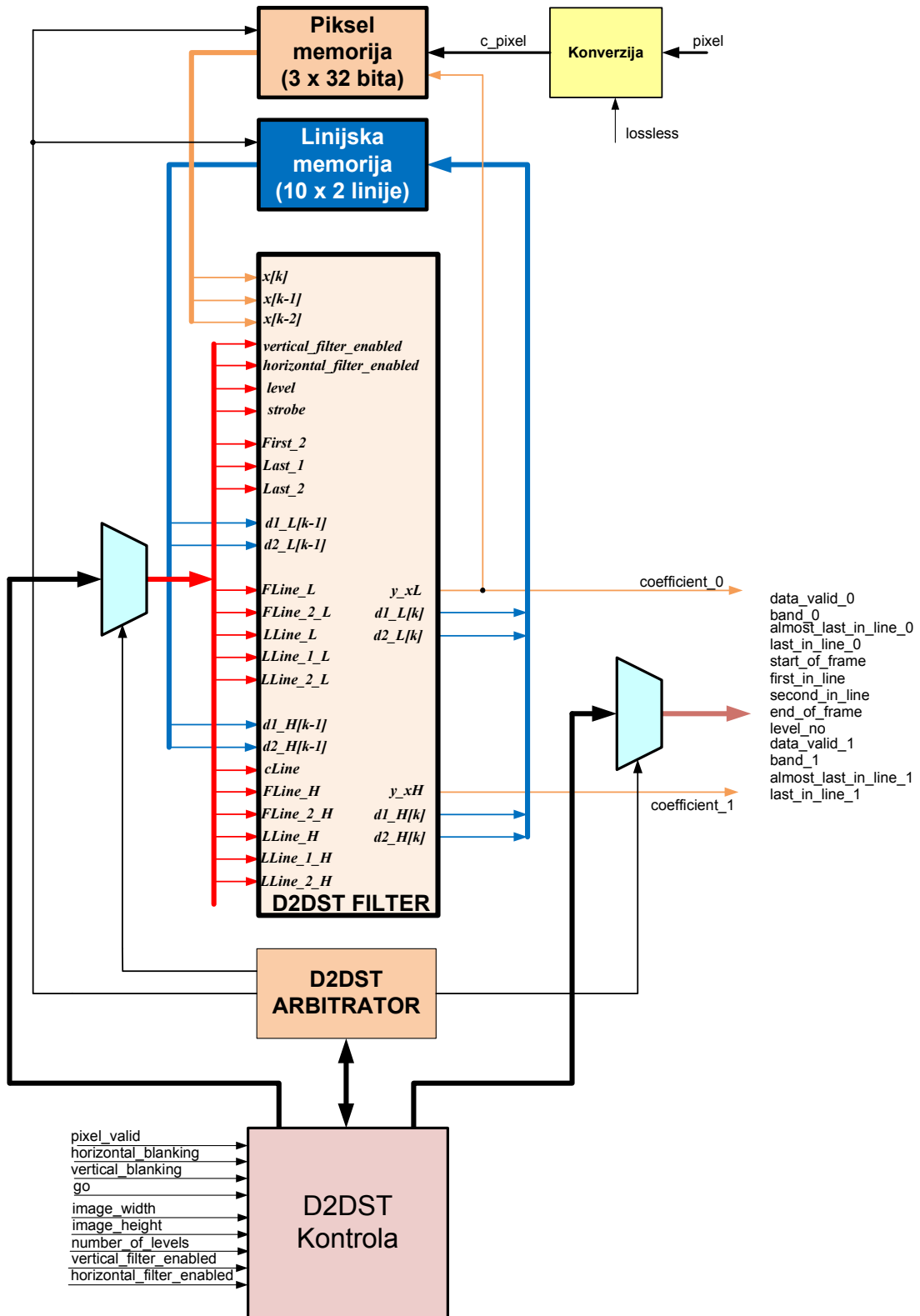
Lifting memorija kapaciteta 10x32 bita sa **Slike 2.25** čuva prethodnu vrednost visokopropusno filtriranog D1DST transformisanog koeficijenta za svaki nivo. Ovaj blok odgovara elementu za kašnjenje  $z^{-2}$  sa **Slike 2.18**. Realizacija horizontalnog D1DST filtra prikazana je na **Slici 2.26**, dok je realizacija jednog od vertikalnih D1DST filtara prikazana na **Slici 2.27**.

Koeficijent memorija sa **Slike 2.24** obezbeđuje liniju za kašnjenje  $x[k]$ ,  $x[k-1]$ ,  $x[k-2]$ , odnosno ulazne podatke za horizontalni filter. Mada se čini da je dovoljno da ova linija za kašnjenje ima svega dva elementa, kako su filteri deljeni resurs trećim elementom linije za kašnjenje se obezbeđuje čuvanje ulaznog podatka horizontalnog filtera za neki od dubljih nivoa dekompozicije do momenta kada je filter slobodan da izvrši filtriranje na datom nivou. Više o ovome biće rečeno u delu u kome se opisuje D2DST ARBITRATOR blok. Jedino za prvi nivo dekompozicije nije neophodno da ova linija ima tri registra. Realizacija Koeficijent memorije prikazana je na **Slici 2.28**.

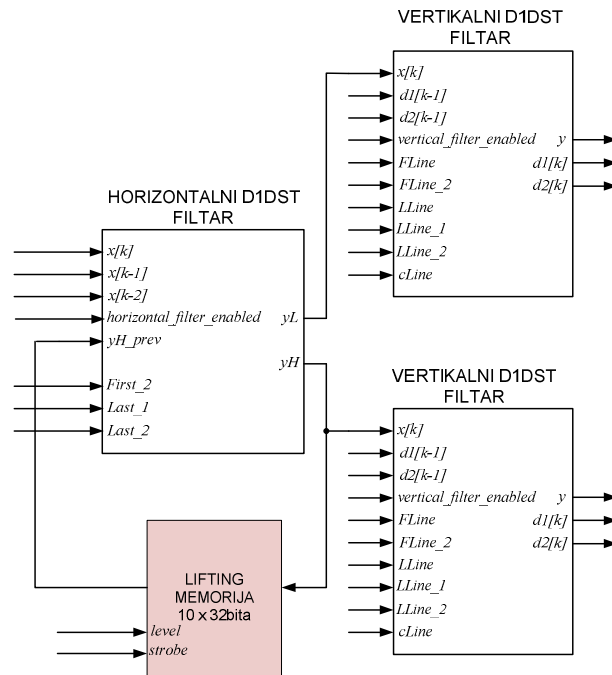
Linajska memorija sa **Slike 2.24** obezbeđuje po dve linije za kašnjenje za dva vertikalna filtera. Ovaj blok odgovara elementima za kašnjenje  $z^{-W}$  sa **Slike 2.8**. Kako dva vertikalna filtera generišu koeficijente ovih linija po tačno određenom redosledu, moguće je koristiti FIFO memorije. Pored toga, za dva vertikalna filtera u jednom trenutku potrebna su ukupno četiri koeficijenta, tako da je širina reči u FIFO memoriji  $4 \times 32 = 128$  bita. Realizacija Linajske memorije prikazana je na **Slici 2.29**. U konkretnom slučaju hardver je dimenzionisan za maksimalnu širinu ulazne slike od  $W = 2048$  piksela.

Blok D2DST KONTROLA sa **Slike 2.24** predstavlja mašine stanja za ukupno 10 nivoa podopsežne transformacija, prikazane na **Slici 2.30**. Svaka od ovih mašina stanja implementira dijagram toka prikazan na **Slikama 2.21 a)-f)**.

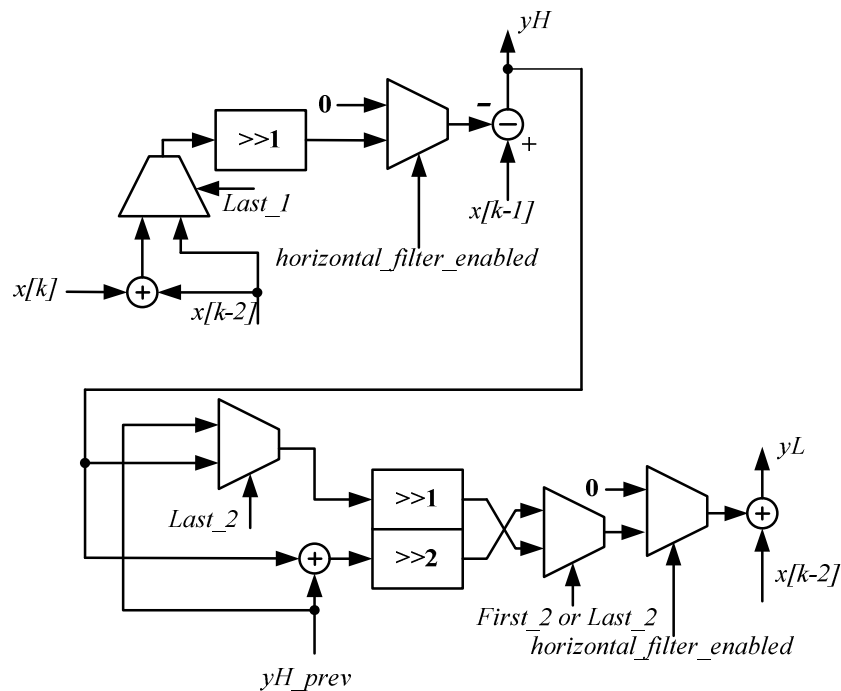
# Direktni dvodimenzionalni podposežni transformator



Slika 2.24 Blok šema dvodimenzionalnog podposežnog transformatora

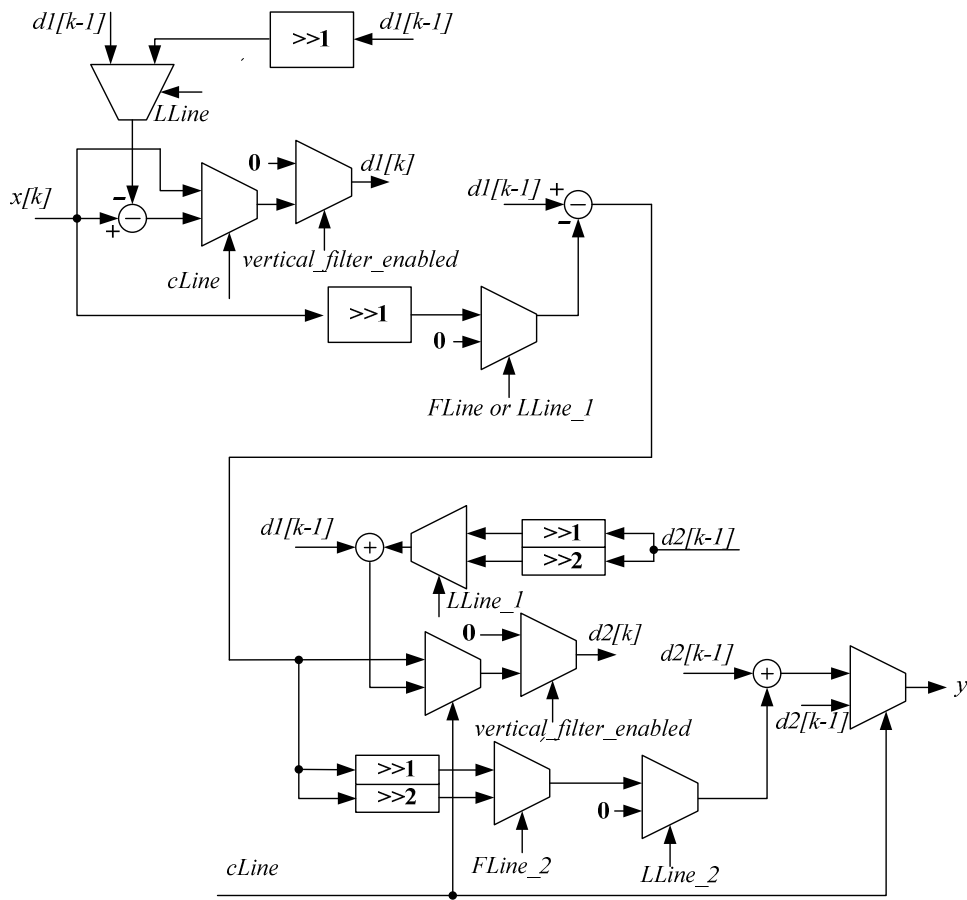


Slika 2.25 Interna blok šema bloka D2DST FILTER

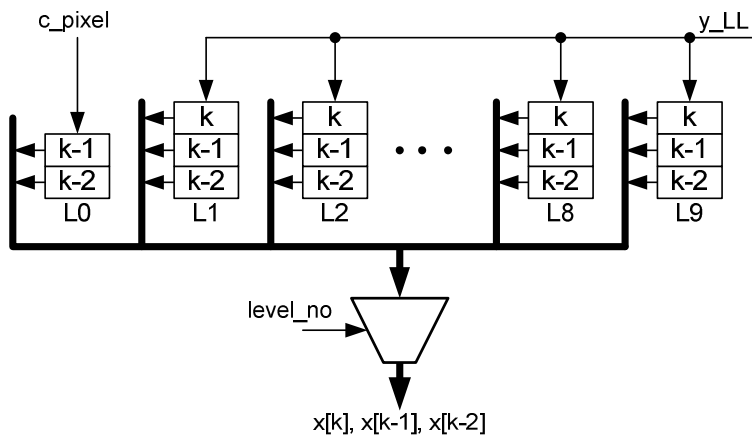


Slika 2.26 Realizacija horizontalnog filtra sa dizanjem

# Direktni dvodimenzionalni podposežni transformator



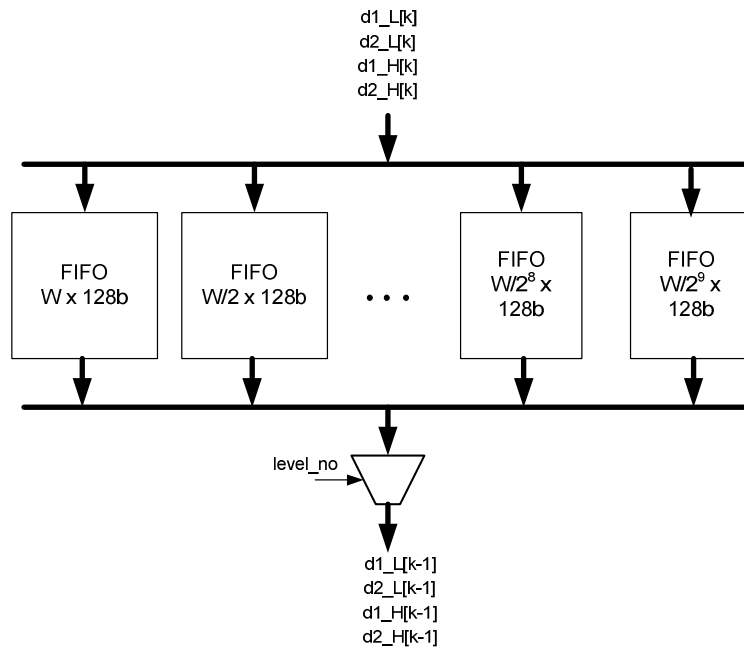
Slika 2.27 Realizacija vertikalnih nestacionarnih filtara



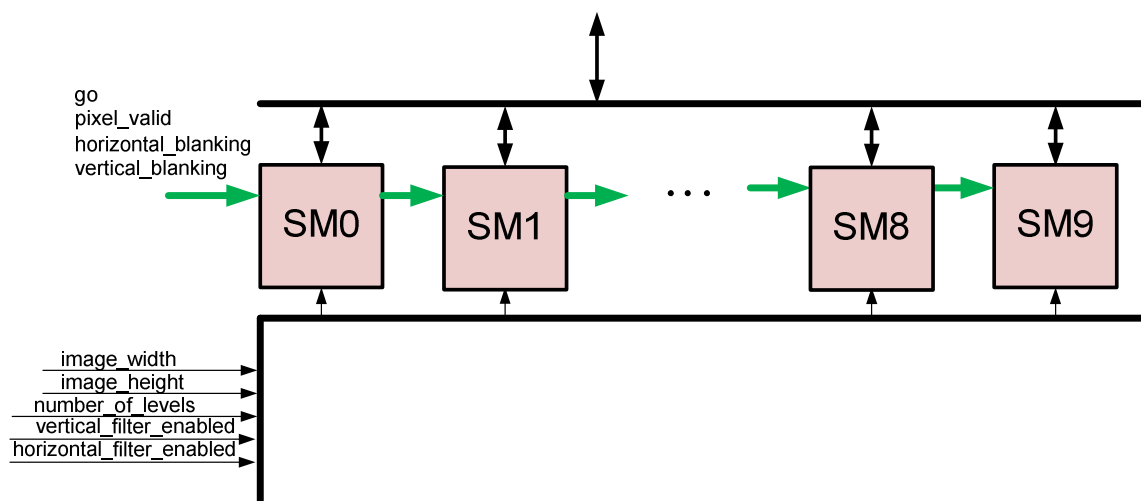
Slika 2.28 Realizacija Koeficijent memorije



## Direktni dvodimenzionalni podopsežni transformator



Slika 2.29 Realizacija Linijske memorije



Slika 2.30 Interna blok šema bloka D2DST KONTROLA

Blok D2DST ARBITRATOR sa **Slike 2.24** određuje koji od jednonivooskih transformatora dobija na raspolaganje filtre. Kako je već zaključeno da će postojati vremenski trenuci kada jednonivooski podopsežni transformatori sa više nivoa imaju spremne podatke za filtriranje, arbitracija dodele resursa je neophodna. Ilustracija ovoga data je na **Slici 2.31**. Kao što se vidi, zahtev za filtriranjem na nivou 2 opslužuje se sa

zakašnjenjem od jednog ciklusa, zahtev za filtriranjem na nivou 3 sa zakašnjenjem od tri ciklusa, na nivou 4 sa zakašnjenjem od 7 ciklusa, itd. Kašnjenje u filtriranju ne postoji na nivoima 0 i 1. Arbitrator je stoga jednostavno realizovati, kao koder prioriteta. Arbitrator preusmerava kontrolne signale iz bloka D2DST KONTROLA bloka, kao i odgovarajuće podatke iz Linijske i Koeficijent memorije na ulaze filtra. Arbitrator takođe na izlaz prosleđuje odgovarajuće kontrolne signale, i generiše signale za upis u memorije.

Blok Konverzija sa **Slike 2.24** vrši konverziju ulaznog piksela iz neoznačenog 8-bitnog formata u označeni 18.14 format sa nepokretnom tačkom ukoliko je u pitanju kompresija sa gubicima, odnosno u označeni 32-bitni celobrojni format ukoliko je u pitanju kompresija bez gubitaka. U oba slučaja se od vrednosti ulaznog piksela oduzima 128 kako bi srednja vrednost ulaznih piksela bila nula, i na taj način povećava raspoloživi dinamički opseg tokom izračunavanja.

Iskorišćenje hardverskih resursa i performanse nakon implementacije direktnog dvodimenzionalnog podopsežnog transformatora opisanog ranije, u integrisanom kolu Altera EP4CE115F29C7, korišćenjem softverskog alata Quartus II 12.1 prikazano je u **Tabeli 2.10**.

**Tabela 2.10**

Hardverski resursi i performanse direktnog dvodimenzionalnog podopsežnog transformatora u Altera EP4CE115F29C7

broj logičkih elemenata	<b>6465</b>
količina memorije	<b>192512 bita</b>
množači	<b>0</b>
maksimalna radna učestanost @85C	<b>138.27 MHz</b>

Pojedinačni naučni doprinos prikazan u ovom poglavlju obuhvata razvoj i optimizaciju direktnih nestacionarnih filtara za implementaciju u programabilnim logičkim kolima, kao i novi algoritam rada direktnog dvodimenzionalnog podopsežnog transformatora.

## Direktni dvodimenzionalni podposežni transformator

	L0	L1	L2	L3	L4
1					
2					
3	1				
4					
5	2				
6					
7	3				
8		1			
9	4				
10					
11	5				
12		2			
13	6				
14					
15	7				
16		3			
17	8		1		
18					
19	9				
20		4			
21	10				
22					
23	11				
24		5			
25	12		2		
26					
27	13				
28		6			
29	14				
30					
31	15				
32		7			
33	16		3		
34					
35	17			1	
36		8			
37	18				
38					
39	19				
40		9			
41	20		4		
42					

43	21				
44		10			
45	22				
46					
47	23				
48		11			
49	24		5		
50					
51	25			2	
52		12			
53	26				
54					
55	27				
56		13			
57	28		6		
58					
59	29				
60		14			
61	30				
62					
63	31				
64		15			
65	32		7		
66					
67	33			3	
68		16			
69	34				
70					
71	35				1
72		17			
73	36		8		
74					
75	37				
76		18			
77	38				
78					
79	39				
80		19			
81	40		9		
82					

**Slika 2.31** Vremenski dijagram opsluživanja zahteva za filterskim resursima za 4 nivoa dekompozicije

## Kvantizator

**K**vantizacija transformacionih koeficijenata se obavlja samo u slučaju kompresije sa gubicima. Transformacioni koeficijenti se direktno propuštaju kroz kvantizator u slučaju kompresije bez gubitaka.

Kvantizacija predstavlja proces približnog predstavljanja neprekidnog skupa vrednosti konačnim (poželjno malim) skupom vrednosti. Ulaz kvantizatora su originalni podaci, dok je njegov izlaz uvek jedan iz konačnog skupa nivoa. Ovo očigledno predstavlja aproksimaciju, i dobar kvantizator je onaj koji originalni signal predstavlja sa minimalnim gubicima ili izobličenjima.

Postoje dve vrste kvantizacije, skalarna i vektorska. Kod skalarne kvantizacije svaki ulazni simbol se posebno tretira prilikom određivanja izlaza, dok su kod vektorske kvantizacije ulazni simboli grupisani u vektore i obrađeni kako bi se odredila vrednost izlaza. Grupisanje podataka i tretiranje jedne grupe kao posebne jedinice povećava optimalnost ali na račun povećane složenosti.

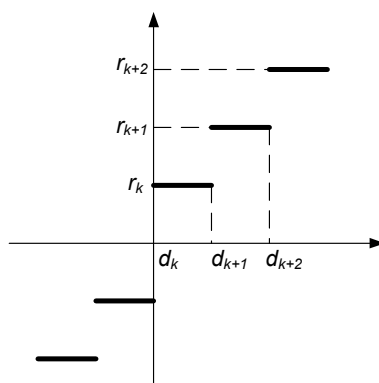
Skalarna kvantizacija omogućava da se pojedinačni transformacioni koeficijenti konvertuju u kvantizovanu vrednost konverzijom koja zavisi samo od tekućeg ulaznog koeficijenta. Najčešće realizacije skalarnog kvantizatora su uniformni kvantizator,

podopsežni uniformni kvantizator, uniformni kvantizator sa mrtvom zonom i neuniformni kvantizator [67].

Kod uniformne kvantizacije nivo kvantizacije je važan parametar, jer se na osnovu njega određuje veličina koraka kvantizacije. Visok nivo kvantizacije će proizvesti mali korak kvantizacije i obrnuto. Za dati nivo kvantizacije, korak kvantizacije je konstantan i kontroliše kvalitet slike. Veliki korak kvantizacije daje lošiju aproksimaciju, dok će manji korak kvantizacije dati bolju aproksimaciju. Korak kvantizacije se određuje kao

$$\Delta = \frac{\max - \min}{L}$$

gde je  $L$  broj nivoa kvantizacije. Opseg ulaznih podataka koji se kvantizuje deli sa na  $L+1$  jednakih intervala kako je ilustrovano na **Slici 3.1**.



**Slika 3.1** Ilustracija rada uniformnog kvantizatora

Za vreme kvantizacije koeficijent  $x$  se menja indeksom  $k$  ako  $x$  leži u datom opsegu

$$d_k \leq x \leq d_{k+1}$$

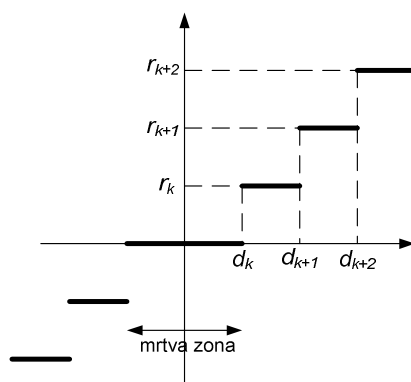
gde  $k \in [1, L]$ . Indeks  $k$  se koristi u dekvantizatoru kako bi se rekonstruisala originalna vrednost koeficijenta. Izobličenje se minimizuje ako su rekonstruisane vrednosti

$$r_k = \frac{d_k + d_{k+1}}{2}.$$

Karakteristika transformacionih koeficijenata je takva da su mnoge vrednosti jako male. Ova osobina se koristi tako što se definiše vrednost praga  $t$  na osnovu koje se takvi koeficijenti predstavljaju nulom

$$-t \leq x \leq t \Rightarrow y = 0.$$

Nakon ovoga se primenjuje uniformna kvantizacija. Ova tehnika zove se uniformna kvantizacija sa mrtvom zonom, i ilustrovana je na **Slici 3.2** uz pretpostavku da je  $t = d_k$ .



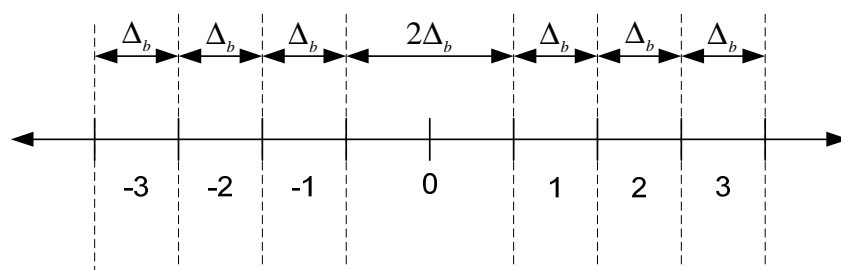
**Slika 3.2** Ilustracija rada uniformnog kvantizatora sa mrtvom zonom

Podopsežna uniformna kvantizacija se zasniva na činjenici da transformacioni koeficijenti nemaju isti dinamički opseg na svim podopsezima D2DST. Iz tog razloga se postiže bolji stepen kompresije ukoliko se transformacioni koeficijenti iz različitih podopsega kvantizuju sa različitim nivoima kvantizacije. Podopsežna uniformna kvantizacija se svodi na uniformnu kvantizaciju sa ili bez mrtve zone sa korakom kvantizacije koji zavisi od podopsega.

U ovom radu usvojena je implementacija kvantizatora kakva se koristi u JPEG2000 standardu [68]. Kvantizacija se sprovodi uniformnim skalarnim kvantizatorom sa mrtvom zonom čiji je korak  $\Delta_b$  i širina mrtve zone  $2\Delta_b$ , kako je prikazano na **Slici 3.3**. Za svaki podopseg se u opštem slučaju koriste različiti koraci kvantizacije. Veličina koraka kvantizacije  $\Delta_b$  za podopseg  $b$  računa se na osnovu dinamičkog opsega vrednosti transformacionih koeficijenata u podopsegu  $b$ . Jednačina koja opisuje rad uniformnog kvantizera sa mrtvom zonom glasi

$$q_b = \text{sign}(y_b) \cdot \left\lfloor \frac{|y_b|}{\Delta_b} \right\rfloor,$$

gde je  $y_b$  transformacioni koeficijent u podopsegu  $b$ , a  $\Delta_b$  korak kvantizacije za podopseg  $b$ . Iz jednačine se vidi da se prilikom deljenja vrši odsecanje. Rezultujući kvantizovani transformacioni koeficijent  $q_b$  je označeni broj. Izlazi direktnog dvodimenzionalnog podopsežnog transformatora su označeni brojevi u komplementu dvojke, ali su izlazi kvantizatora označeni brojevi u formatu znak + apsolutna vrednost. Ovo je posledica rada enkoderskog procenjivača verovatnoća i entropijskog enkodera koji koriste izlaze kvantizatora za svoj rad.



**Slika 3.3** Ilustracija rada uniformnog kvantizatora sa mrtvom zonom u JPEG2000 standardu

Kako bi se izbegla operacija deljenja, i na taj način smanjila složenost hardverske realizacije, u kvantizatoru realizovanom u ovoj tezi koraci kvantizacije definisani su na sledeći način

$$\Delta_{LHi,HLi} = M \cdot 2^{E+14-i}$$

Kako HHi podopsezi imaju veći dinamički opseg od LHi i HLi podopsega, definicija se u tom slučaju razlikuje

$$\Delta_{HHi} = M \cdot 2^{E+14-i+1}$$

LLi podopsezi imaju manji dinamički opseg od LHi i HLi podopsega, tako da se definicija i u tom slučaju razlikuje

$$\Delta_{LLi} = M \cdot 2^{E+14-i-1}$$

Parametar  $M$  je u opsegu  $[64,127]$ , dok je parametar  $E$  u opsegu  $[-6,6]$ . Množenje sa  $2^{14}$  je posledica činjenice da je ulazni format podatka u kvantizatora 18.14 sa fiksnom tačkom. Jednačine koje opisuju rad kvantizatora u ovom slučaju glase

$$q_{LHi} = q_{HLi} = \text{sign}(y_b) \cdot \left\lfloor \frac{|y_b|}{M \cdot 2^{E+14-i}} \right\rfloor = \text{sign}(y_b) \cdot \left\lfloor \frac{|y_b|}{M} \right\rfloor \cdot 2^{-(E+14-i)}$$

$$q_{HHi} = \text{sign}(y_b) \cdot \left\lfloor \frac{|y_b|}{M \cdot 2^{E+14-i+1}} \right\rfloor = \text{sign}(y_b) \cdot \left\lfloor \frac{|y_b|}{M} \right\rfloor \cdot 2^{-(E+14-i+1)}$$

Podopseg LLi se kvantizuje samo u slučaju poslednjeg nivoa dekompozicije, i u njemu je sadržan relativno najbitniji sadržaj, tako da se ne vrši odsecanje već zaokruživanje [69]

$$q_{LLi} = \text{sign}(y_b) \cdot \left\lfloor \frac{2|y_b| + M 2^{E+14-i-1}}{M 2^{E+14-i}} \right\rfloor = \text{sign}(y_b) \cdot \left\lfloor \frac{2|y_b| + M 2^{E+14-i-1}}{M} \right\rfloor \cdot 2^{-(E+14-i)}$$

Zbog ograničenog opsega parametra  $M$ , moguće je deljenje iz gornjih jednačina realizovati pomoću tabela za pretraživanje, a množenje sa stepenom broja dva kao pomeranje udesno. U tom slučaju, jednačine rada kvantizatora glase

$$q_{LHi} = q_{HLi} = \text{sign}(y_b) \cdot \begin{cases} \left( (|y_b| + LUT1[M - 65]) \cdot LUT2[M - 65] \right) \cdot 2^{-(E+14-i+38)}, & M > 64 \\ |y_b| \cdot 2^{-(E+14-i+6)}, & M = 64 = 2^6 \end{cases}$$

$$q_{HHi} = \text{sign}(y_b) \cdot \begin{cases} \left( (|y_b| + LUT1[M - 65]) \cdot LUT2[M - 65] \right) \cdot 2^{-(E+14-i+1+38)}, & M > 64 \\ |y_b| \cdot 2^{-(E+14-i+1+6)}, & M = 64 = 2^6 \end{cases}$$

$$q_{LLi} = \text{sign}(y_b) \cdot \begin{cases} \left( (2|y_b| + M 2^{E+14-i-1} + LUT1[M - 65]) \cdot LUT2[M - 65] \right) \cdot 2^{-(E+14-i+38)}, & M > 64 \\ (2|y_b| + M 2^{E+14-i-1}) \cdot 2^{-(E+14-i+6)}, & M = 64 = 2^6 \end{cases}$$

Tabele pretraživanja iz gornjih jednačina imaju po 63 elementa i izložene su u nastavku.



```

LUT1 =
[
    0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0,
    0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1,
    0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1,
    1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1
]

```

```

LUT2 =
[
    0xFC0FC0FD, 0xF83E0F84, 0xF4898D60,
    0xF0F0F0F1, 0xED7303B6, 0xEA0EA0EA, 0xE6C2B449,
    0xE38E38E4, 0xE070381C, 0xDD67C8A6, 0xDA740DA8,
    0xD79435E5, 0xD4C77B04, 0xD20D20D2, 0xCF6474A9,
    0CCCCCCCD, 0xCA4587E7, 0xC7CE0C7D, 0xC565C87C,
    0xC30C30C3, 0xC0C0C0C1, 0xBE82FA0C, 0xBC52640C,
    0xBA2E8BA3, 0xB81702E1, 0xB60B60B6, 0xB40B40B4,
    0xB21642C9, 0xB02C0B03, 0xAE4C415D, 0xAC769184,
    0AAAAAAAAB, 0xA8E83F57, 0xA72F053A, 0xA57EB503,
    0xA3D70A3E, 0xA237C32B, 0xA0A0A0A1, 0x9F1165E7,
    0x9D89D89E, 0x9C09C09C, 0x9A90E7D9, 0x991F1A51,
    0x97B425ED, 0x964FDA6C, 0x94F2094F, 0x939A85C4,
    0x92492492, 0x90FDBC09, 0x8FB823EE, 0x8E78356D,
    0x8D3DCB09, 0x8C08C08C, 0x8AD8F2FC, 0x89AE408A,
    0x88888889, 0x8767AB5F, 0x864B8A7E, 0x85340853,
    0x84210842, 0x83126E98, 0x82082082, 0x81020408
]

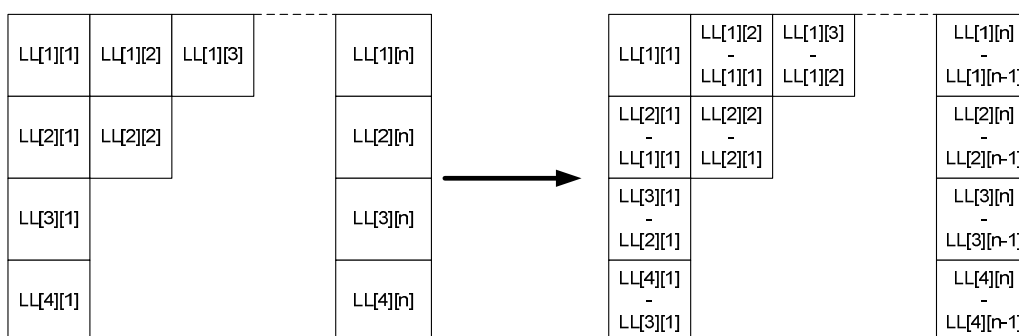
```

Usled načina rada enkoderskog procenjivača verovatnoća i entropijskog enkodera, blok kvantizator bi pored same kvantizacije trebalo da vrši i pohranjivanje blokova od četiri linije kvantizacionih koeficijenata iz svakog od LH, HL i HH podopsega, i razlika kvantizovanih koeficijenata za LL podopseg na poslednjem nivou (kako je ilustrovano na **Slici 3.4**). Takođe je potrebno određivanje i čuvanje maksimuma na svakom od tih blokova.

Postavlja se problem dinamike prenošenja kvantizovanih koeficijenata, a da se kapacitet korišćene memorije minimizuje. Minimalna ukupna količina potrebne blok memorije u jedinici kvantizovanih koeficijenata se izračunava kao

$$B = 3 \cdot 4 \cdot \sum_{i=1}^{n-1} \frac{W}{2^i} + 4 \cdot 4 \cdot \frac{W}{2^n} = 3 \cdot 4 \cdot \sum_{i=1}^n \frac{W}{2^i} + 4 \cdot \frac{W}{2^n} = 12 \cdot W \cdot \left(1 - \frac{1}{2^n}\right) + 4 \cdot \frac{W}{2^n} = 12 \cdot W - \frac{8 \cdot W}{2^n}$$

gde je  $n$  ukupan broj nivoa D2DST. Zbog različitog dinamičkog opsega kvantizovanih transformacionih koeficijenata na pojedinim podopsezima, moguće je optimizovati veličinu ove memorije.



**Slika 3.4** Princip generisanja signala razlike na LL podopsegu

Javlja se problem dinamike pisanja i čitanja u ovu memoriju, koji je ilustrovan na **Slici 3.5**, koja prikazuje generisanje odgovarajućih linija na izlazu D2DST, odnosno na ulazu kvantizatora, za vreme prvih 48 linija ulazne slike, ako postoji 5 nivoa dekompozicije. Kako je kvantizator protočni blok, ovakva dinamika je prisutna i na izlazu kvantizatora. Sa **Slike 3.5** se vidi da za vreme trajanja na primer petnaeste linije ulazne slike D2DST učešljano generiše sedmu liniju LH koeficijenata na nivou 0, treću liniju LH koeficijenata na nivou 1 i prvu liniju LH koeficijenata na nivou 2. Na **Slici 3.5** su označeni i blokovi četiri linije koji se moraju primiti i proslediti, uz maksimum, enkoderskom procenjivaču verovatnoća i entropijskom enkoderu na svakom pojedinačnom opsegu. Posmatrajmo šta se dešava na primer za vreme jedanaeste ulazne linije. Blok memorija od četiri linije na nivou 0 je popunjena za vreme trajanja prethodne linije. Blok memorija bi trebalo da se prazni dok istovremeno pristižu novi podaci na tom istom nivou. Da bi se uštedelo na kapacitetu potrebne memorije, neophodno je da se memorija čita brže nego što se u nju upisuje. Situacija se višestruko komplikuje ukoliko se nalazimo na višim linijama slike, kada se učešljano upisuje na nekoliko nivoa, uz istovremeno čitanje nekih od njih. Primer ovoga je ulazna linija 27. Postoje popunjeni blokovi na nivoima 0 i 1, a upisuju se podaci sa nivoa 0, 1 i 2. Kako

je trajanje pauze između linija ulazne slike parametar koji nije unapred poznat, ne može se sa sigurnošću računati da će biti dovoljno velike vrednosti da moguće anulira izvesne nedostatke algoritma čitanja memorije.

Detaljnom analizom se dolazi do zaključka da će se optimalno iskorišćenje memorije, bez mogućnosti greške u radu, postići ukoliko se iz popunjene memorije čita tačno po redosledu kojim se u nju upisuje. Ovo znači da će se, na primer, za vreme trajanja dvadeset treće linije ulazne slike istovremeno na nivou 0 upisivati u blok memoriju treća linija LH koeficijenata iz trećeg bloka i prema enkoderskom procenjivaču verovatnoća i entropijskom enkoderu prosleđivati treća linija LH koeficijenata iz drugog bloka. Pored toga, istovremeno će se na nivou 1 u memoriju upisivati prva linija LH koeficijenata iz drugog bloka i prema enkoderskom procenjivaču verovatnoća i entropijskom enkoderu prosleđivati prva linija LH koeficijenata iz prvog bloka. Iako se istovremeno u memoriju na nivou 2 upisuje druga linija LH koeficijenata iz prvog bloka, prema enkoderskom procenjivaču verovatnoća i entropijskom enkoderu ne prosleđuju se podaci iz nivoa 2 jer nije formiran ceo blok od četiri linije.

Kako broj linija neće na svim nivoima biti deljiv sa četiri, na kraju slike će moguće ostati nepotpuni blokovi na pojedinim podopsezima koji će se prosleđivati u nepromenjenoj veličini.

U navedenoj tehnici postoji još jedan nedostatak, a to je da će nakon što D2DST završi sa transformacijom ulazne slike i prestane da generiše transformacione koeficijente prestati upis u memoriju, a samim tim i čitanje iz memorije. Iz tog razloga je potrebna dodatna procedura za pražnjenje memorije po svim podopsezima na kraju jedne ulazne slike. Ova procedura će biti jednostavna ukoliko se iskoristi činjenica da će na svakom nivou u blok memoriji ostati po četiri linije koje nisu do tog trenutka isporučene enkoderskim procenjivačima verovatnoća i entropijskim enkoderima. Ovo nije ispunjeno samo ukoliko postoji toliko nivoa podopsežne transformacije u D2DST da je na dubljim nivoima dekompozicije sa ukupno  $N$  nivoa, počevši od nivoa  $j$ , prisutno manje od četiri linije transformacionih koeficijenata, odnosno ako je

$$\frac{H}{2^{j+1}} < 4, j < N$$

Kako je D2DST sa velikim brojem nivoa dekompozicije praktično nepotrebna [70], nadalje će se smatrati da je

$$\min\left(\frac{W}{2^N}, \frac{H}{2^N}\right) \geq 8, \text{ odnosno } N \leq \log_2(\min(W, H)) - 3$$

Za sliku visoke definicije (HD) dimenzija 1920 x 1080, ova relacija daje

$$N \leq 7$$

Prilikom projektovanja bloka kvantizatora postoji još jedna specifičnost. Naime, usled načina realizacije direktnog dvodimenzionalnog podopsežnog transformatora, na ulazu kvantizatora istovremeno se pojavljuju dva transformaciona koeficijenta. Sa druge strane, enkoderski procenjivač verovatnoća i entropijski enkoder obrađuju po jedan kvantizovani transformacioni koeficijent u jednom trenutku. Tako je u bloku kvantizatora potrebno implementirati i translaciju iz dvostrukog u serijski tok podataka. Redosled po kome enkoderski procenjivač verovatnoće i entropijski enkoder očekuju podatke na svom ulazu za jedan blok od četiri linije na jednom nivou D2DST prikazan je na **Slici 3.6**. Podaci iz LL podopsega postojeće samo na poslednjem nivou, dok će se linija podataka završavati podatkom iz LL ili HL podopsega ukoliko je dužina linije neparna.

Kako svaki podopseg u enkoderskom procenjivaču verovatnoće ima svoj skup histograma, i kako je operacija ažuriranja histograma relativno složena, poželjno je da su za ažuriranje histograma na raspolaganju dva ciklusa. U trenucima kada blok kvantizatora emituje liniju HL i HH koeficijenata ovo je samo po sebi zadovoljeno. Međutim, kada blok kvantizatora emituje liniju LH koeficijenata (uvek kada nije u pitanju poslednji nivo), potrebno je umetnuti lažne cikluse. Ulazni i izlazni podaci bloka kvantizatora za nekoliko linija sa **Slike 3.5** dati su na **Slikama 3.7 a)-g)**.

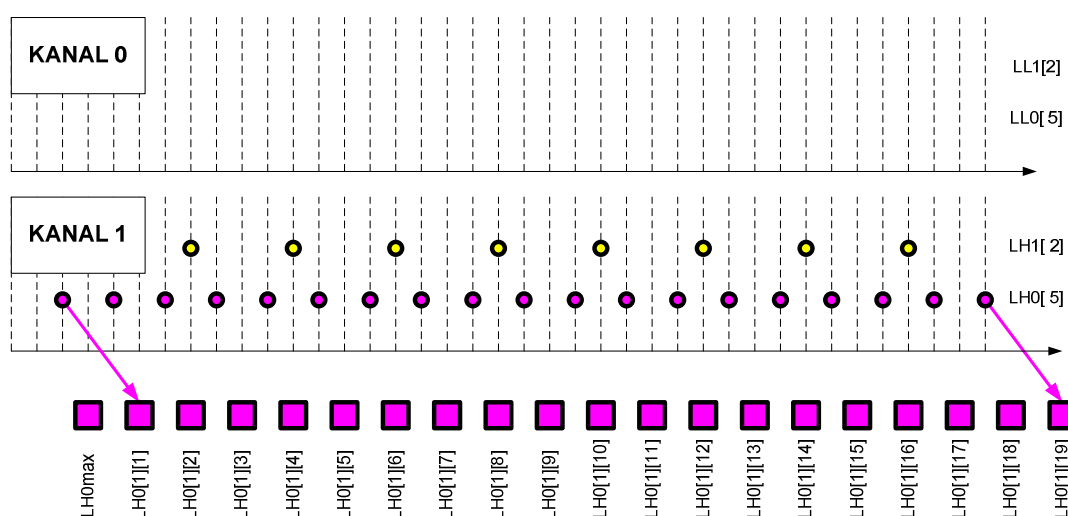
Ulaz	L0	L1	L2	L3
1				
2				
3	1 (LH)			
4	1(HL,HH)			
5	2(LH)			
6	2(HL,HH)			
7	3(LH)	1 (LH)		
8	3(HL,HH)			
9	4(LH)	1(HL,HH)		
10	4(HL,HH)			
11	5(LH)	2(LH)		
12	5(HL,HH)			
13	6(LH)	2(HL,HH)		
14	6(HL,HH)			
15	7(LH)	3(LH)	1 (LH)	
16	7(HL,HH)			
17	8(LH)	3(HL,HH)		
18	8(HL,HH)			
19	9(LH)	4(LH)	1(HL,HH)	
20	9(HL,HH)			
21	10(LH)	4(HL,HH)		
22	10(HL,HH)			
23	11(LH)	5(LH)	2(LH)	
24	11(HL,HH)			
25	12(LH)	5(HL,HH)		
26	12(HL,HH)			
27	13(LH)	6(LH)	2(HL,HH)	
28	13(HL,HH)			
29	14(LH)	6(HL,HH)		
30	14(HL,HH)			
31	15(LH)	7(LH)	3(LH)	1 (LL,LH)
32	15(HL,HH)			
33	16(LH)	7(HL,HH)		
34	16(HL,HH)			
35	17(LH)	8(LH)	3(HL,HH)	
36	17(HL,HH)			
37	18(LH)	8(HL,HH)		
38	18(HL,HH)			
39	19(LH)	9(LH)	4(LH)	1(HL,HH)
40	19(HL,HH)			
41	20(LH)	9(HL,HH)		
42	20(HL,HH)			
43	21(LH)	10(LH)	4(HL,HH)	
44	21(HL,HH)			
45	22(LH)	10(HL,HH)		
46	22(HL,HH)			
47	23(LH)	11(LH)	5(LH)	2 (LL,LH)
48	23(HL,HH)			

Slika 3.5 Vremenski dijagram generisanja linija na izlazima D2DST za 4 nivoa dekompozicije

LLmax	LHmax	LL[1][1]	LH[1][1]	LL[1][2]	LH[1][2]			LL[1][w]	LH[1][w]	LL[1][w]
HLmax	HHmax	HL[1][1]	HH[1][1]	HL[1][2]	HH[1][2]			HL[1][w]	HH[1][w]	HL[1][w]
		LL[2][1]	LH[2][1]	LL[2][2]	LH[2][2]			LL[2][w]	LH[2][w]	LL[2][w]
		HL[2][1]	HH[2][1]	HL[2][2]	HH[2][2]			HL[2][w]	HH[2][w]	HL[2][w]
		LL[3][1]	LH[3][1]	LL[3][2]	LH[3][2]			LL[3][w]	LH[3][w]	LL[3][w]
		HL[3][1]	HH[3][1]	HL[3][2]	HH[3][2]			HL[3][w]	HH[3][w]	HL[3][w]
		LL[4][1]	LH[4][1]	LL[4][2]	LH[4][2]			LL[4][w]	LH[4][w]	LL[4][w]
		HL[4][1]	HH[4][1]	HL[4][2]	HH[4][2]			HL[4][w]	HH[4][w]	HL[4][w]

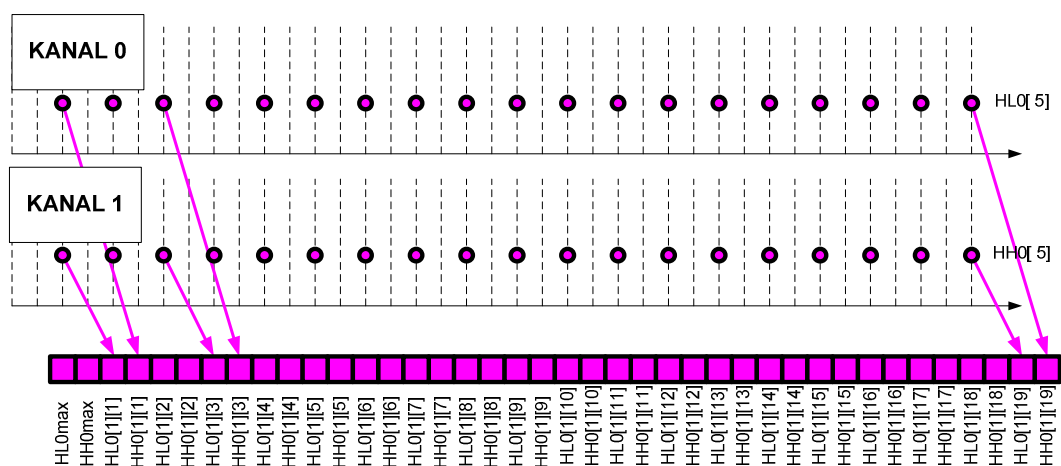
**Slika 3.6** Redosled po kojem enkoderski procenjivač verovatnoće i entropijski enkoder prihvataju jedan blok podataka (četiri linije) na jednom nivou dekompozicije

Na **Slici 3.7 a)** prikazani su ulazni i izlazni podaci bloka kvantizatora na početku jedanaeste ulazne linije. D2DST generiše petu po redu liniju LH koeficijenata na nivou 0, po jedan koeficijent u svakom drugom ciklusu počevši od trećeg ciklusa. Kako je blok memorija za LH podopseg na nivou 0 popunjena sa četiri linije, najpre se na izlaz prosleđuje maksimum LH podopsega na nivou 0 na te četiri linije, praćen redom LH koeficijentima iz prve linije LH koeficijenata na nivou 0. Istovremeno D2DST generiše drugu po redu liniju LH koeficijenata na nivou 1. Kako blok memorija za LH podopseg na nivou 1 nije popunjena, nema generisanja koeficijenata iz ovog podopsega na izlazu.



**Slika 3.7 a)** Ulaz i izlaz bloka kvantizatora za vreme obrade jedanaeste ulazne linije

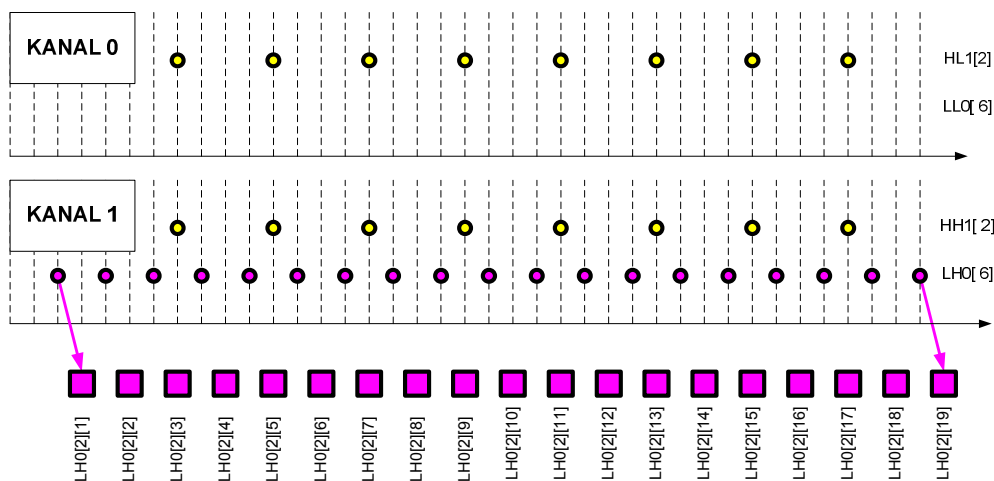
Na **Slici 3.7 b)** prikazani su ulazni i izlazni podaci bloka kvantizatora na početku dvanaeste ulazne linije. D2DST generiše petu po redu liniju HL i HH koeficijenata na nivou 0, po dva koeficijenta u svakom drugom ciklusu počevši od trećeg ciklusa. Kako je blok memorija za HL i HH podopseg na nivou 0 popunjena sa četiri linije, najpre se na izlaz prosleđuju jedan za drugim maksimumi HL i HH podopsega na nivou 0 na te četiri linije, praćeni naizmenično HL i HH koeficijentima iz prvih linija HL i HH koeficijenata na nivou 0.



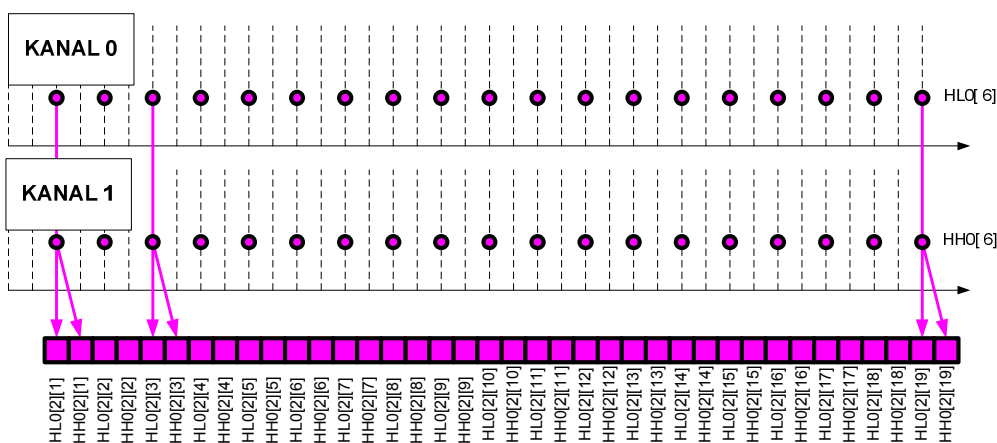
**Slika 3.7 b)** Ulaz i izlaz bloka kvantizatora za vreme obrade dvanaeste ulazne linije

Na **Slici 3.7 c)** prikazani su ulazni i izlazni podaci bloka kvantizatora na početku trinaeste ulazne linije. D2DST generiše šestu po redu liniju LH koeficijenata na nivou 0, po jedan koeficijent u svakom drugom ciklusu počevši od trećeg ciklusa. Na izlaz se iz blok memorije prosleđuju redom LH koeficijenti iz druge linije LH koeficijenata na nivou 0. Istovremeno D2DST generiše drugu po redu liniju HL i HH koeficijenata na nivou 1. Kako je blok memorija za HL i HH podopsege na nivou 1 prazna, nema generisanja koeficijenata iz ovih podopsega na izlazu.

Na **Slici 3.7 d)** prikazani su ulazni i izlazni podaci bloka kvantizatora na početku četrnaeste ulazne linije. D2DST generiše šestu po redu liniju HL i HH koeficijenata na nivou 0, po dva koeficijenta u svakom drugom ciklusu počevši od trećeg ciklusa. Na izlaz se iz blok memorije prosleđuju naizmenično HL i HH koeficijenti iz drugih linija HL i HH koeficijenata na nivou 0.



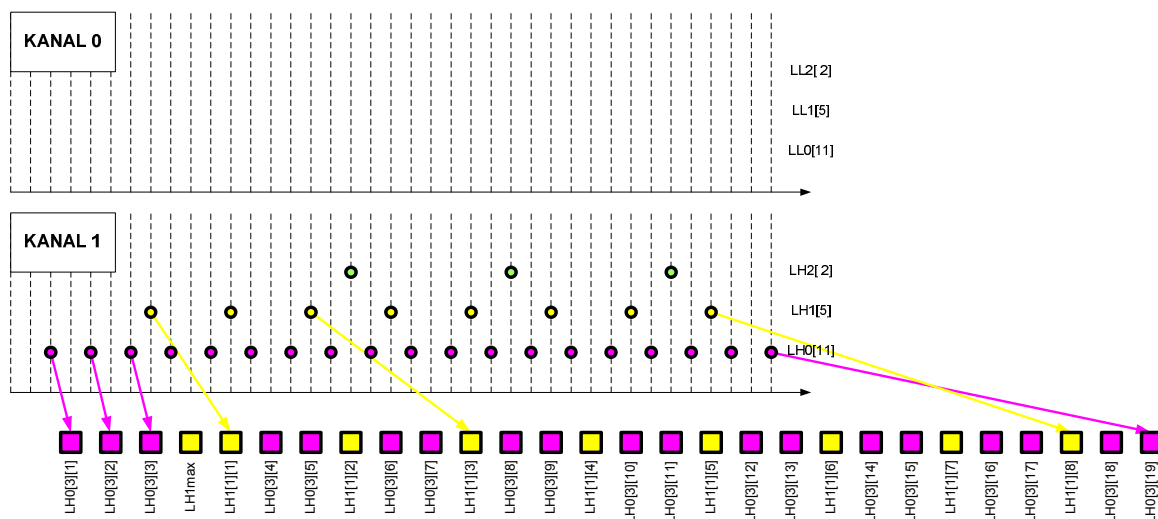
Slika 3.7 c) Ulaz i izlaz bloka kvantizatora za vreme obrade trinaeste ulazne linije



Slika 3.7 d) Ulaz i izlaz bloka kvantizatora za vreme obrade četrnaeste ulazne linije

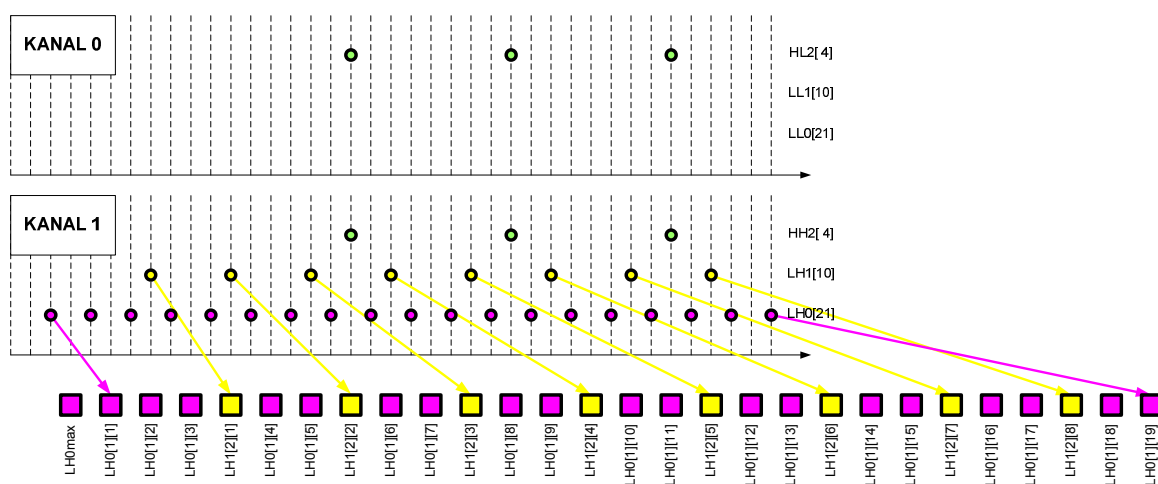
Na Slici 3.7 e) prikazani su ulazni i izlazni podaci bloka kvantizatora na početku dvadeset treće ulazne linije. D2DST generiše jedanaestu po redu liniju LH koeficijenata na nivou 0, po jedan koeficijent u svakom drugom ciklusu počevši od trećeg ciklusa. Na izlaz se iz blok memorije prosleđuju redom LH koeficijenti iz druge linije LH koeficijenata na nivou 0. U isto vreme D2DST generiše petu po redu liniju LH koeficijenata na nivou 1. Kako je blok memorija za LH podopseg na nivou 1 popunjena sa četiri linije, najpre se na izlaz prosleđuje maksimum LH podopsega na nivou 1 na te četiri linije, praćen redom LH koeficijentima iz prve linije LH koeficijenata na nivou 1. Istovremeno D2DST generiše i drugu po redu liniju LH koeficijenata na nivou 2. Kako blok memorija za LH podopseg na nivou 2 nije popunjena, nema generisanja koeficijenata iz ovog podopsega na izlazu.





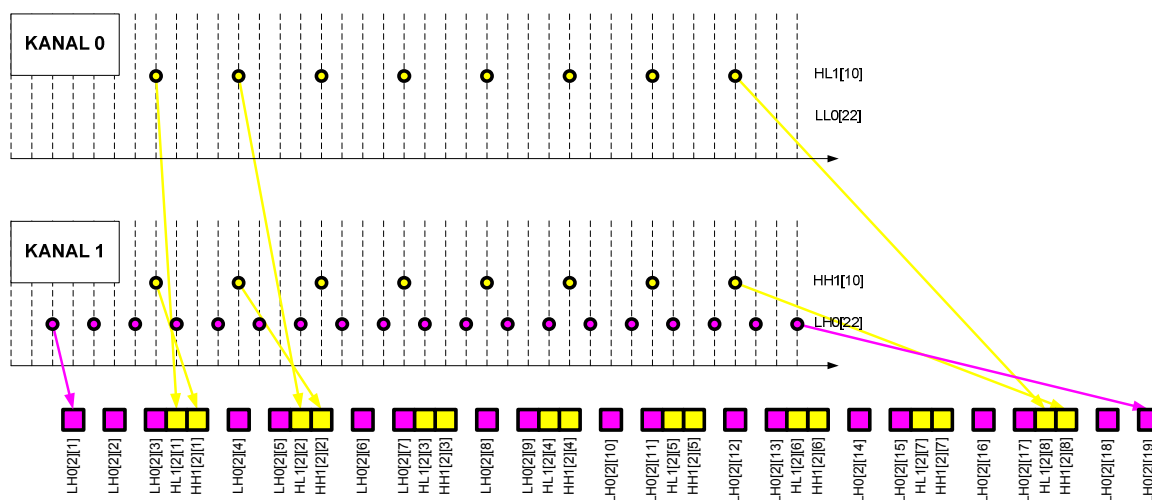
Slika 3.7 e) Ulaz i izlaz bloka kvantizatora za vreme obrade dvadeset treće ulazne linije

Na Slici 3.7 f) prikazani su ulazni i izlazni podaci bloka kvantizatora na početku četrdeset treće ulazne linije. D2DST generiše dvadeset prvu po redu liniju LH koeficijenata na nivou 0, po jedan koeficijent u svakom drugom ciklusu počevši od trećeg ciklusa. Kako je blok memorija za LH podopseg na nivou 0 popunjena sa četiri linije, najpre se na izlaz prosleđuje maksimum LH podopsega na nivou 0 na te četiri linije, praćen redom LH koeficijentima iz prve linije LH koeficijenata na nivou 0. Istovremeno D2DST generiše desetu po redu liniju LH koeficijenata na nivou 1. Na izlaz se iz blok memorije prosleđuju redom LH koeficijenti iz druge linije LH koeficijenata na nivou 1. Istovremeno D2DST generiše drugu po redu liniju HL i HH koeficijenata na nivou 2. Kako blok memorija za HL i HH podopsege na nivou 2 nije popunjena, nema generisanja koeficijenata iz ovih podopsega na izlazu.



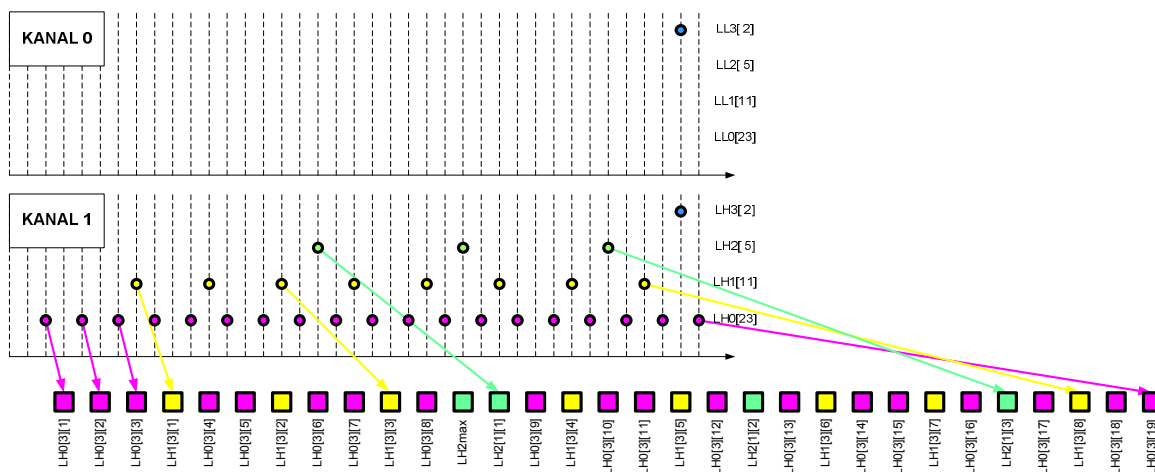
Slika 3.7 f) Ulaz i izlaz bloka kvantizatora za vreme obrade četrdeset treće ulazne linije

Na **Slici 3.7 g**) prikazani su ulazni i izlazni podaci bloka kvantizatora na početku četrdeset pete ulazne linije. D2DST generiše dvadeset drugu po redu liniju LH koeficijenata na nivou 0, po jedan koeficijent u svakom drugom ciklusu počevši od trećeg ciklusa. Na izlaz se iz blok memorije prosleđuju redom LH koeficijenti iz druge linije LH koeficijenata na nivou 0. Istovremeno D2DST generiše desetu po redu liniju HL i HH koeficijenata na nivou 1. Na izlaz se iz blok memorije prosleđuju redom LH koeficijenti iz druge linije LH koeficijenata na nivou 1.



**Slika 3.7 g)** Ulaz i izlaz bloka kvantizatora za vreme obrade četrdeset pete ulazne linije

Na **Slici 3.7 h**) prikazani su ulazni i izlazni podaci bloka kvantizatora na početku četrdeset sedme ulazne linije. D2DST generiše dvadeset treću po redu liniju LH koeficijenata na nivou 0, po jedan koeficijent u svakom drugom ciklusu počevši od trećeg ciklusa. Na izlaz se iz blok memorije prosleđuju redom LH koeficijenti iz treće linije LH koeficijenata na nivou 0. U isto vreme D2DST generiše jedanaestu po redu liniju LH koeficijenata na nivou 1. Na izlaz se iz blok memorije prosleđuju redom LH koeficijenti iz treće linije LH koeficijenata na nivou 1. Istovremeno D2DST generiše petu po redu liniju LH koeficijenata na nivou 2. Kako je blok memorija za LH podopseg na nivou 2 popunjena sa četiri linije, najpre se na izlaz prosleđuje maksimum LH podopsega na nivou 2 na te četiri linije, praćen redom LH koeficijentima iz prve linije LH koeficijenata na nivou 2. U isto vreme D2DST generiše drugu po redu liniju LH i LL koeficijenata na nivou 3. Kako blok memorija za LH podopseg na nivou 3 nije popunjena, nema generisanja koeficijenata iz ovog podopsega na izlazu.



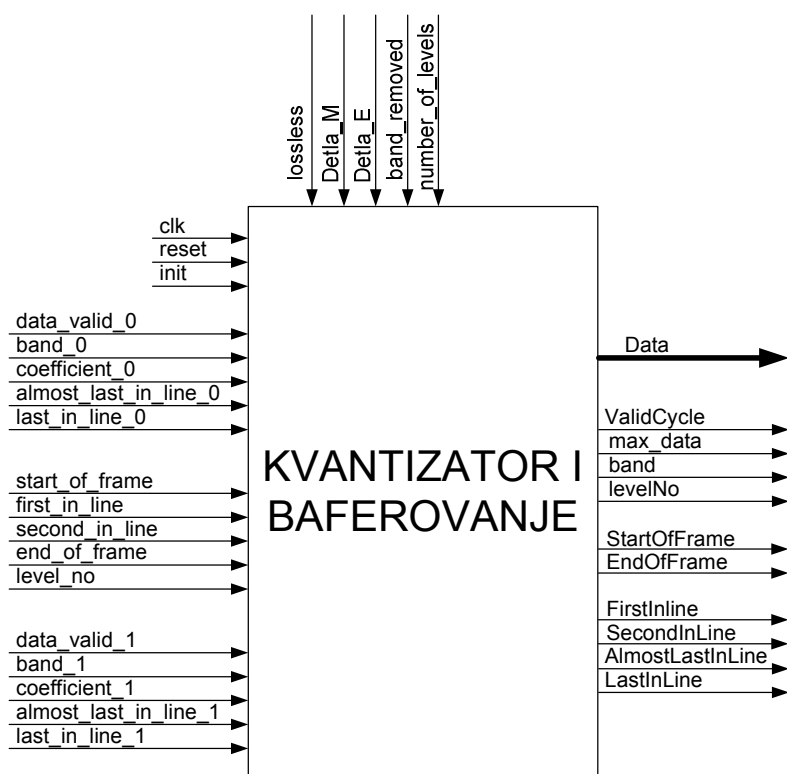
Slika 3.7 h) Ulaz i izlaz bloka kvantizatora za vreme obrade četrdeset sedme ulazne linije

Iz analize rada bloka kvantizatora nameće se očigledan zaključak da je u okviru ovoga bloka neophodno pored blok memorije implementirati i liniju za kašnjenje u formi FIFO memorije jer je izlaz serijski dok je ulaz dvoparalelni. Kapacitet ove FIFO memorije bi morao biti jednak  $W / 2$  reči, gde jedna reč obuhvata dva podatka uz prateće kontrolne signale.

## Hardverska realizacija kvantizatora sa baferovanjem

Najviši hijerarhijski nivo hardverske realizacije kvantizatora sa baferovanjem za jednu od YUV ravni boje prikazan je na Slici 3.8. Značenja pojedinih signala sa data su u Tabeli 3.1.

Ulazni signal *number\_of\_levels* signalizira koliko nivoa podopsežne transformacije postoji. Arhitektura (veličina blok memorije) trenutno podržava maksimalno 7 nivoa dekompozicije. Ovaj parametar je bitan da bi blok kvantizator sa baferovanjem korektno ispraznio blok memoriju kada se završi generisanje transformacionih koeficijenata.



Slika 3.8 Najviši hijerarhijski nivo hardverske realizacije kvantizatora sa baferovanjem

Tabela 3.1

Signali na najvišem hijerarhijskom nivou kvantizatora sa baferovanjem

Ime signala	Širina signala	Smer signala	Funkcija signala
coefficient_0	32 bita	ulaz	LL ili HL transformacioni koeficijent
coefficient_1	32 bita	ulaz	LH ili HH transformacioni koeficijent
level_no	4 bita	ulaz	označava redni broj nivoa dekompozicije kome pripadaju coefficient_0 i/ili coefficient_1
data_valid_0	1 bit	ulaz	kvalifikator za ulaz coefficient_0
data_valid_1	1 bit	ulaz	kvalifikator za ulaz coefficient_1
band_0	2 bita	ulaz	podopseg kome pripada coefficient_0
band_1	2 bita	ulaz	podopseg kome pripada coefficient_1
start_of_frame	1 bit	ulaz	ako je aktivan, signalizira da su coefficient_0 i/ili coefficient_1 prvi koeficijenti na datom nivou dekompozicije za datu ulaznu sliku
end_of_frame	1 bit	ulaz	ako je aktivan, signalizira da su coefficient_0 i/ili coefficient_1 poslednji za datu ulazni sliku
first_in_line	1 bit	ulaz	signalizira da su coefficient_0 i/ili coefficient_1 prvi u tekućoj liniji transformacionih koeficijenata

## Kvantizator

second_in_line	1 bit	ulaz	signalizira da su coefficient_0 i/ili coefficient_1 drugi u tekućoj liniji transformacionih koeficijenata
almost_last_in_line_0	1 bit	ulaz	signalizira da je coefficient_0 pretposlednji u tekućoj liniji transformacionih koeficijenata
almost_last_in_line_1	1 bit	ulaz	signalizira da je coefficient_1 pretposlednji u tekućoj liniji transformacionih koeficijenata
last_in_line_0	1 bit	ulaz	signalizira da je coefficient_0 poslednji u tekućoj liniji transformacionih koeficijenata
last_in_line_1	1 bit	ulaz	signalizira da je coefficient_1 poslednji u tekućoj liniji transformacionih koeficijenata
Delta_M	8 bita	ulaz	mantisa osnovnog koraka kvantizacije
Delta_E	8 bita	ulaz	eksponent osnovnog koraka kvantizacije
band_removed	30 bita	ulaz	signalizira koji od podopsega ne postoje i/ili ne bi trebalo da postoje
lossless	1 bit	ulaz	signalizira da li je kompresija bez ili sa gubicima
number_of_levels	4 bita	ulaz	definiše ukupan broj nivoa dekompozicije
Data	17 bita	izlaz	izlazni podatak u formatu znak + apsolutna vrednost
ValidCycle	1 bit	izlaz	kvalifikator za izlaz
max_data	1 bit	izlaz	kvalifikator da li je izlazni podatak ili kvantizovani transformacioni koeficijent
band	2 bita	izlaz	podopseg u opsegu nivoa kojem pripada Data
levelNo	4 bita	izlaz	nivo kojem pripada data
StartOfFrame	1 bit	izlaz	aktivan uz prvi maksimum svakog nivoa svake ulazne slike
EndOfFrame	1 bit	izlaz	nezavisan signal koji signalizira da više nema podataka na datom nivou svake ulazne slike
FirstInLine	1 bit	izlaz	signalizira da je Data prvi koeficijent u tekućoj liniji transformacionih koeficijenata na svakom od podopsega posebno
SecondInLine	1 bit	izlaz	signalizira da je Data drugi koeficijent u tekućoj liniji transformacionih koeficijenata na svakom od podopsega posebno
AlmostLastInLine	1 bit	izlaz	signalizira da je Data pretposlednji

			koeficijent u tekućoj liniji transformacionih koeficijenata na svakom od podopsega posebno
LastInLine	1 bit	izlaz	signalizira da je Data poslednji koeficijent u tekućoj liniji transformacionih koeficijenata na svakom od podopsega posebno

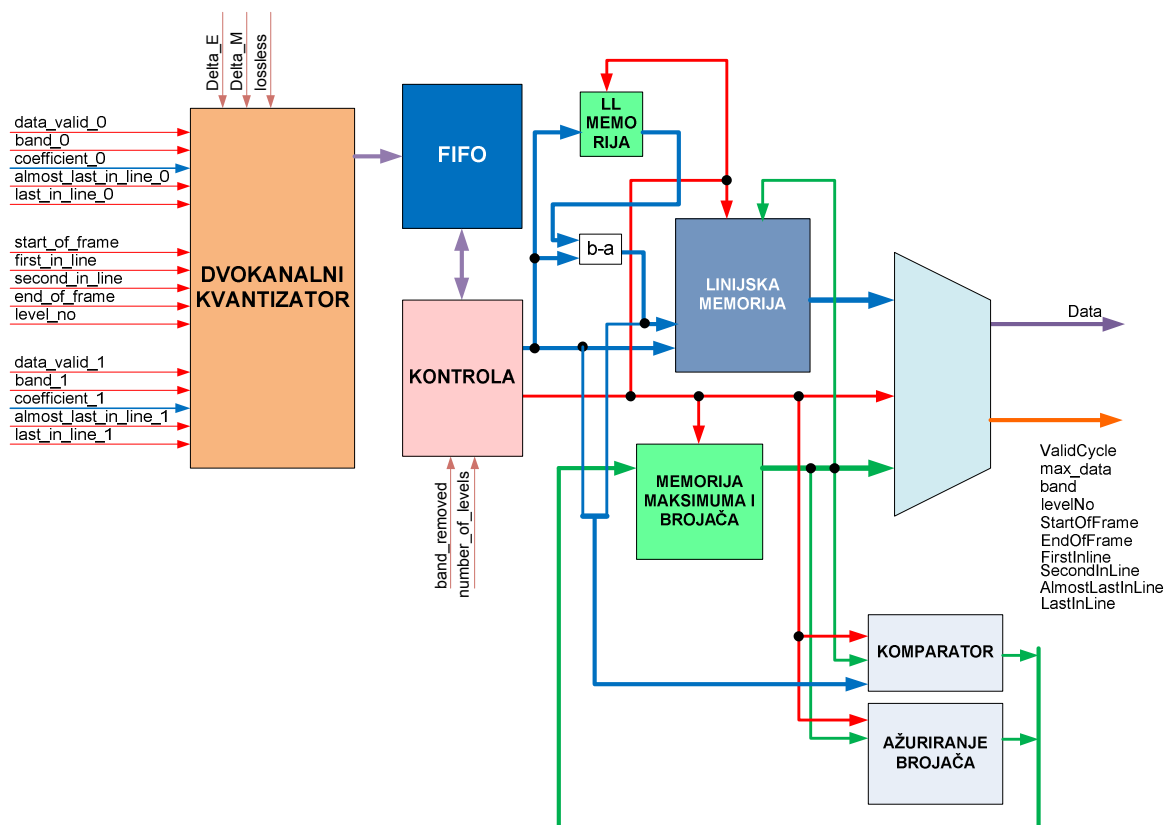
Ulazni signal *band\_removed* koristi se zbog različitih scenarija D2DST u zavisnosti od formata ulazne YUV slike. Ukoliko je ulazna slika u formatu 4:2:2 na nivou 0 neće postojati podposezi LH i HH. Ukoliko je ulazna slika u formatu 4:2:0 na nivou 0 neće postojati nijedan podopseg (**Slika 2.23**). Pored toga, moguće je je podopsege koji postoje jednostavno ukloniti iz skupa podataka korišćenjem ovog signala. Signal je dimenzionisan za maksimalno deset nivoa D2DST.

Ulazni signali **Delta\_M** i **Delta\_E** određuju osnovni korak kvantizacije

$$\Delta = \text{Delta\_M} \cdot 2^{\text{Delta\_E}+14}$$

Ulazni signal **lossless** se u bloku kvantizatora sa baferovanjem koristi kao signalizacija da li se ulazni podaci kvantizuju ili propuštaju neizmenjeni osim promene formata iz celobrojnog komplementa dvojke u format znak + apsolutna vrednost.

Valja primetiti da među ulaznim signalima nema onih koji definišu veličinu ulazne slike. Veličina pojedinih podopsega se signalizira pomoću signala *first\_in\_line* i *last\_in\_line\_0* i *last\_in\_line\_1*. Dva posebna signala za kraj linije potrebna su jer su, ukoliko je ulazna slika na nekom podopsegu neparne dužine, linije LL i HL koeficijenata na tom podopsegu su za jedan duže od linija LH i HH koeficijenata na tom podopsegu. Iz istog razloga potrebna su dva signala *almost\_last\_in\_line\_0* i *almost\_last\_in\_line\_1*, koji se, zajedno sa signalom *second\_in\_line*, koriste u enkoderskom procenjivaču verovatnoća. Broj nivoa dekompozicije nije neophodan za funkcionisanje bloka kvantizatora sa baferovanjem dokle god je taj broj manji od maksimalno projektovanog.

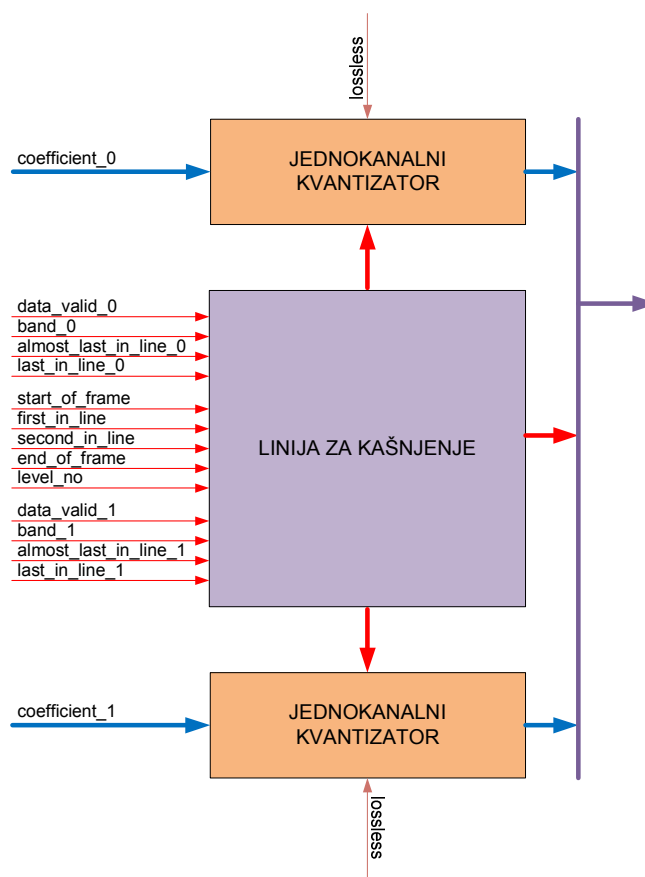


Slika 3.9 Blok šema kvantizatora sa baferovanjem

Blok šema realizacije kvantizatora sa baferovanjem sa **Slike 3.8** prikazana je na **Slici 3.9**.

Blok DVOKANALNI KVANTIZATOR sa **Slike 3.9** predstavlja dva pojedinačna kvantizatora, kako je prikazano na **Slici 3.10**. Jednokanalni kvantizatori kvantizuju svaki po jedan od para ulaznih koeficijenata, i to protočnom obradom kako bi se postigla veća brzina rada. Linija za kašnjenje obezbeđuje zakašnjene kontrolne signale za jednokanalne kvantizatore, kao i zakašnjene kontrolne signale koji se zajedno sa kvantizovanim transformacionim koeficijentima prosleđuju sledećem bloku.

FIFO memorija sa **Slike 3.9**, kako je ranije navedeno, obezbeđuje privremeno pohranjivanje podataka kako bi se na izlazu bloka kvantizatora sa baferovanjem podaci prosleđivali prema protokolu sa **Slike 3.6**.

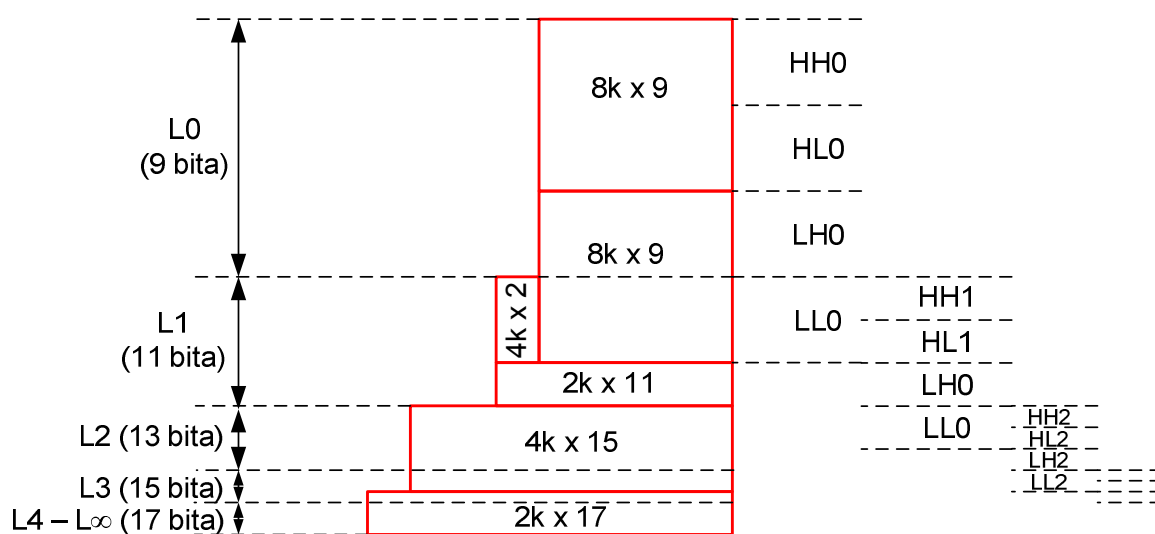


Slika 3.10 Blok šema dvokanalnog kvantizatora

Blok Kontrola sa **Slike 3.9** predstavlja mašinu stanja koja obezbeđuje izlazni protokol sa **Slike 3.6**, u skladu sa vremenskim zahtevima ilustrovanim na **Slikama 3.7 a)-h)**. U ovom bloku vrši se brojanje linija ulaznih koeficijenata, umetanje maksimuma u izlazni niz itd.

Blok LINIJSKA MEMORIJA sa **Slike 3.9** predstavlja blok memoriju od četiri linije svakog podopsega. U ovoj izvedbi, podržan je proizvoljan broj nivoa D2DST za ulaznu sliku maksimalne širine  $W = 1024$ . Činjenica da je broj nenultih binarnih cifara različit na različitim nivoima dekompozicija iskorišćena je u cilju optimizacije utroška memorijskih blokova koji su u Altera Cyclone IV familiji organizovani u blokove od po 9kbita. Fizička i logička realizacija blok memorije prikazana je na **Slici 3.11**.





Slika 3.11 Blok memorija

Širina memorije na prvih 12k adresa je 9 bita, na adresama 12k-18k je 11 bita, u opsegu adresa 18k-22k je 15 bita, i u opsegu 22k-24k je 17 bita.

Memorija maksimuma i brojača sa **Slike 3.9** čuva trenutnu vrednost maksimuma na svim blokovima svih podopsega sa indeksom sledećeg kvantizovanog transformacionog koeficijenta koji će se upisivati i/ili čitati. Kada bi prema protokolu trebalo na izlaz proslediti maksimum na bloku linija nekog podopsega, prosleđuje se maksimum iz ove memorije. Paralelno sa upisom novog kvantizovanog koeficijenta u blok memoriju, maksimum koji se čuva u memoriji upoređuje se trenutnom vrednošću kvantizovanog koeficijenta u bloku Komparator sa **Slike 3.9** i ažurira njegova vrednost. Indeks kvantizovanog koeficijenta je u opsegu  $[0, 4 \cdot W_j - 1]$ , gde je  $W_j$  širina linije koeficijenta podopsega na nivou  $j$ . Indeks se inkrementira prilikom svakog upisa u memoriju, i vraća na nulu na početku upisa novog bloka od četiri linije.

Oduzimač sa **Slike 3.9** koristi se za generisanje razlike koja se pohranjuje u slučaju LL podopsega, prema **Slici 3.4**. Vrednosti prethodnog koeficijenta u liniji kao i prvog koeficijenta iz prethodne linije čuvaju se u LL memoriji realizovanoj kao dva registra.

Izlazni multiplekser na izlaz prosleđuje odgovarajuće podatke (maksimum ili podatak iz blok memorije) uz kontrolne podatke prema definisanom protokolu. Kao specijalan slučaj se pojavljuje komanda narednom stepenu da je za datu ulaznu sliku završena obrada na nekom nivou (kada entropijski enkoder treba da završi svoju operaciju), kada se prosleđuju aktivan signal EndOfFrame i broj nivoa koji je završen.

Iskorišćenje hardverskih resursa i performanse nakon implementacije bloka kvantizatora sa baferovanjem opisanog ranije, u integrisanom kolu Altera EP4CE115F29C7, korišćenjem softverskog alata Quartus II 12.1 prikazano je u **Tabeli 3.2**.

**Tabela 3.2**

Hardverski resursi i performanse kvantizatora sa baferovanjem u Altera EP4CE115F29C7

broj logičkih elemenata	<b>3361</b>
količina memorije	<b>329664 bita</b>
množači (9-bitni)	<b>28</b>
maksimalna radna učestanost @85C	<b>115.11 MHz</b>

U ovom poglavlju prikazani su rezultati analize rada i optimizacije hardverske realizacije kvantizatora sa baferovanjem. Naučno postignuće predstavlja definisani novi protok podataka unutar modula enkodera slike.

## Enkoderski procenjivač verovatnoća i entropijski enkoder

**K**vantizovani ili nekvantizovani transformacioni koeficijenti (u zavisnosti od toga da li je kompresija sa ili bez gubitaka) u svakom podopsegu se enkoduju odvojeno bez ikakve informacije iz bilo kog drugog podopsega.

Jedan broj autora je uočio da podopsežni transformacioni koeficijenti imaju izrazito ne-Gausovsku statistiku [71]-[74]. Razlog je prostorna struktura tipičnih slika, koja se sastoji od glatkih oblasti prošaranih sa oštrim ivicama. Glatke oblasti proizvode skoro nulte transformacione koeficijente, dok strme ivice proizvode transformacione koeficijente velike amplitude. Statistika transformacionih koeficijenata se može modelovati pomoću dvoparametarske “uopštene Laplasove” funkcije gustine, koja ima oštar vrh u nuli, sa izraženijim repovima u odnosu na Gausovu funkciju gustine [75], [76]. Nažalost, dvoprolazni algoritam je potreban za izračunavanje parametara funkcije

gustine. Pored toga, eksperimentalni rezultati prikazuju značajnu razliku između ove funkcije gustine i stvarnih histograma samo na višim nivoima podopsežne transformacije. Podopseg niskih učestanosti sadrži skoro samo pozitivne transformacione koeficijente, odgovarajuće uniformnoj funkciji gustine.

Veći stepeni kompresije se mogu postići definisanjem simbola na osnovu modela konteksta, tj. na osnovu susednih transformacionih koeficijenata analogno postupcima za kompresiju teksta. Analiza nultih stabala i ostalih kontekstnih modela sa koeficijentima talasića je objašnjena u [77]-[80].

Moguće je pronaći bitski kod, koji je efikasniji od koda fiksne dužine, ako je poznata verovatnoća pojavljivanja pojedinog simbola. Dodela kodnih reči se obično obavlja kodovanjem promenljive dužine, "run-length" kodovanjem, Hafmanovim kodovanjem i aritmetičkim kodovanjem. Tehnike za uklanjanje alfabetske suvišnosti većinom proizvode prefiksne kodove, i većinom transformišu poruke u bitski niz, dodelom dužih kodova simbolima sa manjom verovatnoćom, kao u [81]-[83].

Najveći stepen kompresije je postignut aritmetičkim kodovanjem, koje teoretski može ukloniti svu suvišnu informaciju iz digitalizovane poruke, prema [84]-[86]. Aritmetički Q-coder je obelodanjen u [87]-[90], dok je aritmetički Z-coder obelodanjen u [91]-[94].

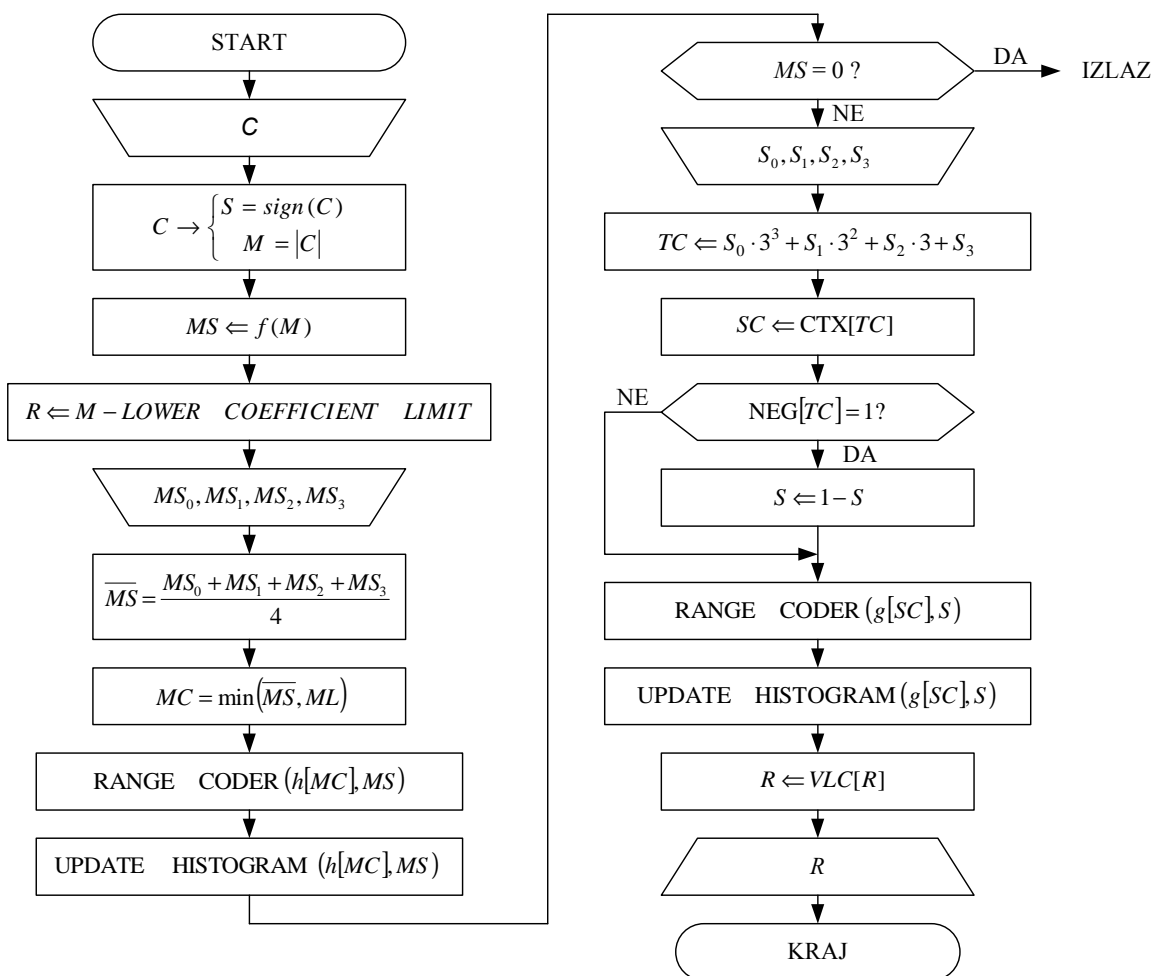
JPEG2000 koristi MQ aritmetički koder sličan QM koderu usvojenom u originalnom JPEG standardu objašnjenom u [95]-[97].

Postupak objavljen u [55], koji je osnova hardverske realizacije u ovom radu, zasniva se na koderu opsega obelodanjenom u [98].

## Enkoderski procenjivač verovatnoća

**Slika 4.1** je dijagram toka enkoderskog procenjivača verovatnoće i entropijskog enkodera iz [55], zasnovanom na jednoprolaznim adaptivnim histogramima. Adaptacija započinje od uniformne raspodele i zahteva nekoliko odbiraka da bi se završila. Vreme

adaptacije je proporcionalno broju odeljaka histograma i razlici između uniformne raspodele i stvarne raspodele promenljive koja se enkoduje. Pozitivni i negativni transformacioni koeficijenti imaju istu verovatnoću svim u podopsezima, što obezbeđuje mogućnost adaptacije histograma samo prema amplitudi, tako da se upola smanjuje ukupan broj nepoznatih verovatnoća i dvostruko povećava brzina adaptacije.



**Slika 4.1** Dijagram toka enkoderskog procenjivača verovatnoće i entropijskog enkodera [55]

Transformacioni koeficijenti  $C$ , se dele na parove znaka  $S$  i amplitude  $M$ . Ovaj postupak je takođe objašnjen i u [99]. Znak  $S$  uzima tri vrednosti, kao u [25], samo sa različitim kodom.

$$S = \begin{cases} 0, & C > 0 \\ 2, & C = 0 \\ 1, & C < 0 \end{cases}$$

Statistika transformacionih koeficijenata za većinu slika direktno vodi do definisanja granica odeljaka zasnovanih na logaritamskoj progresiji, kao u [18], [100]. Međutim, u [55] se koristi veći broj odeljaka histograma (ukupno 32), od postojećih postupaka. Logaritmi za osnovu 2 definišu indeks amplitudne grupe  $MS$  tako da bude jednak zbiru dvostruke pozicije prvog nenultog bita najveće težine i vrednosti prvog sledećeg bita manje težine u binarnoj predstavi amplitude  $M$ , prema **Tabeli 4.1**, datoj za 16-bitne koeficijente.

Ostatak  $R$  je određen kao razlika između amplitude  $M$  i donje granice koeficijenta, jednake binarnoj vrednosti amplitude  $M$  sa svim anuliranim bitima osim prvog nenultog bita najveće težine i prvog sledećeg bita manje težine u binarnoj predstavi amplitude  $M$ , prema **Tabeli 4.1**. Prema tome, transformacioni koeficijenti se dele u trojstvo znaka  $S$ , indeksa amplitudne grupe  $MS$  i ostatka  $R$ . Pri tome se raspodele gustine verovatnoće  $MS$  i  $R$  mogu aproksimativno tretirati kao uniformne.

Predviđanje za  $MS$  na osnovu lokalne varijanse se koristi za povećanje stepena kompresije. Kako se vrednosti  $MS$  mogu smatrati kao aproksimacija logaritma, logaritam lokalne varijanse (proporcionalan entropiji) se može izračunati kao srednja vrednost  $\overline{MS}$  susednih indeksa  $MS_i$  amplitudne grupe već enkodovanih transformacionih koeficijenata, prikazanih na **Slici 4.2**. Na osnovu  $\overline{MS}$  se definiše kontekst amplitude  $MC$  kao indeks odgovarajućeg adaptivnog histograma amplitude  $h[MC]$ , koji se zatim koristi za stvarno enkodovanje indeksa amplitudne grupe  $MS$  korišćenjem enkodera opsega. Međutim, lokalna varijansa može enormno porasti blizu oštih ivica, što dovodi do velikog broja histograma i njihove spore adaptacije. Usled toga se  $MC$  ograničava konstantom  $ML$ , sa preferencijalnom vrednošću  $ML = 4$ . Broj  $MH$  histograma amplitude, to jest broj različitih  $MC$ , je preferencijalno ograničen na  $MH = 1 + ML = 5$ . Nakon toga se obavlja ažuriranje histograma amplitude  $h[MC]$ .

**Tabela 4.1**  
Definicija indeksa amplitudne grupe prema [55]

Granice koeficijenata (uključne)		Opseg koeficijenta	MS
Niža	Viša		
0	0	1	0
1	1	1	1
2	2	1	2
3	3	1	3
4	5	2	4
6	7	2	5
8	11	4	6
12	15	4	7
16	23	8	8
24	31	8	9
32	47	16	10
48	63	16	11
64	95	32	12
96	127	32	13
128	191	64	14
192	255	64	15
256	383	128	16
384	511	128	17
512	767	256	18
768	1023	256	19
1024	1535	512	20
1536	2047	512	21
2048	3071	1024	22
3072	4095	1024	23
4096	6143	2048	24
6144	8191	2048	25
8192	12287	4096	26
12288	16383	4096	27
16384	24575	8192	28
24576	32767	8192	29
32768	49151	16384	30
49152	65535	16384	31

Nešto bolje predviđanje je moguće postići u slučaju više prolaza, izračunavanjem srednje vrednosti  $\overline{MS}$  na osnovu svih 8 suseda iz modifikovanog ulaznog bloka.

$$\overline{MS} = \frac{1}{8} \cdot \sum_{i=0}^7 MS_i$$

MS <sub>0</sub>	MS <sub>1</sub>	MS <sub>2</sub>
MS <sub>3</sub>	MS	

**Slika 4.2** Indeksi amplitudnih grupa već enkodovanih transformacionih koeficijenata

Takođe je moguće postići nešto bolje predviđanje u slučaju više prolaza, zasnovanom na vrednosti  $MSP$  “roditeljskog” transformacionog koeficijenta, tj. onog koji leži u istom relativnom prostornom položaju, ali u višem nivou 2DST.  $MC$  je ograničen konstantom  $MLP$ , sa preferencijalnom vrednošću  $MLP = 3$ . Broj  $MH$  histograma amplitude, to jest broj različitih  $MC$ , je preferencijalno ograničen na  $MH = 1 + ML + MP \cdot MLP = 20$ , pomoću konstante  $MP$ , sa preferencijalnom vrednošću  $MP = 5$ . Međutim, ovako dobijeno povećanje stepena kompresije je neznatno u odnosu na enormno povećanje memorijskog kapaciteta, zbog pamćenja svih transformacionih koeficijenata.

$$MC = \min(\overline{MS}, ML) + MP \cdot \min(MSP, MLP)$$

Višeprolazni postupak sa različitom konstantom  $MLP$  je objašnjen u [99].

U slučaju da je  $MS = 0$ , znak  $S$  se uopšte ne enkoduje. U suprotnom, predviđanje za znak  $S$  se koristi za poboljšanje stepena kompresije. Susedne vrednosti znakova  $S_i$  već enkodovanih transformacionih koeficijenata, prikazane na **Slici 4.3**, koriste se za enkodovanje ternarnog konteksta  $TC$ . Broj različitih vrednosti ternarnog konteksta  $TC$  je  $3^4 = 81$  u jednoprolaznom slučaju sa 4 susedna transformaciona koeficijenta, ili  $3^8 = 6561$  u višeprolaznom slučaju sa 8 suseda.



$S_0$	$S_1$	$S_2$
$S_3$	$S$	

**Slika 4.3** Znakovi već enkodovanih transformacionih koeficijenata

Na osnovu  $TC$  se definiše kontekst znaka  $SC$  kao indeks odgovarajućeg adaptivnog histograma znaka  $g[SC]$ , koji se zatim koristi za stvarno enkodovanje znaka  $S$  korišćenjem enkodera opsega. U oba navedena slučaja, veliki broj različitih vrednosti konteksta znaka  $SC$  dovodi do histograma koji se uopšte ne adaptiraju. Usled toga, CTX tabela prevodi 81 ili 6561 različitih vrednosti ternarnog konteksta  $TC$  u preferencijalni broj od 5 različitih vrednosti konteksta znaka  $SC$  za svaki od podopsega (**Tabela 4.3**), što istovremeno predstavlja broj  $SH$  histograma znaka.

Ovako neverovatno mali broj je opravdan enkodovanjem verovatnijeg znaka  $S$ , što je obezbeđeno ispitivanjem, i inverzijom znaka  $S$  pomoću NEG tabele (**Tabela 4.3**). Ternarni konteksti  $TC$  sa  $NS = NEG[TC] = 0$  su odgovarajući većoj verovatnoći pozitivnog znaka  $P(0)$  od verovatnoće negativnog znaka  $P(1)$ . Ternarni konteksti  $TC$  sa  $NS = NEG[TC] = 1$  su odgovarajući većoj verovatnoći negativnog znaka  $P(1)$  od verovatnoće pozitivnog znaka  $P(0)$ . Konačno se obavlja ažuriranje histograma znaka  $g[SC]$ .

Na kraju se ostatak  $R$  enkoduje korišćenjem koda promenljive dužine (VLC). Broj bita za VLC je odgovarajući opsezima koeficijenata iz **Tabele 4.1**.

Kapacitet zahtevane memorije za smeštanje informacije o kontekstu za sve LH, HL i HH podopsege u toku operacija enkodovanja prikazanog na **Slici 4.1** je približno tri ekvivalentne linije slike, prema **Tabeli 4.4**.

**Tabela 4.2**  
CTX tabela

$S_0$	$S_1$	$S_2$	$S_3$	$TC$	$SC$	$S_0$	$S_1$	$S_2$	$S_3$	$TC$	$SC$	$S_0$	$S_1$	$S_2$	$S_3$	$TC$	$SC$
0	0	0	0	0	0	1	0	0	0	27	2	2	0	0	0	54	1
0	0	0	1	1	0	1	0	0	1	28	1	2	0	0	1	55	0
0	0	0	2	2	0	1	0	0	2	29	2	2	0	0	2	56	0
0	0	1	0	3	4	1	0	1	0	30	0	2	0	1	0	57	1
0	0	1	1	4	1	1	0	1	1	31	1	2	0	1	1	58	0
0	0	1	2	5	2	1	0	1	2	32	1	2	0	1	2	59	2
0	0	2	0	6	1	1	0	2	0	33	1	2	0	2	0	60	0
0	0	2	1	7	0	1	0	2	1	34	0	2	0	2	1	61	0
0	0	2	2	8	1	1	0	2	2	35	0	2	0	2	2	62	0
0	1	0	0	9	0	1	0	0	0	36	1	2	0	0	0	63	0
0	1	0	1	10	1	1	1	0	1	37	1	2	1	0	1	64	0
0	1	0	2	11	1	1	1	0	2	38	1	2	1	0	2	65	0
0	1	1	0	12	1	1	1	1	0	39	0	2	1	1	0	66	0
0	1	1	1	13	0	1	1	1	1	40	1	2	1	1	1	67	1
0	1	1	2	14	0	1	1	1	2	41	0	2	1	1	2	68	1
0	1	2	0	15	0	1	1	2	0	42	1	2	1	2	0	69	0
0	1	2	1	16	4	1	1	2	1	43	0	2	1	2	1	70	0
0	1	2	2	17	0	1	1	2	2	44	0	2	1	2	2	71	3
0	2	0	0	18	4	1	2	0	0	45	3	2	2	0	0	72	0
0	2	0	1	19	2	1	2	0	1	46	0	2	2	0	1	73	4
0	2	0	2	20	2	1	2	0	2	47	1	2	2	0	2	74	3
0	2	1	0	21	1	1	2	1	0	48	1	2	2	1	0	75	2
0	2	1	1	22	0	1	2	1	1	49	2	2	2	1	1	76	2
0	2	1	2	23	3	1	2	1	2	50	2	2	2	1	2	77	0
0	2	2	0	24	0	1	2	2	0	51	0	2	2	2	0	78	2
0	2	2	1	25	0	1	2	2	1	52	0	2	2	2	1	79	1
0	2	2	2	26	1	1	2	2	2	53	1	2	2	2	2	80	0

Podopseg  $LL_{N-1}$  na najvišem  $N-1$  nivou 2DST može biti direktno smešten bez enkodovanja ili se enkodovati korišćenjem bilo koje metode kompresije za male slike. U ovoj realizaciji,  $LL_{N-1}$  se enkoduje korišćenjem samo jednog konteksta, i svi transformacioni koeficijenti se enkoduju korišćenjem iste procedure prikazane na **Slici 4.1**, koja se koristi za enkodovanje transformacionih koeficijenata u svim ostalim podopsezima.

**Tabela 4.3**  
NEG tabela

<i>TC</i>	<i>P</i> (0)	<i>P</i> (1)	<i>NS</i>	<i>TC</i>	<i>P</i> (0)	<i>P</i> (1)	<i>NS</i>	<i>TC</i>	<i>P</i> (0)	<i>P</i> (1)	<i>NS</i>
0	0.5276	0.4724	0	27	0.6147	0.3853	0	54	0.4168	0.5832	1
1	0.5333	0.4667	0	28	0.4170	0.5830	1	55	0.5012	0.4988	0
2	0.4901	0.5099	1	29	0.6326	0.3674	0	56	0.5302	0.4698	0
3	0.2961	0.7039	1	30	0.4889	0.5111	1	57	0.5467	0.4533	0
4	0.4321	0.5679	1	31	0.4176	0.5824	1	58	0.5061	0.4939	0
5	0.6300	0.3700	0	32	0.4469	0.5531	1	59	0.4039	0.5961	1
6	0.4463	0.5537	1	33	0.5505	0.4495	0	60	0.5024	0.4976	0
7	0.4754	0.5246	1	34	0.5240	0.4760	0	61	0.4613	0.5387	1
8	0.4397	0.5603	1	35	0.4731	0.5269	1	62	0.4837	0.5163	1
9	0.5012	0.4988	0	36	0.4299	0.5701	1	63	0.5106	0.4894	0
10	0.5796	0.4204	0	37	0.5880	0.4120	0	64	0.5440	0.4560	0
11	0.4117	0.5883	1	38	0.5806	0.4194	0	65	0.5343	0.4657	0
12	0.5842	0.4158	0	39	0.4698	0.5302	1	66	0.4918	0.5082	1
13	0.5457	0.4543	0	40	0.4119	0.5881	1	67	0.4521	0.5479	1
14	0.5364	0.4636	0	41	0.5193	0.4807	0	68	0.5841	0.4159	0
15	0.5243	0.4757	0	42	0.4539	0.5461	1	69	0.5211	0.4789	0
16	0.7224	0.2776	0	43	0.4871	0.5129	1	70	0.4783	0.5217	1
17	0.5050	0.4950	0	44	0.4953	0.5047	1	71	0.6651	0.3349	0
18	0.7235	0.2765	0	45	0.3502	0.6498	1	72	0.4561	0.5439	1
19	0.3963	0.6037	1	46	0.4688	0.5312	1	73	0.6998	0.3002	0
20	0.6019	0.3981	0	47	0.5802	0.4198	0	74	0.6531	0.3469	0
21	0.4508	0.5492	1	48	0.4432	0.5568	1	75	0.6163	0.3837	0
22	0.5286	0.4714	0	49	0.3927	0.6073	1	76	0.5956	0.4044	0
23	0.6598	0.3402	0	50	0.6199	0.3801	0	77	0.5022	0.4978	0
24	0.4770	0.5230	1	51	0.5357	0.4643	0	78	0.6148	0.3852	0
25	0.5417	0.4583	0	52	0.4830	0.5170	1	79	0.4368	0.5632	1
26	0.4398	0.5602	1	53	0.4464	0.5536	1	80	0.5065	0.4935	0

Na **Slici 4.4** je prikazan dijagram toka rada enkodera na početku enkodovanja bloka od četiri linije. Ova procedura nije opisana u [55], koji je osnova hardverske realizacije. Kao kompromis između potrebne memorije u blok memoriji koja se nalazi u bloku kvantizatora sa baferovanjem i stepena kompresije, izabrano je da se maksimum računa na četiri linije, kako je i implementirano u bloku kvantizatora sa baferovanjem.

**Tabela 4.4**

Kapacitet zahtevane memorije za smeštanje informacije o kontekstu

2DST nivo	Ukupan kapacitet memorije za kontekste 3 podopsega (LH, HL i HH) na jednom nivou [koeficijent]
0	$\frac{3W}{2}$
1	$\frac{3W}{4}$
2	$\frac{3W}{8}$
$n$	$\frac{3W}{2^{n+1}}$
$N-1$	$\frac{3W}{2^N}$
Svi nivoi i svi podopsezi	$3 \cdot \sum_{n=0}^{N-1} \frac{W}{2^n} = 3 \cdot W \cdot \left(1 - \frac{1}{2^N}\right) < 3 \cdot W$

Maksimum na četiri linije jednog podopsega predstavlja se svojom amplitudskom grupom, prema **Tabeli 4.1**. Ukoliko je maksimum na četiri linije nula vrši se VLC enkodovanje nule (jednim bitom) i procedura je završena. Ukoliko maksimum na četiri linije nije nula vrši se: a) VLC ekodovanje jedinice (jednim bitom) ukoliko je maksimum na četiri linije veći od svih prethodnih maksimuma na istom podopsegu, inicijalizacija histograma za dati podopseg sa amplitudskom grupom maksimuma  $i_{max}$  kao ulaznim parametrom, i konačno VLC enkodovanje amplitudne grupe maksimuma sa 5 bita, ili b) VLC enkodovanje nule (jednim bitom) ukoliko je maksimum na četiri linije manji ili jednak svim prethodnim maksimumima na istom podopsegu.

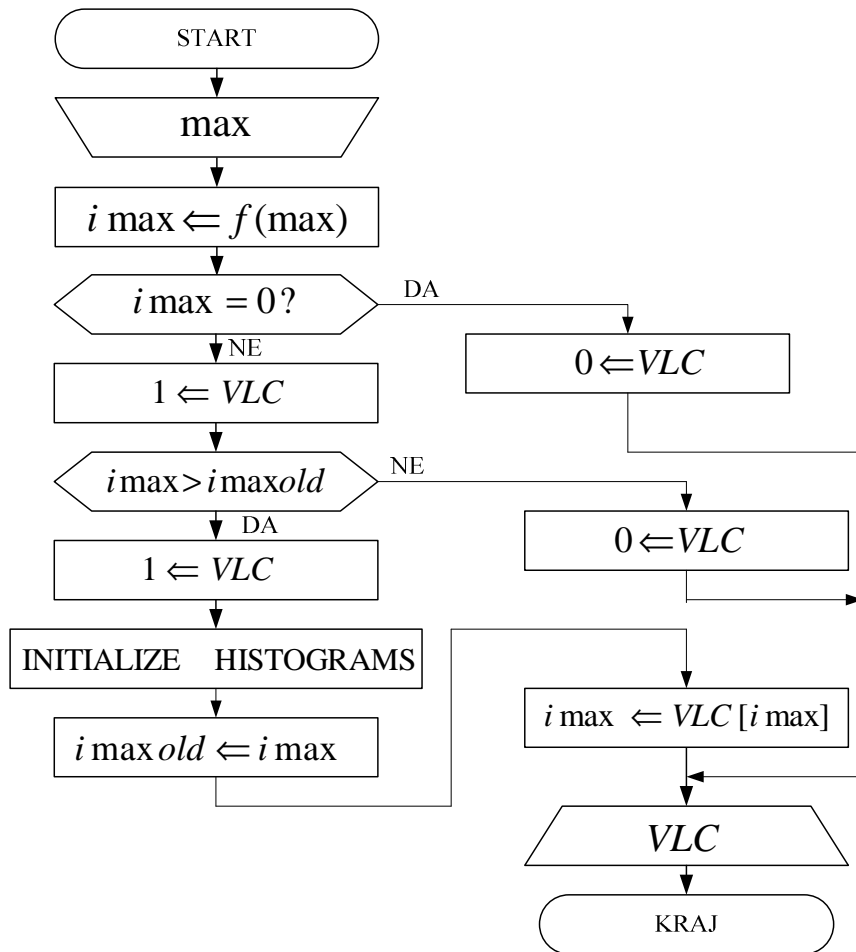
**Slika 4.5** je dijagram toka inicijalizacije histograma sa brzom adaptacijom. Svaki odeljak histograma odgovara jednom simbolu  $x$ , koji može biti  $MS$  za histogram amplitude ili  $S$  za histogram znaka. Jednostavan postupak procene verovatnoće  $p(x)$  pojavljivanja simbola  $x$  prema postojećem stanju tehnike je zasnovan na broju pojavljivanja  $u(x)$  simbola  $x$  i broju  $Total$  pojavljivanja svih simbola. Takođe se definiše kumulativna verovatnoća  $P(x)$  svih simbola  $y$  koji u alfabetu prethode simbolu  $x$ .

$$p(x) = \frac{u(x)}{Total}$$

$$Total = \sum_x u(x)$$

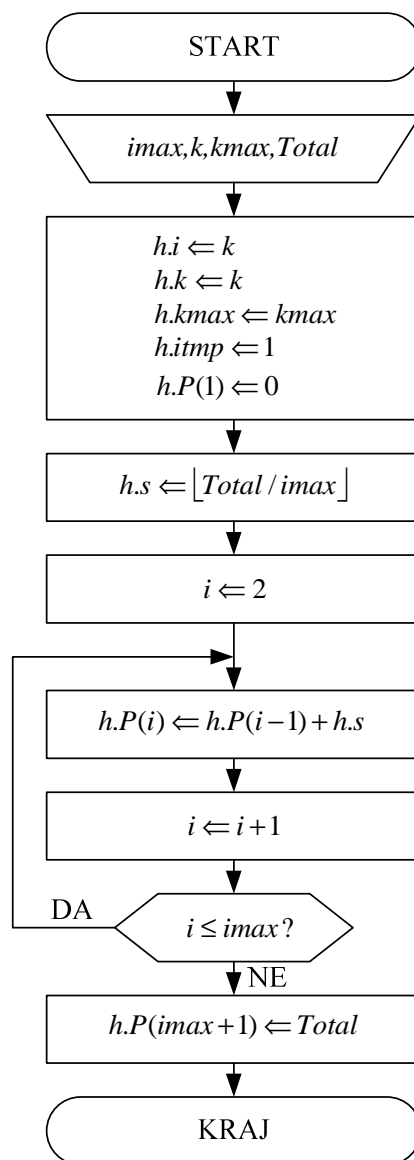
$$P(x) = \sum_{y < x} p(y) = \frac{U(x)}{Total}$$

$$U(x) = \sum_{y < x} u(y)$$



Slika 4.4 Dijagram toka rada entropijskog enkodera na početku bloka od četiri linije

Osnovni nedostatak ovog inače jednostavnog postupka je što je  $Total$  proizvoljan ceo broj, zbog čega je potrebna operacija deljenja za određivanje verovatnoće  $p(x)$ . Međutim, operacija deljenja u [55] je zamenjena pomeranjem udesno za  $w_3$  bita, pod pretpostavkom da je  $Total = 2^{w_3}$ . Drugi nedostatak ovog postupka je spora adaptacija verovatnoće  $p(x)$  usled usrednjavanja.



Slika 4.5 Dijagram toka inicijalizacije histograma

Adaptacija verovatnoće  $p(x)$  prema [55] se obezbeđuje niskopropusnim filtriranjem binarne sekvence  $I(j)$  koja predstavlja pojavljivanje simbola  $x$  u sekvenci  $y$

$$I(j) = \begin{cases} 1, & y(j) = x \\ 0, & y(j) \neq x \end{cases}$$

Vremenski odziv filtra propusnika niskih učestanosti je izuzetno važan iz dva razloga: veća vremenska konstanta omogućava preciznije predviđanje u ustaljenom stanju, dok manja vremenska konstanta omogućava brže predviđanje. Ovaj problem je posebno

izražen na početku adaptacije usled nedostatka informacija. Umesto pravljenja kompromisa pri izboru fiksnog dominantnog pola ostvarena je promenljivost dominantnog pola između minimalne i maksimalne vrednosti.

Na **Slici 4.5**:

- $i$  je indeks odeljka histograma, sa vrednostima od 1 do  $imax$
- $imax$  je maksimalni indeks  $i$  nepraznog odeljka histograma, drugim rečima,  $imax$  je ukupan broj različitih simbola u alfabetu,  $i$  manji je ili jednak 32 za histogram amplitude ili jednak 2 za histogram znaka.
- $h.P[]$  je niz kumulativnih verovatnoća  $h.P[i] = P(y|y < i) = \sum_{y < i} p(y)$ .
- $h.k$  je recipročna vrednost apsolutnog dominantnog pola filtra propusnika niskih učestanosti. Njegova promenljivost između  $h.kmax$  i  $h.kmin$  obezbeđuje brzu adaptaciju histograma posle početka rada.
- $h.kmax$  je recipročna vrednost minimalnog apsolutnog dominantog pola filtra propusnika niskih učestanosti,  $i$  predstavlja fiksni empirijski parametar manji od  $Total$ .
- $h.kmin$  je recipročna vrednost maksimalnog apsolutnog dominantog pola filtra propusnika niskih učestanosti,  $i$  predstavlja fiksni empirijski parametar sa preferencijalnom vrednošću  $h.kmin = 2$ .
- $h.i$  je ukupan broj simbola unutar histograma plus 1.
- $h.itmp$  je privremeni  $h.i$  pre promene  $h.k$ .
- $h.s$  je veličina koraka.

**Slika 4.6** je dijagram toka algoritma za ažuriranje histograma sa brzom adaptacijom, zasnovanog na ulazu simbola  $x$  i već objašnjenjnoj strukturi histograma  $h$ . Međutim, kako enkoder opsega ne može da radi sa predviđenom nultom verovatnoćom  $p(x) = 0$ , čak ni za simbole koji se ne pojavljuju, potrebno je modifikovati binarnu sekvencu  $I(j)$ . Prema [55] se ustvari predviđa modifikovana verovatnoća  $Mp(x) = Total \cdot p(x)$  primenom aritmetike u fiksnom zarezu, što je dodatni razlog za modifikaciju binarne

sekvence  $I(j)$ . Dakle, adaptacija verovatnoće  $p(x)$  se obavlja niskopropusnim filtriranjem modifikovane binarne sekvence  $MI(j)$ .

$$MI(j) = \begin{cases} Total - imax, & y(j) = x \\ 1, & y(j) \neq x \end{cases}$$

Maksimalna verovatnoća  $\max p(x)$  i minimalna verovatnoća  $\min p(x)$  postaju:

$$\max p(x) = \frac{Total - imax}{Total} < 1$$

$$\min p(x) = \frac{1}{Total} > 0$$

Preferirani filter propusnik niskih učestanosti je IIR filter prvog reda, prema donjoj definiciji, gde se operacija deljenja izbegava očuvavanjem  $h.k$  u obliku stepena dvojke u toku promene.

$$Mp(x) \Leftarrow Mp(x) \cdot \left(1 - \frac{1}{h.k}\right) + MI(j)$$

Umesto modifikovane verovatnoće  $Mp(x)$ , ažurira se modifikovana kumulativna verovatnoća  $MP(x) = Total \cdot P(x)$ , tj. niz kumulativnih verovatnoća  $h.P[i]$ . Konstanta  $K_h$  i indeks  $i$  odeljka histograma se inicijalizuju na početku. Dodavanje  $i-1$  kumulativnoj verovatnoći  $h.P[i]$  preskaliranoj sa  $K_h$  je ekvivalentno dodavanju jedinice broju  $u(x)$ . Ažuriranje kumulativne verovatnoće  $h.P[i]$  se obavlja samo za histograme sa  $i$  većim ili jednakim  $x$ , čija se verovatnoća procenjuje.

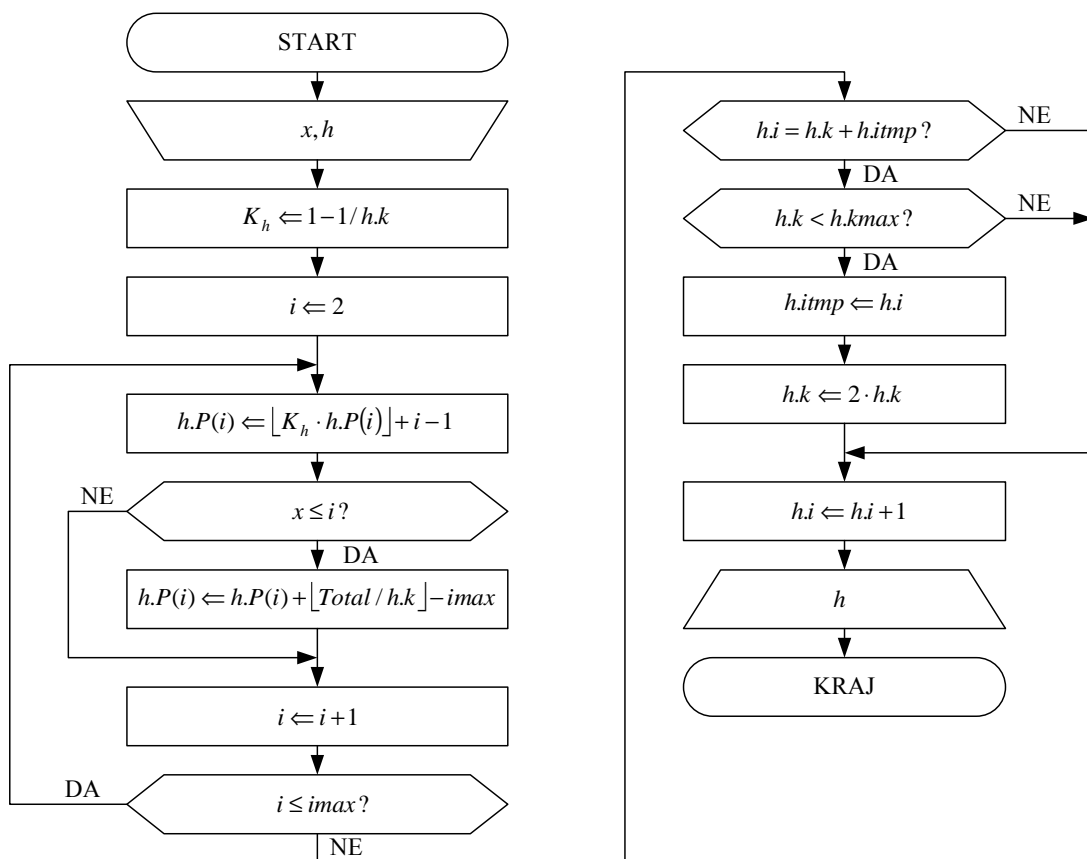
Drugi deo algoritma za ažuriranje histograma omogućava brzu adaptaciju histograma. Matematički ekvivalent je

$$h.k = \min\left[2^{\lceil \log_2(h.i + h.kmin - 2) \rceil}, h.kmax\right]$$

$$h.k = \max(h.k, h.kmin),$$

gde je preferencijalno  $h.kmin = 2$ , što je važno za prvi  $h.k$  u toku procesa brze adaptacije.





Slika 4.6 Dijagram toka ažuriranja histograma

## Entropijski enkoder

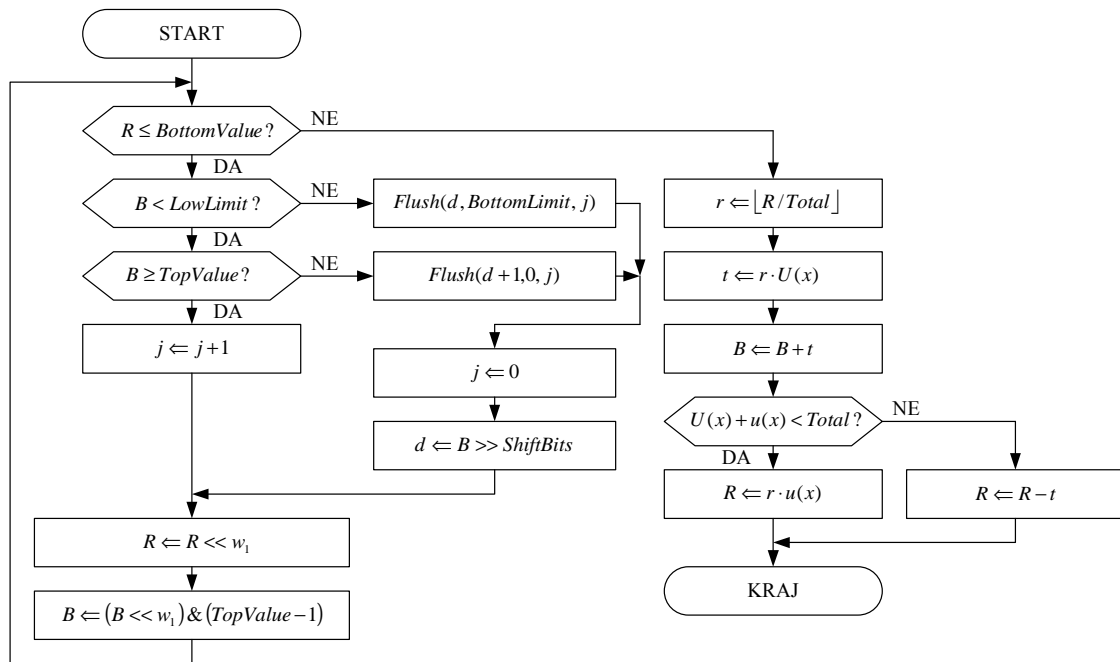
Dijagram toka enkodera opsega [98], [101], [102] prikazan je na **Slici 4.7**. Simbol  $x$  će biti enkodovan u baferu širine  $s = b^w$  u vidu broja  $i$ :

$$i \in (\lfloor s \cdot P(x) \rfloor, \lfloor s \cdot (P(x) + p(x)) \rfloor)$$

$$\lfloor s \cdot P(x) \rfloor \leq i < \lfloor s \cdot (P(x) + p(x)) \rfloor$$

$$s \cdot P(x) < i + 1 \leq s \cdot (P(x) + p(x))$$

$$P(x) < \frac{i+1}{s} \leq P(x) + p(x).$$



Slika 4.7 Dijagram toka enkodera opsega

Stanje kodera se predstavlja promenljivim  $(d, j), [B, B + R)$ :

$B$  = donja granica opsega;

$R$  = opseg  $R = T - B$ , umesto  $T$  = gornja granica opsega;

$d$  = izlazni bajt;  $i$

$j$  = broj bajtova podkoračenja.

Algoritam enkodera opsega u pokretnom zarezu posle renormalizacije i bez provere graničnih uslova je:

$$t \leftarrow R \cdot P(x);$$

$$B \leftarrow B + t;$$

$$R \leftarrow R \cdot p(x).$$

Posle uvođenja preskaliranog opsega  $r$ , algoritam enkodera opsega u celobrojnoj aritmetici posle renormalizacije i bez provere graničnih uslova postaje:

$$r \leftarrow \left\lfloor \frac{R}{Total} \right\rfloor;$$

$$t \leftarrow r \cdot U(x);$$

$$B \leftarrow B + t;$$
$$R \leftarrow r \cdot u(x).$$

Algoritam za enkoder opsega prema postojećem stanju tehnike se najlakše objašnjava pomoću aritmetičkih operatora iz C/C++ jezika, kao što su:

$$x \ll y = \text{pomeraj } x \text{ ulevo za } y \text{ bita};$$

$$x \gg y = \text{pomeraj } x \text{ udesno za } y \text{ bita};$$

$$x \% y = \text{ostatak od } x/y;$$

$$x|y = x \text{ ili } y; \text{ i}$$

$$x \& y = x \text{ i } y.$$

Konstante *TopValue*, *BottomValue*, *ShiftBits*, *ExtraBits*, *BottomLimit* i *LowLimit* sa navedenom poželjnim vrednostima su definisane zavisno od dve konstante  $w_1$  i  $w_2$ , sa poželjnim vrednostima 8 i 32, respektivno.

$$TopValue = 1 \ll (w_2 - 1) = 40000000h$$

$$BottomValue = TopValue \gg w_1 = 00400000h$$

$$ShiftBits = w_2 - w_1 - 1 = 23$$

$$ExtraBits = (w_2 - 2) \% w_1 + 1 = 4$$

$$BottomLimit = (1 \ll w_1) - 1 = 0FFh$$

$$LowLimit = BottomLimit \ll ShiftBits$$

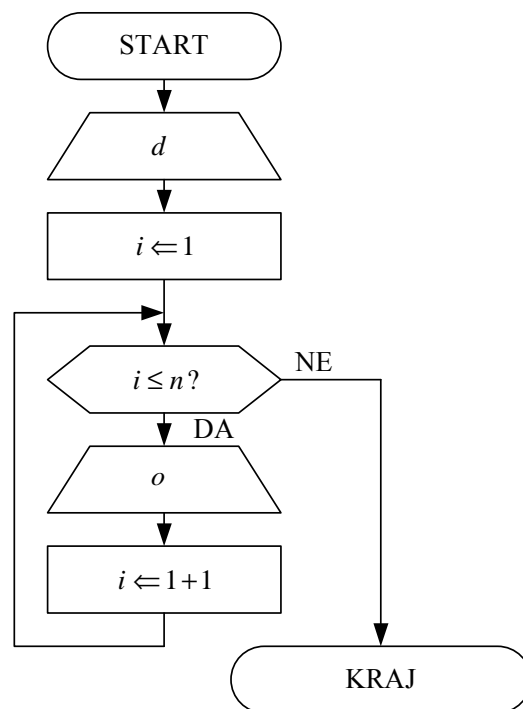
Osnovna ideja upotrebe *ExtraBits* je da se  $B$ ,  $T$  i  $s$  smatraju za vrednosti u nepokretnom zarezu, sa *ExtraBits* bita posle decimalne tačke. Ova modifikacija je prvenstveno urađena zbog dekodera, što izlazi iz opsega ovoga rada. Pre nego što se započne sa enkodovanjem opsega, sledeće promenljive moraju biti inicijalizovane:

$$B = 0$$

$$R = TopValue$$

$$d = 0$$

$$j = 0$$



**Slika 4.8** Dijagram toka *Flush procedure*

**Slika 4.8** predstavlja dijagram toka *flush procedure* prema postojećem stanju tehnike, koja izbacuje bajt  $d$ , inicijalizuje brojač petlje  $i$ , unutar petlje testira  $i$ , izbacuje  $j$  bajtova  $o$ , i inkrementira  $i$ .

Prvi deo algoritma za enkodovanje opsega prikazan na **Slici 4.7** obavlja renormalizaciju pre enkodovanja. U slučaju opsega sa mogućim prenosom (kada je  $B \geq LowLimit$ ) prva *flush procedure* definisana na **Slici 4.8** izbacuje bajt  $d$  i sve bajtove podkoračenja  $0FFh$ . U slučaju opsega sa stvarnim prenosom (kada je  $B < TopValue$ ) druga *flush procedure* izbacuje bajt  $d + 1$  i sve bajtove podkoračenja  $0h$ , jer  $0FFh$  postaje  $0h$  usled prenosa. U oba slučaja posle *flush procedure*, broj bajtova podkoračenja  $j$  se inicijalizuje na nulu i generiše se izlazni bajt  $d$ , dodelom višeg bajta  $B$ .  $R$  se ažurira pomeranjem bajta koji je već bio izlaz ili će biti izlaz u slučaju  $0FFh$ .  $B$  se ažurira pomeranjem i brisanjem bita prenosa. Ukoliko nije u pitanju opseg sa ili mogućim ili stvarnim prenosom,  $j$  se inkrementira.

Drugi deo algoritma za enkodovanje opsega prikazan na **Slici 4.7** ažurira opseg. Preskalirani opseg  $r$  za sve simbole se ažurira korišćenjem prve operacije deljenja. Opseg  $t$  tekućeg simbola je izveden pomoću prve operacije množenja sa  $U(x)$  za tekući simbol  $x$ .  $B$  se takođe ažurira.  $R$  se ažurira pomoću druge operacije množenja sa  $u(x)$  za tekući simbol  $x$ , za sve simbole osim poslednjeg. U slučaju poslednjeg simbola,  $R$  se ažurira pomoću operacije oduzimanja.

Broj iteracija u ovoj petlji je u najgorem slučaju  $\log_b(s)$  i smanjuje se izborom većeg  $b$ . Ovo je osnovna prednost koder opsega prema postojećem stanju tehnike u odnosu na aritmetičke kodere prema postojećem stanju tehnike. Za  $s = 2^{31}$ , aritmetički koder koristi  $b = 2$ , dok koder opsega koristi  $b = 2^8 = 256$ . Dakle, u najgorem slučaju, aritmetičkom koderu trebaju 32 iteracije za svaki simbol, dok koder opsega zahteva samo 4 iteracije. Pored toga, prvi setovan bit opsega ne mora biti na fiksiranoj poziciji. Binarni aritmetički koderi prema postojećem stanju tehnike češće zahtevaju renormalizaciju opsega u poređenju sa koderom opsega prema postojećem stanju tehnike, čime se brzina izvršavanja skoro dvostruko smanjuje. Ova činjenica je ilustrovana u **Tabeli 4.5** za koder opsega i **Tabeli 4.6** za aritmetički koder, koji prikazuju enkodovanje istog ulaznog niza 00001123000, sa neuslovnim verovatnoćama simbola  $P(0) = 50/100$ ,  $P(1) = 25/100$ ,  $P(2) = 15/100$  i  $P(3) = 10/100$ . Međutim, usled kašnjenja koder opsega, do 2 bita viška se emituju kasnije u slučaju aritmetičkog koder opsega, i do 8 bita viška se emituju kasnije u slučaju koder opsega.

U enkoderu opsega prema postojećem stanju tehnike moguće je realizovati prvu operaciju deljenja sa  $Total$  pomoću pomeraja udesno za  $w_3$  bita u slučaju kada je  $Total = 2^{w_3}$ , što je obezbeđeno pomoću enkoderskog procenjivača verovatnoće. Dodatno, postoje dve operacije množenja u enkoderu po svakom simbolu komprimovane slike, što smanjuje brzinu obrade. Ovi nedostaci su otklonjeni u enkoderu opsega prema [55].

**Slika 4.9 a)** i **Slika 4.9 b)** su dijagrami toka enkodera opsega prema [55], bez operacija deljenja i množenja. Prva operacija deljenja sa  $Total = 2^{w_3}$  realizuje se pomoću

pomeranja udesno za  $w_3$  bita, zahvaljujući brznoj adaptaciji histograma. Preskalirani opseg se predstavlja kao  $r = V \cdot 2^l$ . Prva operacija množenja sa **Slike 4.7** se realizuje pomoću prve operacije množenja sa malim brojem  $V$  i pomeranjem ulevo za  $l$  bita na **Slici 4.9 b**). Druga operacija množenja sa **Slike 4.7** se realizuje pomoću druge operacije množenja sa malim brojem  $V$  i pomeranjem ulevo za  $l$  bita na **Slici 4.9 b**). Pored toga, operacija množenja sa malim neparnim brojem  $V = 3$  ili  $V = 5$  može biti realizovana pomoću kombinacije operacija pomeranja i sabiranja, čime se u potpunosti uklanjaju operacije množenja.

**Tabela 4.5**  
Enkodovanje ulaznog niza 00001123000 koderom opsega

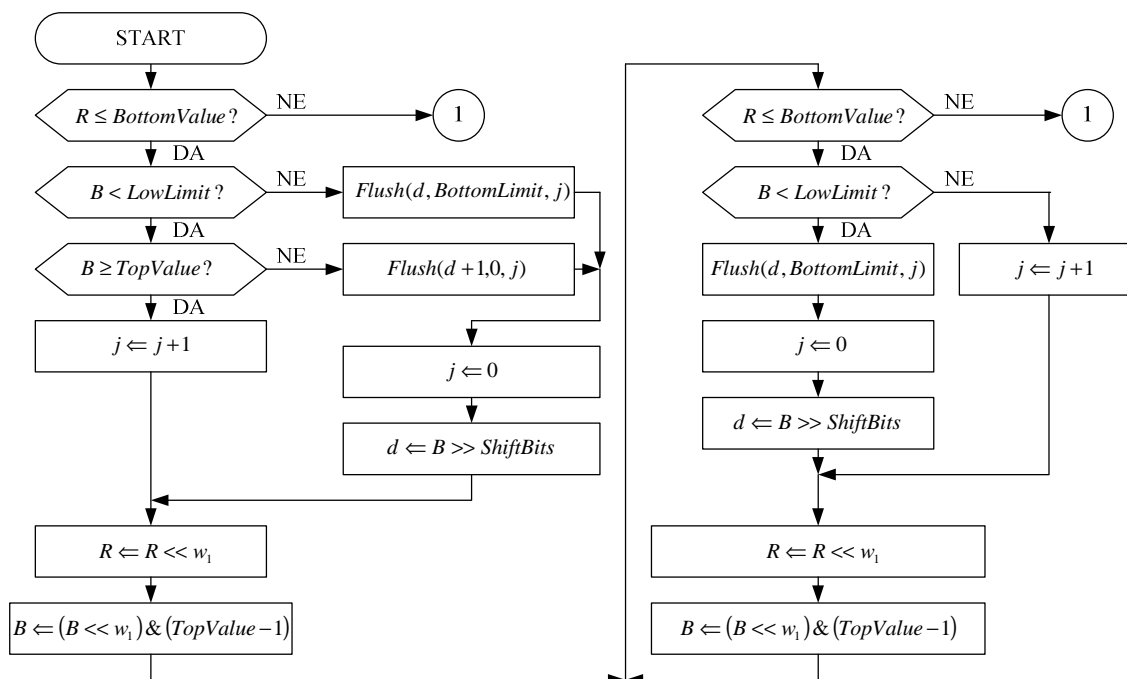
KODER OPSEGA				
Događaj	$B$	$T$	$R$	Izlaz (binarni)
Početak	0	2147483648	2147483648	
Simbol (0)	0	1073741800	1073741800	
Simbol (0)	0	536870900	536870900	
Simbol (0)	0	268435450	268435450	
Simbol (0)	0	134217700	134217700	
Simbol (1)	67108850	100663275	33554425	
Simbol (1)	83886050	92274650	8388600	
Renormalizacija	2147475968	4294957568	2147481600	00000000 (ignorisan)
Simbol (2)	3758087168	4080209408	322122240	
Simbol (3)	4047997148	4080209368	32212220	
Simbol (0)	4047997148	4064103248	16106100	
Simbol (0)	4047997148	4056050198	8053050	
Renormalizacija	1200151552	3261732352	2061580800	00001010
Simbol (0)	1200151552	2230941952	1030790400	

Prvi deo algoritma za enkoder opsega prikazan na **Slici 4.7**, koji obavlja renormalizaciju pre enkodovanja, može se direktno iskoristiti uz razdvajanje dela obrade sa **Slike 4.7** u dva dela prema **Slici 4.9 a**), čime se direktno povećava brzina obrade, izbacivanjem jednog ispitivanja u drugom delu obrade.

**Tabela 4.6**  
Enkodovanje ulaznog niza 00001123000 aritmetičkim koderom

ARITMETIČKI KODER				
Događaj	$B$	$T$	$R$	Izlaz (binarni)
Početni	0	2147483648	2147483648	
Simbol (0)	0	1073741800	1073741800	
Simbol (0)	0	536870900	536870900	
Renormalizacija	0	1073741800	1073741800	0 (ignorisan)
Simbol (0)	0	536870900	536870900	
Renormalizacija	0	1073741800	1073741800	0
Simbol (0)	0	536870900	536870900	
Renormalizacija	0	1073741800	1073741800	0
Simbol (1)	536870900	805306350	268435450	
Renormalizacija	1073741800	1610612700	536870900	0
Renormalizacija	1073741776	2147483576	1073741800	
Simbol (1)	1610612676	1879048126	268435450	
Renormalizacija	1073741704	1610612604	536870900	10
Renormalizacija	1073741584	2147483384	1073741800	
Simbol (2)	1879047934	2040109204	161061270	
Renormalizacija	1610612220	1932734760	322122540	10
Renormalizacija	1073740792	1717985872	644245080	1
Simbol (3)	1653561292	1717985792	64424500	
Renormalizacija	1159638936	1288487936	128849000	1
Renormalizacija	171794224	429492224	257698000	1
Renormalizacija	343588448	858984448	515396000	0
Renormalizacija	687176896	1717968896	1030792000	0
Simbol (0)	687176896	1202572896	515396000	
Renormalizacija	300611968	1331403968	1030792000	
Simbol (0)	300611968	816007968	515396000	
Renormalizacija	601223936	1632015936	1030792000	01
Simbol (0)	601223936	1116619936	515396000	

Potrebno je primetiti da aproksimacije korišćene u [55] dovode do manjeg stepena kompresije. Na primer, maksimalna moguća greška aproksimacije se postiže fiksiranjem  $V = 1$  i time potpunim uklanjanjem svih operacija deljenja i množenja. U tom slučaju, stepen kompresije se smanjuje za manje od 5%. Ako je dozvoljeno da  $V$  bude 1 ili 3, stepen kompresije se smanjuje za manje od 1%.



Slika 4.9 a) Dijagram toka enkodera opsega bez množenja i deljenja, prvi deo

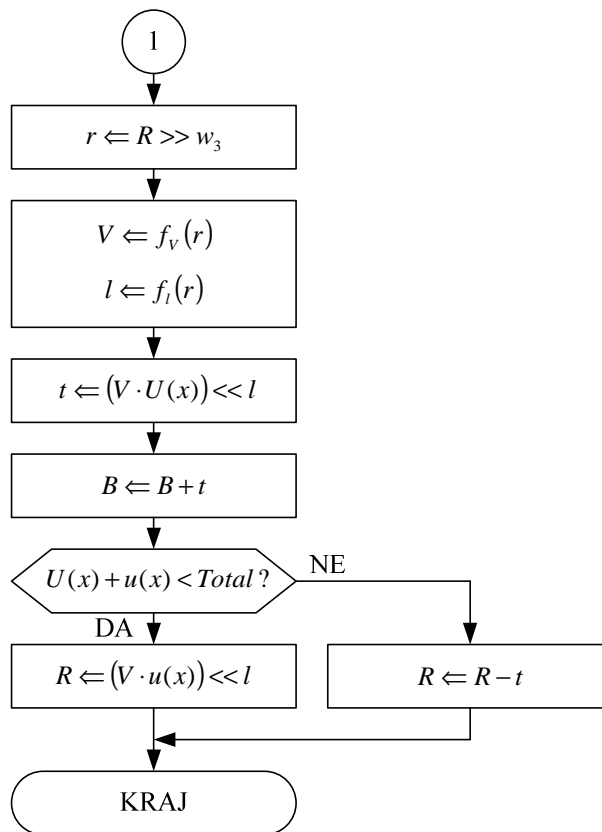
Tabela 4.7 i Tabela 4.8 prikazuju razliku u broju operacija množenja i deljenja po enkodovanom simbolu između koder opsega prema postojećem stanju tehnike i koder opsega prema [55].

Tabela 4.7

Poređenje broja operacija množenja i deljenja između koder opsega prema postojećem stanju tehnike i koder opsega prema [55], kada  $Total \neq 2^{w_3}$

$Total \neq 2^{w_3}$				
OPERACIJA	koder opsega prema postojećem stanju tehnike	koder [55] $r = V \cdot 2^l$		
		$V = 1$	$V = 3$ $V = 5$	$V \geq 7$
MNOŽENJE	2	0	0	2
DELJENJE	1	1	1	1





Slika 4.9 b) Dijagram toka enkodera opsega bez množenja i deljenja, drugi deo

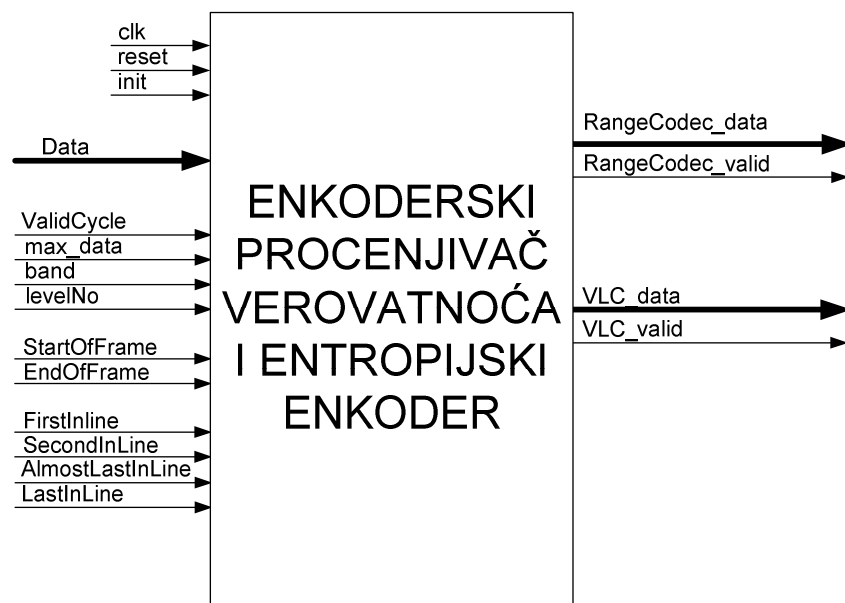
**Tabela 4.8**

Poređenje broja operacija množenja i deljenja između koder opsega prema postojećem stanju tehnike i koder opsega prema [55], kada  $Total = 2^{w_3}$

$Total = 2^{w_3}$				
OPERACIJA	koder opsega prema postojećem stanju tehnike	koder [55] $r = V \cdot 2^l$		
		$V = 1$	$V = 3$ $V = 5$	$V \geq 7$
MNOŽENJE	2	0	0	2
DELJENJE	0	0	0	0

## Hardverska realizacija enkoderskog procenjivača verovatnoća i entropijskog enkodera

Najviši hijerarhijski nivo hardverske realizacije enkoderskog procenjivača verovatnoća i entropijskog enkodera za jednu od YUV ravni boje prikazan je na **Slici 4.10**.



**Slika 4.10** Najviši hijerarhijski nivo hardverske realizacije enkoderskog procenjivača verovatnoća i entropijskog enkodera

Značenja signala sa **Slike 4.10** data su u **Tabeli 4.9**.

**Tabela 4.9**  
Signali na najvišem hijerarhijskom nivou enkoderskog procenjivača verovatnoća i entropijskog enkodera

Ime signala	Širina signala	Smer signala	Funkcija signala
Data	17 bita	ulaz	ulazni podatak u formatu znak + apsolutna vrednost
ValidCycle	1 bit	ulaz	kvalifikator za ulaz
max_data	1 bit	ulaz	kvalifikator da li je ulazni podatak maksimum ili kvantizovani transformacioni koeficijent
band	2 bita	ulaz	podopseg u opsegu nivoa kojem pripada Data
levelNo	4 bita	ulaz	nivo kojem pripada data
StartOfFrame	1 bit	ulaz	aktivan uz prvi maksimum svakog nivoa svake ulazne slike
EndOfFrame	1 bit	ulaz	nezavisan signal koji signalizira da više nema podataka na datom nivou svake ulazne slike
FirstInLine	1 bit	ulaz	signalizira da je Data prvi koeficijent u tekućoj liniji transformacionih koeficijenata na svakom od podopsega posebno
SecondInLine	1 bit	ulaz	signalizira da je Data drugi koeficijent

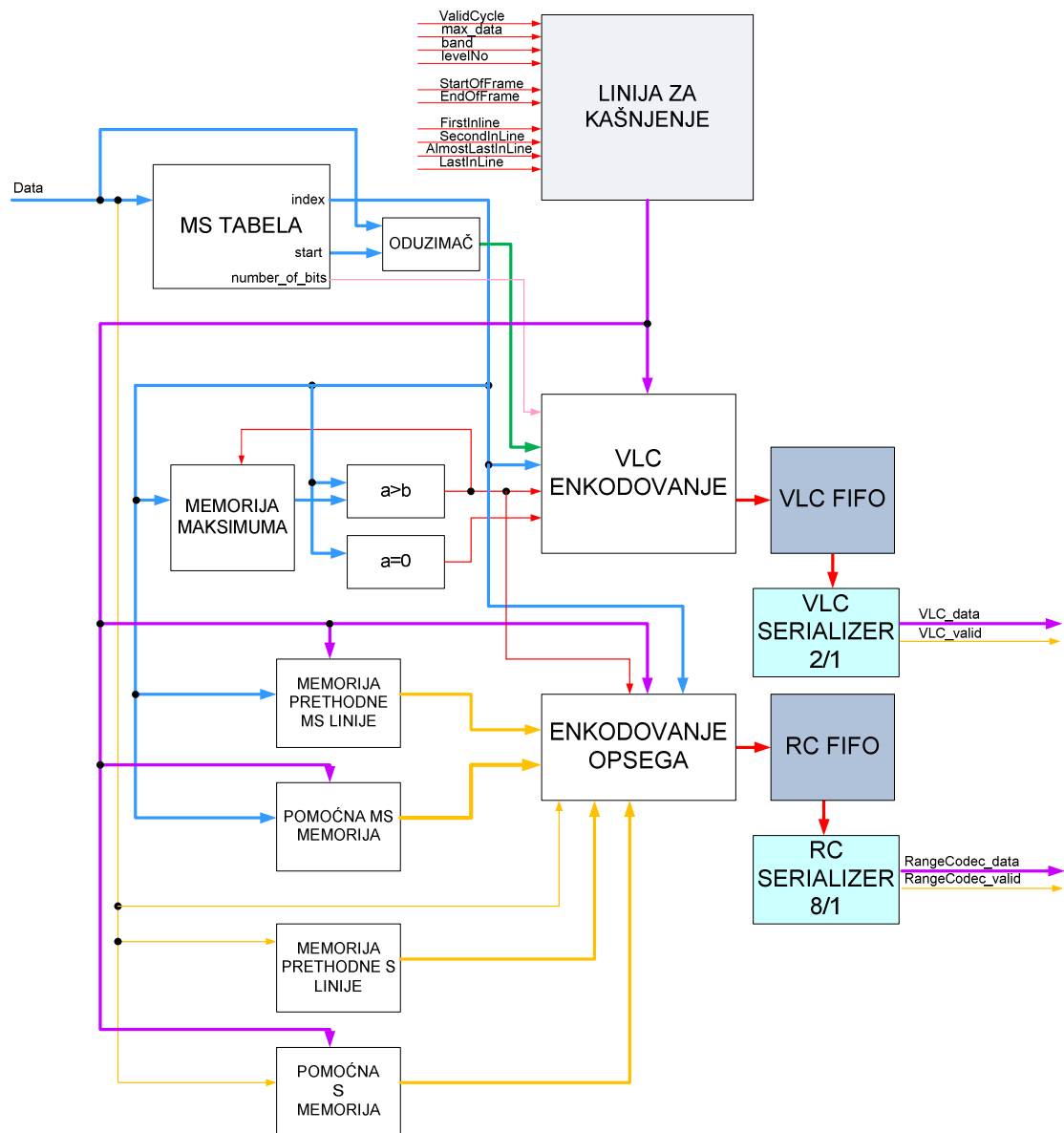
			u tekućoj liniji transformacionih koeficijenata na svakom od podopsega posebno
AlmostLastInLine	1 bit	ulaz	signalizira da je Data pretposlednji koeficijent u tekućoj liniji transformacionih koeficijenata na svakom od podopsega posebno
LastInLine	1 bit	ulaz	signalizira da je Data poslednji koeficijent u tekućoj liniji transformacionih koeficijenata na svakom od podopsega posebno
RangeCodec_data	13 bita	izlaz	niz podataka enkodera opsega
RangeCodec_valid	1 bit	izlaz	kvalifikator za izlaz enkodera opsega
VLC_data	13 bita	izlaz	niz podataka VLC enkodera
VLC_valid	1 bit	izlaz	kvalifikator za izlaz VLC enkodera

Niz podataka oba enkodera sastoji se od 13 bita: 8 bita podatka, 4 bita koji ukazuju kom nivou pripadaju podaci, i 1 bit koji signalizira da je generisanje podataka na datom nivou završeno.

Blok šema realizacije enkoderskog procenjivača verovatnoća i entropijskog enkodera sa **Slike 4.10** prikazana je na **Slici 4.11**.

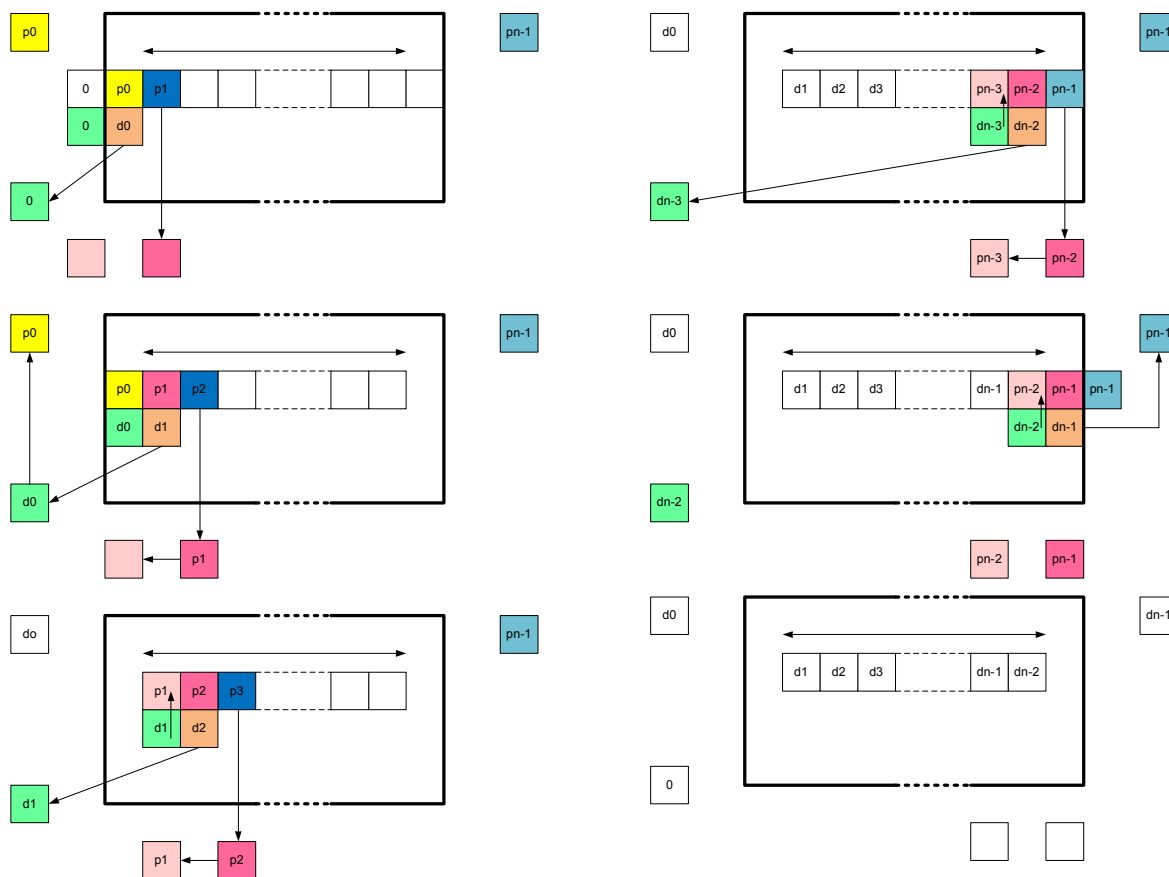
Blok MS TABELA implementira modifikovanu tabelu sa **Slike 4.1**. Izlaz *index* predstavlja broj amplitudske grupe koeficijenta ili maksimuma, u zavisnosti od tipa podatka na ulazu. Izlaz *start* predstavlja donju granicu opsega amplitudske grupe, dok izlaz *number\_of\_bits* predstavlja broj potrebnih bita za predstavljanje ostatka *R*. Broj amplitudne grupe se koristi u VLC enkoderu u ciklusu kada je na ulazu prisutan maksimum na nekom od podopsega, koji se VLC enkoduje u slučaju da je veći od svih prethodnih sa fiksnim brojem od 5 bitova. Komparator koji upoređuje novi maksimum sa nulom koristi se za VLC enkodovanje maksimuma, to jest ukoliko je ovaj maksimum nula u VLC izlazni niz se upisuje samo jedan bit, i to nula. Komparator koji upoređuje novi maksimum sa najvećim prethodnim na istom podopsegu koristi se: a) za VLC enkodovanje maksimuma, ukoliko je novi maksimum manji u VLC izlazni niz se upisuje samo jedan bit, i to nula, a ukoliko je veći upisuje se jedan bit, jedinica, praćen amplitudnom grupom; b) kao uslov za inicijalizaciju histograma u bloku ENKODOVANJE OPSEGA. Broj amplitudne grupe se u ciklusu kada je na ulazu

prisutan transformacioni koeficijent koristi u enkoderu opsega. Oduzimač sa **Slike 4.11** izračunava ostatak  $R$  koji se u slučaju kada je na ulazu prisutan transformacioni koeficijent VLC enkoduje sa  $number\_of\_bits$  bita.



**Slika 4.11** Blok šema enkoderskog procenjivača verovatnoća i entropijskog enkodera

Prethodna vrednost najvećeg maksimuma po svim podopsezima čuva se u u bloku MEMORIJA MAKSIMUMA, zajedno sa brojačem koeficijenta u tekućoj liniji po svim podopsezima, a koji se koristi za računanje adrese prilikom očitavanja iz blokova MEMORIJA PRETHODNE MS LINIJE i MEMORIJA PRETHODNE S LINIJE.



Slika 4.12 Ilustracija računanja konteksta za vreme jedne linije

Za računanje konteksta amplitudne grupe i znaka potrebno je čuvati vrednosti amplitudnih grupa i znaka koeficijenata iz prethodne linije. U tu svrhu koriste se blokovi MEMORIJA PRETHODNE MS LINIJE i MEMORIJA PRETHODNE S LINIJE. Kako su u jednom prolasku potrebne vrednosti tri koeficijenta iz prethodne linije, kao i prethodnog koeficijenta iz tekuće linije, koriste se dve pomoćne memorije, POMOĆNA MS MEMORIJA i POMOĆNA S MEMORIJA. Na Slici 4.12 prikazan je proces skaniranja jedne linije za računanje konteksta za koeficijente sa indeksima 0, 1, 2, n-2 i n-1, kao i nakon završetka skaniranja linije. Kako se iz memorije može u jednom trenutku čitati i upisivati samo po jedan podatak, neophodni su pomoćni registri (ukupno 5 po jednoj pomoćnoj memoriji). Prilikom računanja konteksta za koeficijent sa indeksom 0 iz memorije se čita koeficijent (ili znak) sa indeksom 1 iz prethodne linije, dok se koeficijent sa indeksom 0 čita iz prvog pomoćnog registra. Drugi pomoćni registar koji u sebi sadrži levi prethodnik iz tekuće linije ažurira se vrednošću tekućeg koeficijenta. Zbog kauzalnog proširenja sa leve strane nije potreban nikakav podatak o

koeficijentu sa leve strane slike, to jest on se uzima da je jednak nuli. Prilikom računanja konteksta za koeficijent sa indeksom 1 iz memorije se čita koeficijent sa indeksom 2 iz prethodne linije, koeficijent sa indeksom 0 iz prethodne linije se čita iz prvog pomoćnog registra, levi prethodnik iz tekuće linije se čita iz drugog pomoćnog registra, dok se koeficijent sa indeksom 1 iz prethodne linije čita iz trećeg pomoćnog registra koji u sebi sadrži prethodno pročitani sadržaj memorije. Istovremeno se ažurira drugi pomoćni registar tekućim koeficijentom (ili znakom) iz tekuće linije. Operacija prilikom računanja konteksta za koeficijent (ili znak) sa indeksom 2 ima generičku važnost, za indeks  $k$  osim za dva poslednja koeficijenta iz linije. Iz memorije se čita koeficijent sa indeksom  $k+1$  iz prethodne linije, koeficijent sa indeksom  $k$  iz prethodne linije se dobija iz trećeg pomoćnog registra, levi prethodnik iz tekuće linije se čita iz drugog pomoćnog registra, dok se koeficijent sa indeksom  $k-1$  iz prethodne linije čita iz četvrtog pomoćnog registra koji u sebi sadrži pročitani sadržaj iz preposlednjeg čitanja memorije. Istovremeno se ažurira drugi pomoćni registar tekućim koeficijentom (ili znakom) iz tekuće linije, kao i sadržaj memorije na adresi  $k-1$  sadržajem drugog pomoćnog registra. Prilikom računanja konteksta za koeficijent sa indeksom  $n-2$  koeficijent sa indeksom  $n-1$  iz prethodne linije se dobija iz petog pomoćnog registra, koeficijent sa indeksom  $n-2$  iz prethodne linije se čita iz trećeg pomoćnog registra, levi prethodnik iz tekuće linije se čita iz drugog pomoćnog registra, dok se koeficijent sa indeksom  $n-3$  iz prethodne linije čita iz četvrtog pomoćnog registra. Kao što se vidi, u ovom ciklusu nema čitanja iz memorije. U memoriju se upisuje sadržaj drugog pomoćnog registra na adresu  $n-3$ . Prilikom računanja konteksta za koeficijent sa indeksom  $n-1$  koeficijent sa indeksom  $n$  iz prethodne linije, koji predstavlja simetrično produženje, dobija se iz petog pomoćnog registra, koeficijent sa indeksom  $n-1$  iz prethodne linije se čita iz trećeg pomoćnog registra, levi prethodnik iz tekuće linije se čita iz drugog pomoćnog registra, dok se koeficijent sa indeksom  $n-2$  iz prethodne linije čita iz četvrtog pomoćnog registra. Ni u ovom ciklusu nema čitanja iz memorije. U memoriju se upisuje sadržaj drugog pomoćnog registra na adresu  $n-2$ , dok se u peti registar upisuje vrednost tekućeg koeficijenta (ili znaka). Na kraju ove procedure, u prvom pomoćnom registru nalazi se koeficijent sa indeksom 0, u petom pomoćnom registru koeficijent sa indeksom  $n-1$ , dok se koeficijenti sa indeksima  $1...n-1$  nalaze u memoriji.

Blokovi VLC ENKODOVANJE i ENKODOVANJE OPSEGA imaju složenu strukturu, i u njima se vrši stvarno entropijsko enkodovanje.

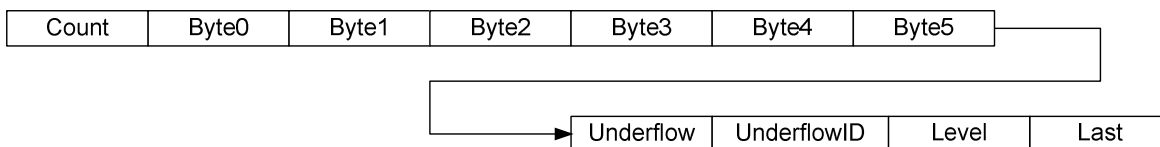
Kako entropijski enkoderi mogu u jednom trenutku generisati više izlaznih bajtova, potreban je mehanizam za pretvaranje niza podataka koji se sastoji iz varijabilnog (ali ograničenog) broja bajtova u niz individualnih bajtova.

U slučaju VLC enkodovanja koristi se VLC FIFO koji ima strukturu zapisa prikazanu na **Slici 4.13**. Polje *Count* definiše koliko u zapisu ima validnih bajtova (maksimalno dva). Naredna dva polja sadrže vrednosti do dva važeća bajta. Polje *Level* označava kom nivou transformacionih koeficijenata pripada zapis. Polje *Last* signalizira da je u pitanju poslednji zapis za dati nivo transformacionih koeficijenata. VLC SERIALIZER 2/1 čita zapise iz bloka VLC FIFO i na izlazu generiše niz bajtova.

Count	Byte0	Byte1	Level	Last
-------	-------	-------	-------	------

**Slika 4.13** Organizacija zapisa u VLC FIFO

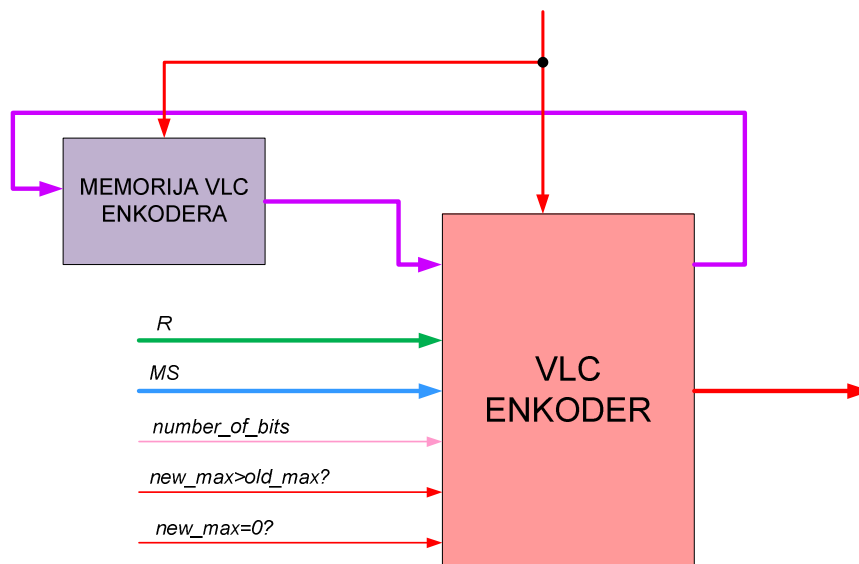
U slučaju enkodera opsega koristi se RC FIFO čiji zapis ima strukturu prikazanu na **Slici 4.14**. Polje *Count* definiše koliko u zapisu ima validnih bajtova (maksimalno šest). Narednih šest polja sadrže vrednosti do šest važećih bajtova. U jednom prolasku algoritma enkodera opsega generisaće se maksimalno jedno potkoračenje, tako da se umesto upisa Underflow bajtova potkoračenja upisuju tri podatka: vrednost bajta potkoračenja na nekom od polja *Byte0...Byte5*, pozicija bajta koji predstavlja potkoračenje na polju *UnderflowID* i vrednost potkoračenja na polju *Underflow*. Polje *Level* označava kom nivou transformacionih koeficijenata pripada zapis. Polje *Last* signalizira da je u pitanju poslednji zapis za dati nivo transformacionih koeficijenata. Blokovi MS SERIALIZER i S SERIALIZER čitaju svaki odgovarajući FIFO i generišu niz bajtova na izlazu, korišćenjem RLE enkodovanja [1], [2], [103].



**Slika 4.14** Organizacija zapisa u MS FIFO i S FIFO

Linija za kašnjenje sa **Slike 4.11** obezbeđuje zakašnjene kontrolne signale za protočnu obradu, što povećava maksimalnu radnu frekvenciju.

Blok šema bloka VLC ENKODOVANJE sa **Slike 4.11** prikazana je na **Slici 4.15**.



**Slika 4.15** Blok šema bloka VLC ENKODOVANJE

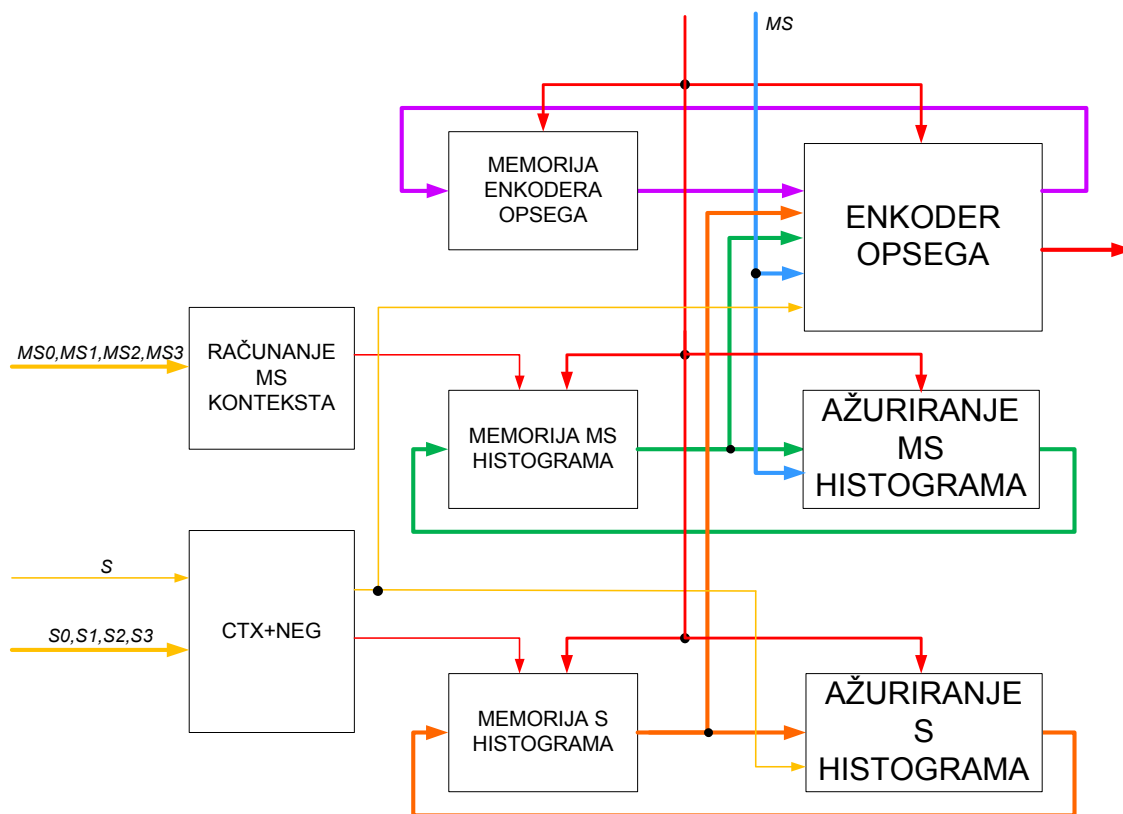
VLC enkoder ima svoj kontekst koji se čuva u bloku MEMORIJA VLC ENKODERA. Kontekst se sastoji od maksimalno 7 bita još uvek neposlalih podataka, kao i informaciju o broju neposlalih bita. Ukoliko se u jednom ciklusu enkoduje ostatak  $R$  sa maksimalno 16 bita, u VLC enkoderu se proverava da li je ukupan novi broj bita (broj neposlalih plus broj novih za enkodovanje na ulazu) veći ili jednak od 16 bita kada se na izlaz prosleđuju dva bajta, ili manji od 16 ali veći ili jednak od 8 bita kada se na izlaz prosleđuje jedan bajt. Ukoliko je ukupan novi broj bita manji od 8 na izlaz se ne prosleđuje nikakav podatak. U sva tri slučaja se ažurira kontekst novim brojem neposlalih bita i novom vrednošću neposlalih bita. MEMORIJA VLC ENKODERA ima



ukupno 10 zapisa u formatu (neposlati biti, broj neposlatah bita), za maksimalno 10 nivoa transformacionih koeficijenata.

Blok šema bloka ENKODOVANJE OPSEGA sa **Slike 4.11** prikazana je na **Slici 4.16**.

U bloku RAČUNANJE MS KONTEKSTA računa se kontekst amplitudne grupe na osnovu vrednosti četiri susedna koeficijenta. Ovo nije potrebno ukoliko je u pitanju LL podopseg koji ima samo jedan kontekst. Na osnovu broja konteksta, iz bloka MEMORIJA MS HISTOGRAMA očitava se potrebnii histogram. Histogram se ažurira u bloku AŽURIRANJE MS HISTOGRAMA, i istovremeno je ulazni podatak u blok ENKODER OPSEGA. Enkoder opsega ima svoj kontekst, koji se čuva u bloku MEMORIJA ENKODERA OPSEGA, za svaki od nivoa posebno.



**Slika 4.16** Blok šema bloka ENKODOVANJE OPSEGA

U bloku CTX+NEG implementirane su tabele za računanje konteksta znaka na osnovu vrednosti četiri susedna koeficijenta prema **Tabeli 4.2** i eventualnu promenu znaka koji

se enkoduje, prema **Tabeli 4.3**. Kontekst znaka nije važan ukoliko je u pitanju LL podopseg koji ima samo jedan kontekst. Na osnovu broja konteksta, iz bloka MEMORIJA S HISTOGRAMA očitava se potrebni histogram. Histogram se ažurira u bloku AŽURIRANJE S HISTOGRAMA, i istovremeno je ulazni podatak u blok ENKODER OPSEGA.

Iskorišćenje hardverskih resursa i performanse nakon implementacije bloka enkoderskog procenjivača verovatnoća i entropijskog enkodera opisanog ranije, u integrisanom kolu Altera EP4CE115F29C7, korišćenjem softverskog alata Quartus II 12.1 prikazano je u **Tabeli 4.10**.

**Tabela 4.10**

Hardverski resursi i performanse enkoderskog procenjivača verovatnoća i entropijskog enkodera u Altera EP4CE115F29C7

broj logičkih elemenata	<b>7850</b>
količina memorije	<b>148106 bita</b>
množači	<b>0</b>
maksimalna radna učestanost @85C	<b>109.71 MHz</b>

Pojedinačni naučni doprinos prikazan u ovom poglavlju obuhvata optimizaciju izračunavanja konteksta i procenjivača verovatnoća simbola za implementaciju u programabilnim logičkim kolima, kao i optimizaciju entropijskog enkodera za implementaciju u programabilnim logičkim kolima, za novi protok podataka unutar modula enkodera slike definisanog u prethodnom poglavlju.

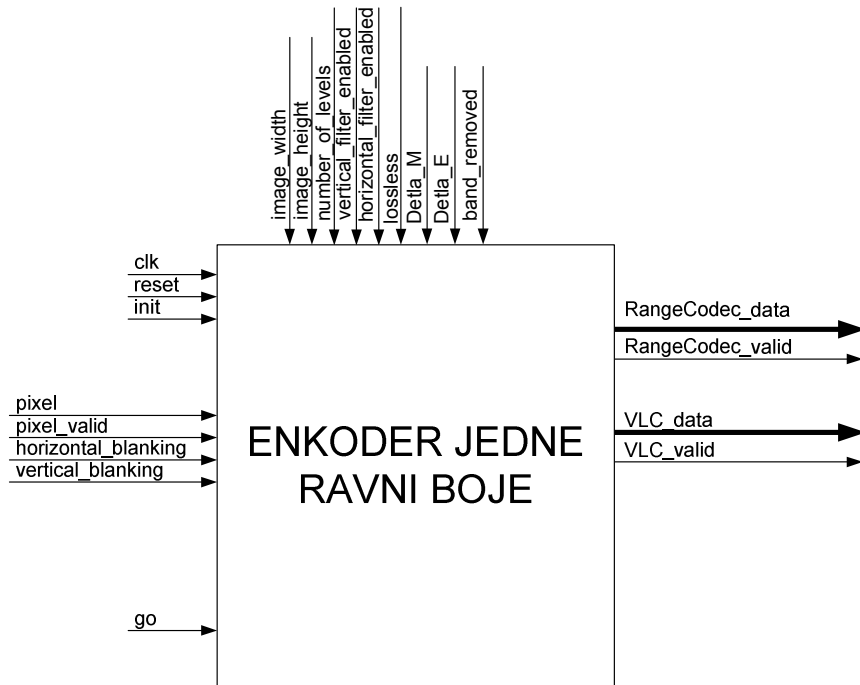
## Simulacija hardverskog enkodera jedne ravni boje

**N**ajviši nivo hijerarhije hardverskog enkodera za jednu ravan boje prikazan je na **Slici 5.1**, dok je interna struktura hardverskog enkodera prikazana na **Slici 5.2**. Značenja pojedinih signala već su objašnjena prilikom prikaza hardverske realizacije tri sastavna bloka hardverskog enkodera.

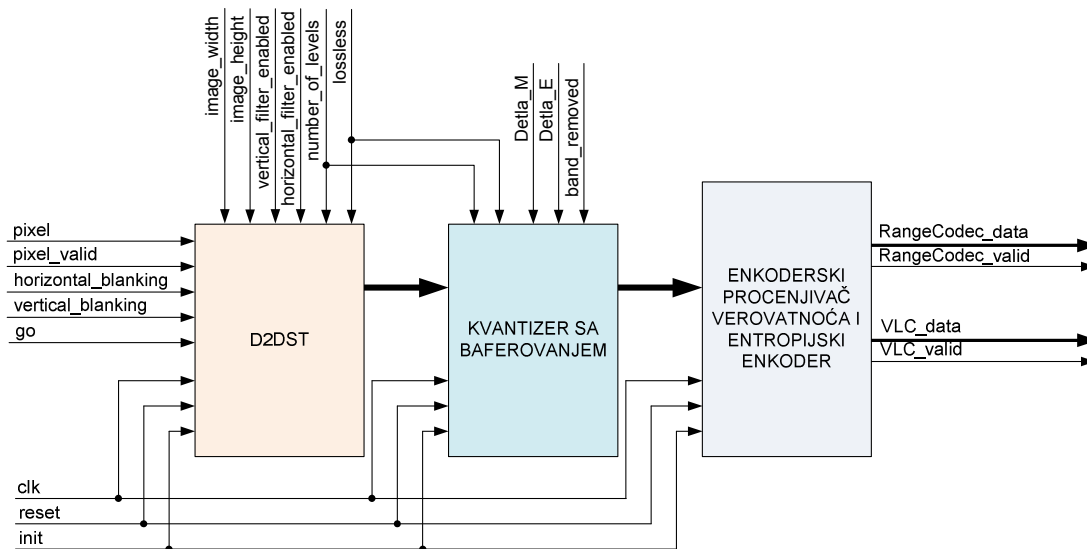
Funkcionalna simulacija hardverskog enkodera urađena je u softverskom alatu *Modelsim SE 6.5d*. Kako je ciljna grupa ove realizacije hardvera bila proizvod kompanije Altera, u toku dizajniranja hardvera korišćene su pojedine ugrađene komponente (memorijski blokovi i množači) obezbeđene od strane proizvođača. Posledica ovoga je da su bile potrebne i simulacione biblioteke za ispravnu simulaciju ugrađenih komponenti. Rezultati simulacije su nezavisni od konkretne komponente, dokle god je komponenta proizvod Altere i podržava ugrađene hardverske blokove koji

## Simulacija hardverskog enkodera jedne ravni boje

se koriste. U konkretnom slučaju ti blokovi obuhvataju ugrađene 9-bitne množače i memorijske blokove kapaciteta 9kbita (označenih sa M9K od strane proizvođača).



Slika 5.1 Najviši hijerarhijski nivo hardverskog enkodera za jednu ravan boje



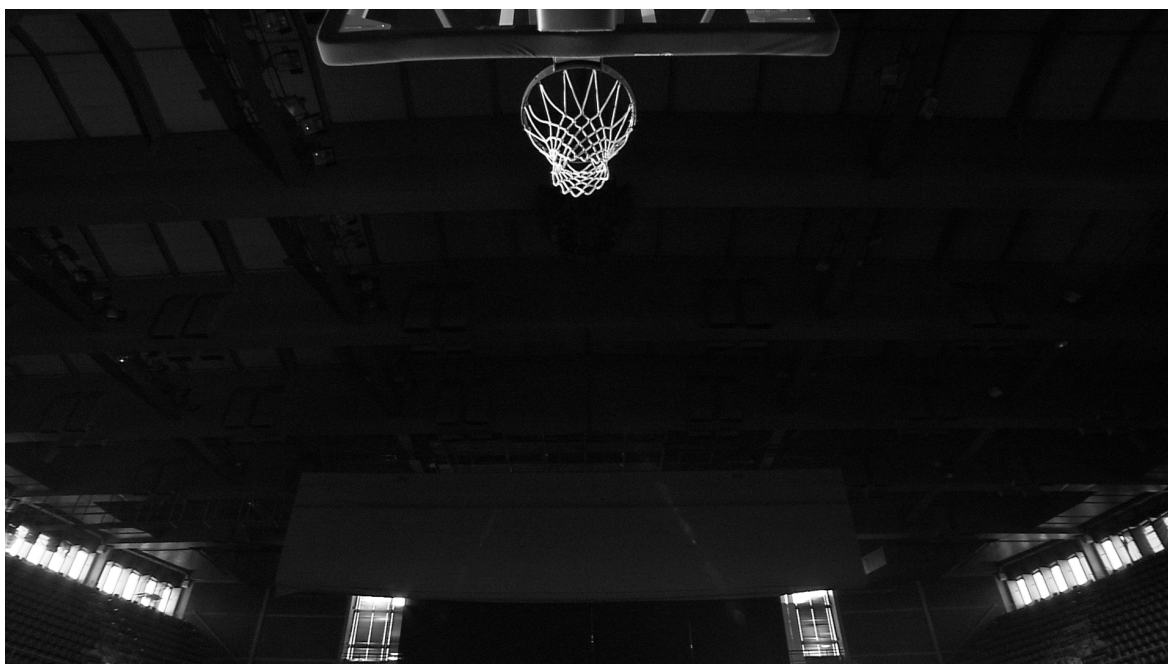
Slika 5.2 Interna struktura hardverskog enkodera jedne ravni boje

## Simulacija hardverskog enkodera jedne ravni boje

Simulacija je rađena sa sledećim vrednostima nepromenljivih ulaznih signala:

<i>lossless</i>	<i>0b</i>	<i>kompresija sa gubicima</i>
<i>number_of_levels</i>	<i>7d</i>	<i>sedam nivoa dekompozicije</i>
<i>horizontal_filter_enabled</i>	<i>0001111111b</i>	<i>uključeni svi horizontalni filtri</i>
<i>vertical_filter_enabled</i>	<i>0001111111b</i>	<i>uključeni svi vertikalni filtri</i>
<i>band_removed</i>	<i>1110000000</i>	<i>svi podopsezi prisutni</i>
<i>Delta_M</i>	<i>111d</i>	<i>mantisa za kvantizaciju</i>
<i>Delta_E</i>	<i>-2d</i>	<i>eksponent za kvantizaciju</i>
<i>image_width</i>	<i>1920</i>	<i>slika visoke rezolucije (HDTV)</i>
<i>image_height</i>	<i>1080</i>	<i>slika visoke rezolucije (HDTV)</i>

Trajanje pauze između dve linije ulazne slike izabrano je da bude 900 ciklusa takta ulaznih piksela (clk), što je minimalna vrednost definisana kamerom koja se koristi za implementaciju u realnom hardveru. Naredni rezultati simulacije odnose se na Y ravan boje prve slike test sekvence *basket* dimenzija 1920 x 1080 (**Slika 5.3**).

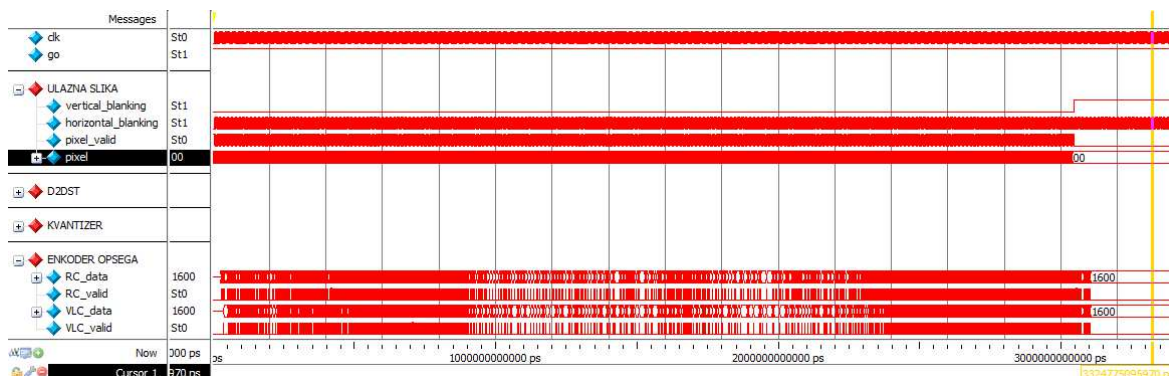


**Slika 5.3** Test slika

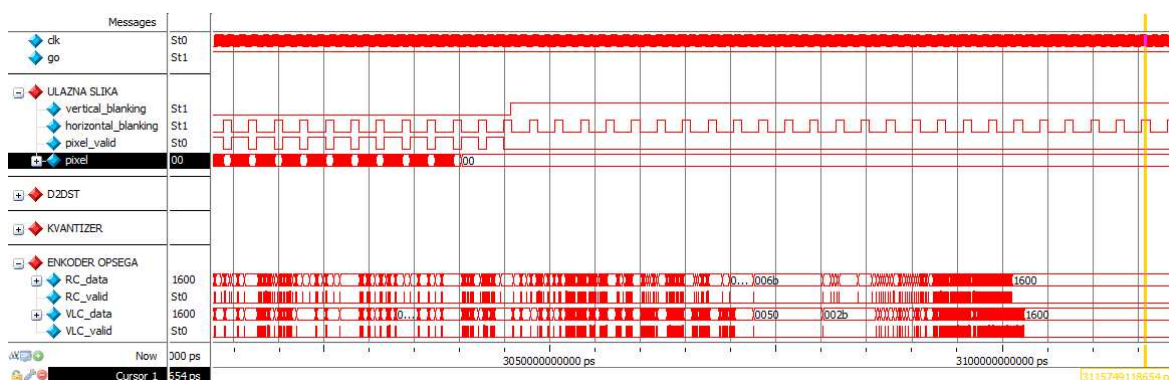
Na **Slici 5.4** prikazani su rezultati simulacije tokom jedne ulazne slike. Naglašeni su ulazni i izlazni signali enkodera. Primećuje se da enkodovanje slike završava nešto

## Simulacija hardverskog enkodera jedne ravni boje

kasnije od završetka ulazne slike. Na **Slici 5.5** je prikazan kraj ulazne slike sa većim nivoom detalja, gde se može proceniti da je kašnjenje približno nešto manje od trajanja dvadeset i jedne ulazne linije.



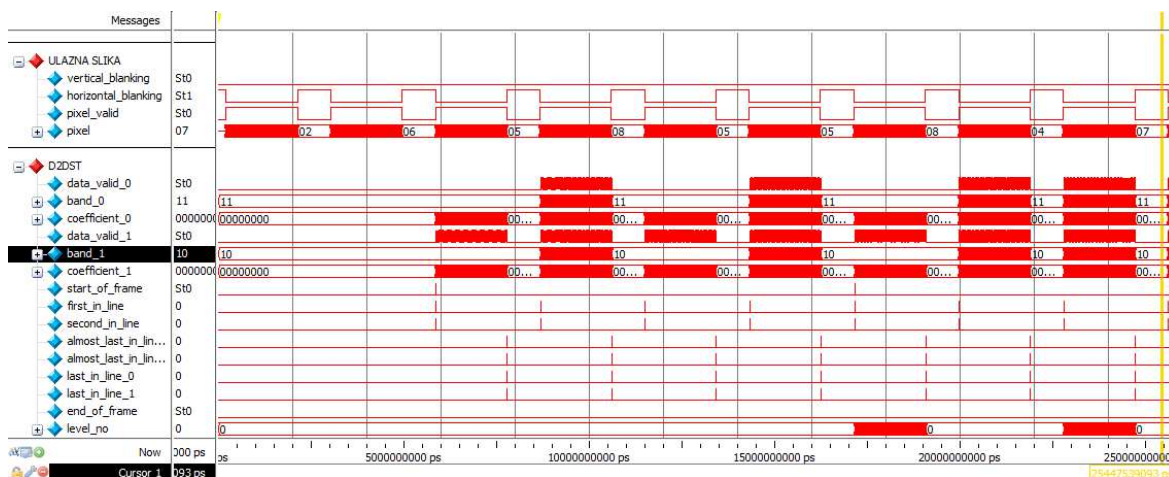
**Slika 5.4** Vremenski oblici ulaznih i izlaznih signala hardverskog enkodera jedne ravni boje, za vreme trajanja jedne ulazne slike



**Slika 5.5** Vremenski oblici ulaznih i izlaznih signala hardverskog enkodera jedne ravni boje, na kraju ulazne slike

Na **Slici 5.6** prikazani su rezultati simulacije za vreme trajanja prvih devet ulaznih linija. Naglašeni su ulazni signali enkodera, koji su istovremeno ulazni signali podopsežnog transformatora, i izlazni signali podopsežnog transformatora. Primećuje se da generisanje transformacionih koeficijenata na izlazu podopsežnog transformatora počinje za vreme treće linije ulazne slike, što se i očekuje prema **Slici 3.5**. Kako je i zahtevano, za vreme trajanja treće i pete linije generišu se samo transformacioni koeficijenti iz LH opsega na nivou 0. Za vreme trajanja četvrte, šeste i osme linije generišu se transformacioni koeficijenti iz HL i HH podopsega na nivou 0. Prvi koeficijenti na drugom nivou podopsežne transformacije pojavljuju se za vreme trajanja sedme ulazne linije, kada se učešljano generišu LH koeficijenti na nivouima 0 i 1. Tokom

devete ulazne linije učešljano se generišu LH koeficijenti na nivou 0 i HL i HH koeficijenti na nivou 1. Rezultati simulacije koji sa većim nivoom detalja prikazuju ulazne i izlazne signale podopsežnog transformatora za vreme jednog dela devete ulazne linije prikazani su na **Slici 3.6**. Podopsezi se kodiraju na sledeći način: 11 – LL, 10 – LH, 01 – HL, 00 – HH.



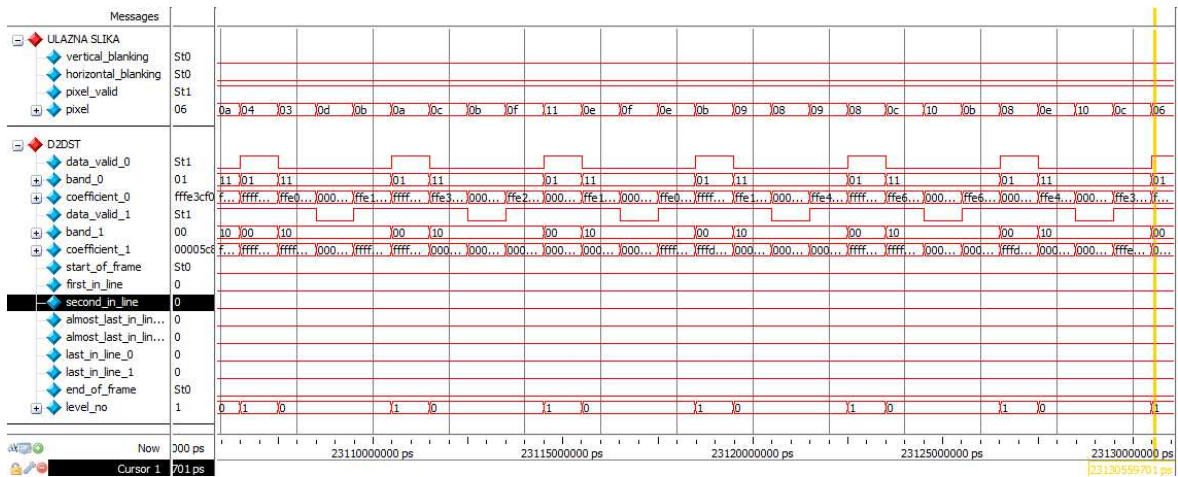
**Slika 5.6** Vremenski oblici ulaznih i izlaznih signala D2DST, za vreme prvih devet linija ulazne slike

Iz rezultata simulacije je moguće uočiti kada relativno u odnosu na ulaznu sliku počinju da generišu izlazne transformacione koeficijente pojedinačni nivoi podopsežnog transformatora. Na **Slici 5.8** to su trenuci kada se aktivira signal `start_of_frame`. Kako postoji sedam nivoa podopsežne transformacije, to se ovaj signal aktivira sedam puta, u trenucima koji su u skladu sa predviđanjima.

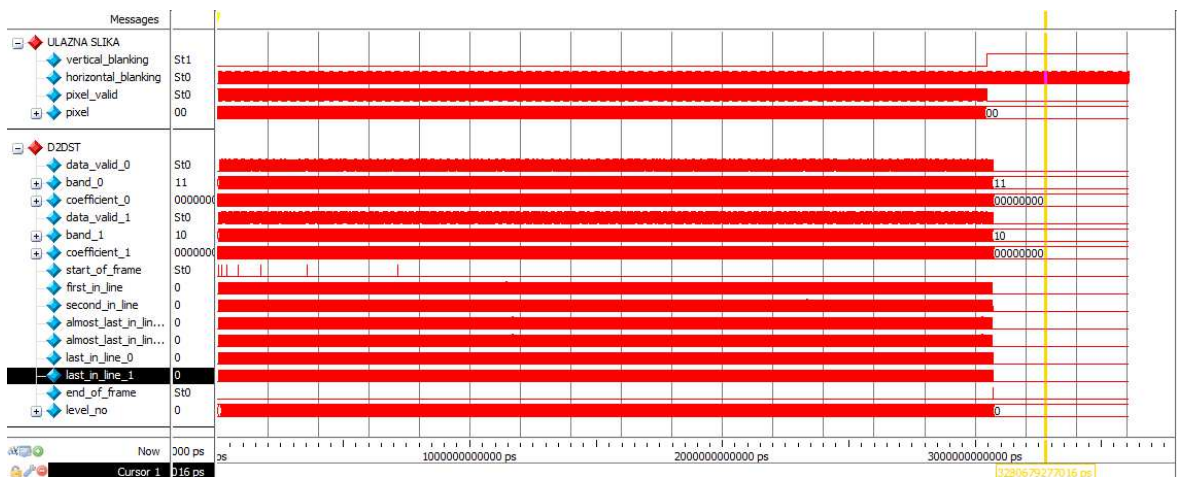
Od interesa je utvrditi koliko ciklusa je približno potrebno nakon završetka ulazne linije da podopsežni transformator završi obradu tekuće linije. Najgori slučaj se javlja kada podopsežni transformator učešljano generiše koeficijete na svim nivoima, što se dešava na primer za vreme 255-e ulazne linije. Sa **Slike 5.9** se vidi da je potrebno nešto manje od 70 ciklusa.

Vreme potrebno da podopsežni transformator završi operaciju na kraju ulazne slike može se utvrditi na osnovu **Slike 5.10**, koja prikazuje vremenske oblike ulaznih i izlaznih signala transformatora na kraju ulazne slike. Ovo vreme iznosi devet ulaznih linija.

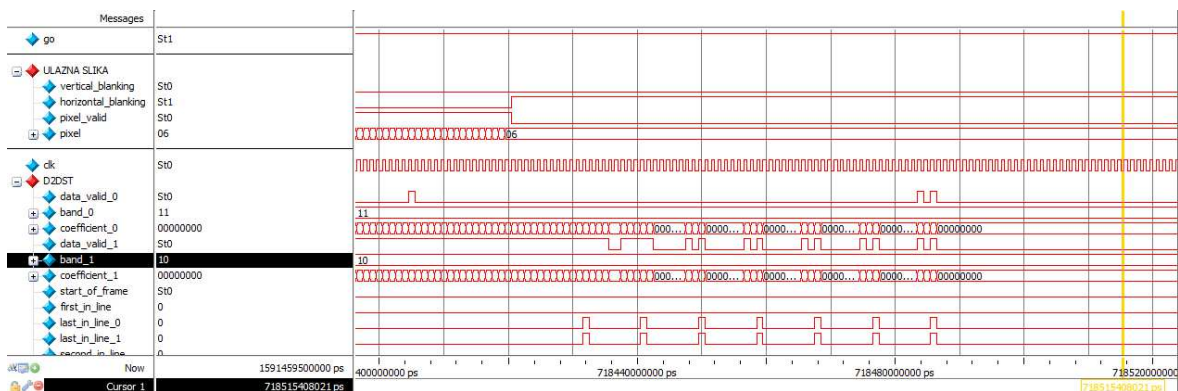
# Simulacija hardverskog enkodera jedne ravni boje



Slika 5.7 Vremenski oblici ulaznih i izlaznih signala D2DST, za vreme jednog dela trajanja devete linije ulazne slike



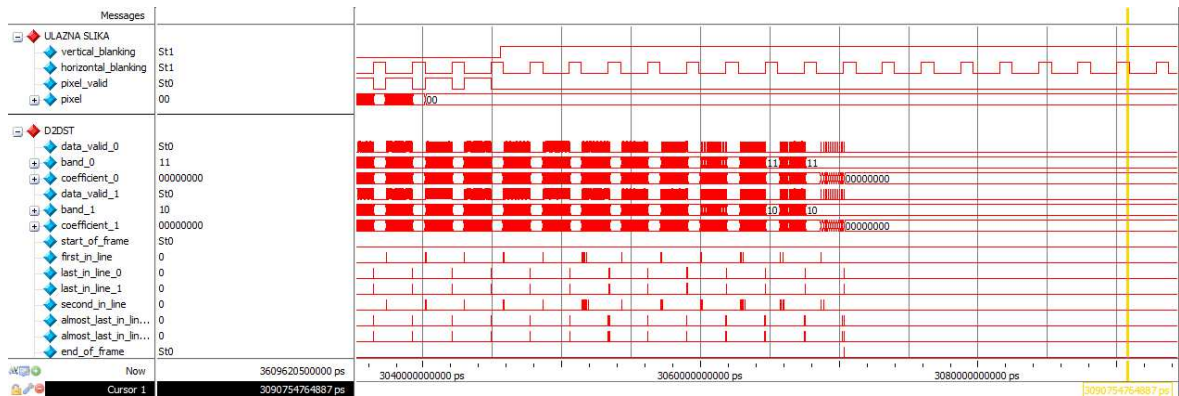
Slika 5.8 Vremenski oblici ulaznih i izlaznih signala D2DST, za vreme trajanja jedne ulazne slike



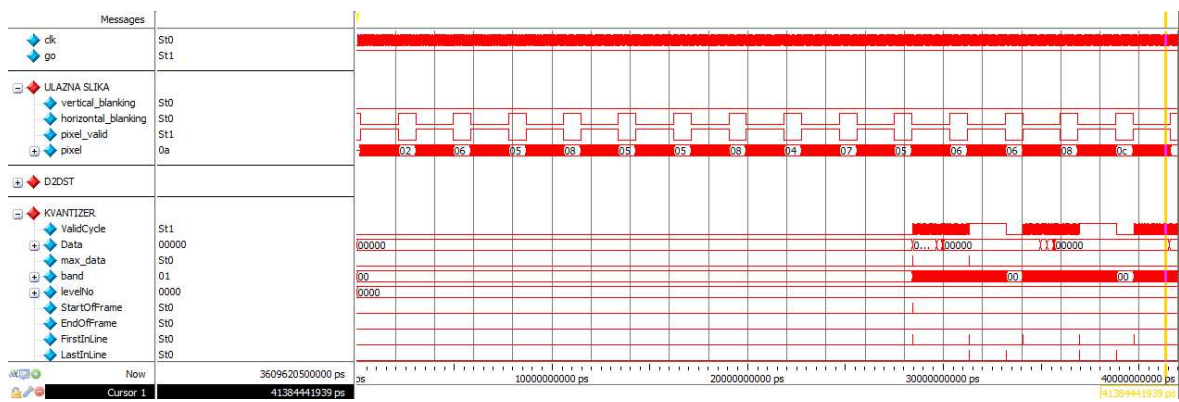
Slika 5.9 Vremenski oblici ulaznih i izlaznih signala D2DST na kraju 255. ulazne linije



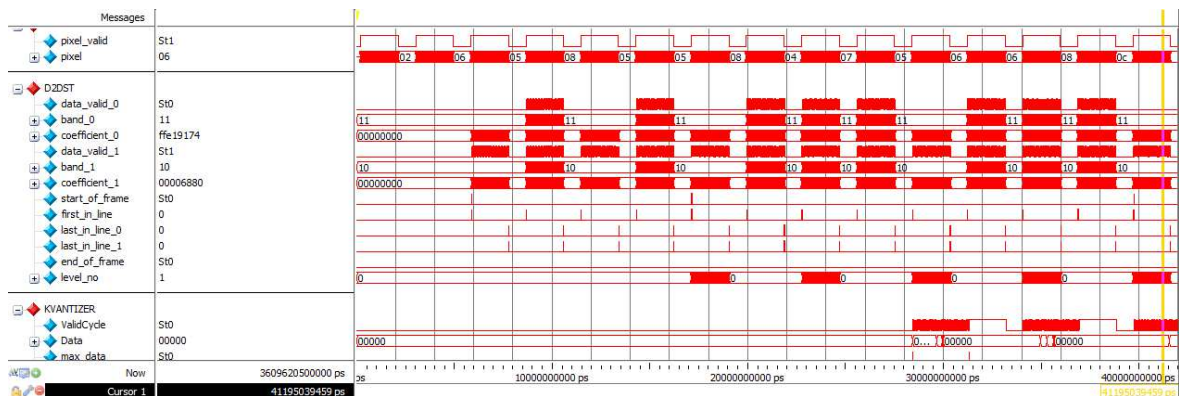
# Simulacija hardverskog enkodera jedne ravni boje



Slika 5.10 Vremenski oblici ulaznih i izlaznih signala D2DST na kraju ulazne slike



Slika 5.11 Vremenski oblici ulaznih signala enkodera i izlaznih signala kvantizatora za vreme prvih petnaest linija ulazne slike

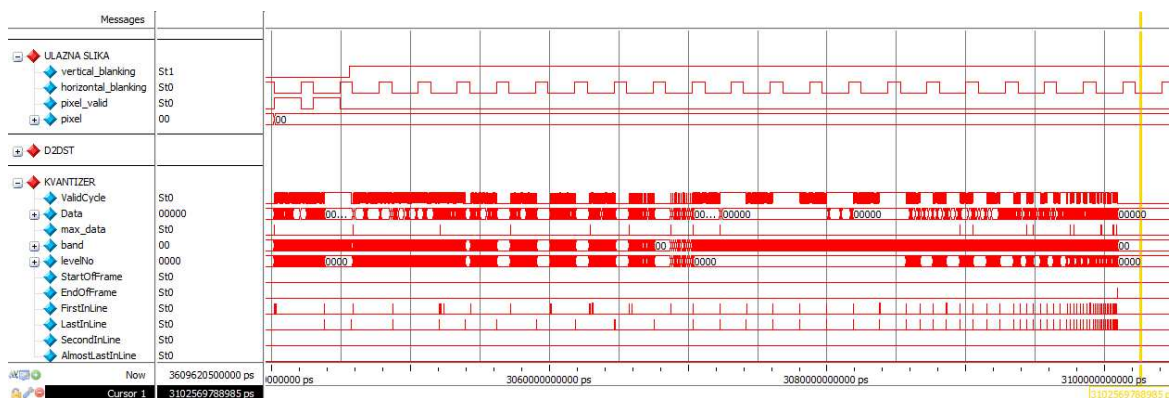


Slika 5.12 Vremenski oblici ulaznih i izlaznih signala bloka kvantizatora sa baferovanjem za vreme prvih petnaest linija ulazne slike

Sa Slike 5.11, na kojoj su prikazani ulazni signali enkodera i izlazni signali bloka kvantizatora sa baferovanjem, potvrđuje se da se početak generisanja signala na izlazu bloka kvantizatora sa baferovanjem javlja tokom jedanaeste ulazne linije, što je

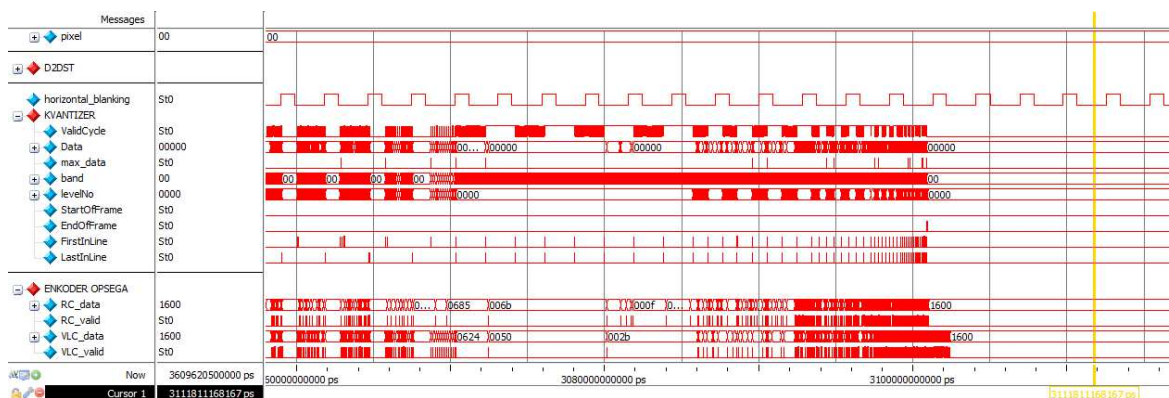
## Simulacija hardverskog enkodera jedne ravni boje

predviđeno na **Slici 3.5**. Isti vremenski period, ali sa naglaskom na ulaznim i izlaznim signalima bloka kvantizatora sa baferovanjem, prikazan je na **Slici 5.12**. Kašnjenje od jedanaest linija predstavlja ugrađeno kašnjenje enkodera na početku ulazne slike, koje je zbir kašnjenja podopsežnog transformatora (dve linije) i kašnjenja bafera u bloku kvantizatora sa baferovanjem (po četiri linije svih podopsega na nivou 0, odnosno osam ulaznih linija).



**Slika 5.13** Vremenski oblici ulaznih signala enkodera i izlaznih signala bloka kvantizatora sa baferovanjem na kraju ulazne slike

Vreme potrebno da se isprazne baferi u bloku kvantizatora sa baferovanjem nakon završetka ulazne slike očitava se sa **Slike 5.13**, i iznosi dvadeset ulaznih linija.



**Slika 5.14** Vremenski oblici izlaznih signala bloka kvantizatora sa baferovanjem i izlaznih signala enkodera na kraju ulazne slike

Kako se vidi sa **Slike 5.14**, koja prikazuje izlazne signale bloka kvantizatora sa baferovanjem i izlazne signale enkodera, postoji dodatno kašnjenje na izlazu VLC

## Simulacija hardverskog enkodera jedne ravni boje

enkodera, koje je posledica rada VLC serializera u okviru bloka enkoderskog procenjivača verovatnoća i entropijskog enkodera.

U ovom poglavlju prikazani su rezultati simulacije modula hardverskog enkodera samo jedne ravni boje, čime je verifikovana funkcionalnost ovog modula.

## Implementacija sistema za kompresiju slike u boji

**P**rojektovanje hardvera korišćenjem HDL jezika omogućava jednostavnu portabilnost dizajna sa jedne platforme na drugu. Tako je i realizacija enkodera iz ove teze pogodna za implementaciju u bilo kojem programabilnom logičkom kolu, pod uslovom da je korišćeno programabilno logičko kolo dovoljnog kapaciteta. Prilikom migracija između različitih porodica programabilnih logičkih kola različitih proizvođača potrebno je samo uzeti u obzir specifičnosti arhitektura, kao što su postojanje i vrsta realizacije blokova ugrađene memorije i ugrađenih množača.

Sistem za kompresiju slike u boji implementiran je u okviru razvojne ploče tPad Multimedia Development Kit, proizvođača Terasic Technologies [104], [105]. U okviru ove razvojne ploče nalazi se komponenta EP4CE115F29C7 iz familije Cyclone IVE

## Implementacija sistema za kompresiju slike u boji

proizvođača Altera, u kojoj se nalazi 114480 osnovnih logičkih ćelija, 3888 Kbita ugrađene memorije organizovane u 432 M9K bloka, 4 fazno kontrolisane petlje, 532 ugrađenih 9-bitnih brojača, i 529 pinova na raspoloaganju korisniku [106]. Na ploči je dostupno nekoliko tipova spoljašnjih memorija: 128MB SDRAM, 2MB SRAM, 8MB fleš i 32Kbit EEPROM.

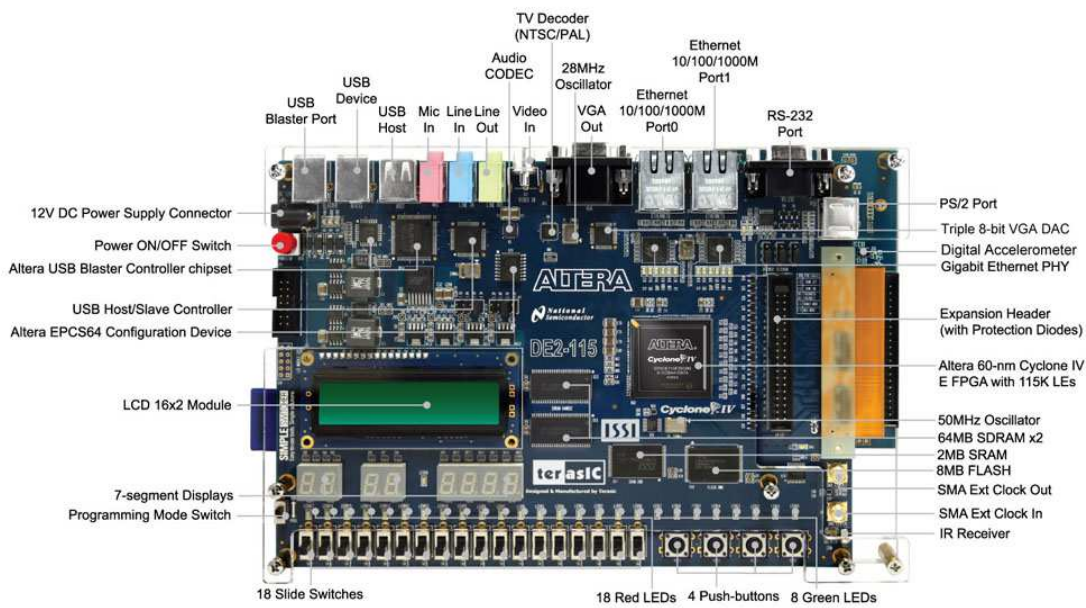
Vežano za multimedijalne aplikacije, na razvojnoj ploči je montiran 8-inčni LCD ekran osetljiv na dodir (*touchscreen*) i senzor slike veličine 5 Megapiksela. Mrežne mogućnosti razvojne ploče obuhvataju dva mrežna interfejsa. Izgled prednje i zadnje strane razvojne ploče dati su na **Slici 6.1** i **Slici 6.2**, respektivno.



**Slika 6.1** Prednja strana razvojne ploče tPad Multimedia Development Kit

U cilju verifikacije rada hardverskog enkodera slike, postavljeno je eksperimentalno okruženje koji uključuje razvojnu ploču na kojoj je implementiran mrežni server koji služi za daljinsku kontrolu i očitavanje hardverskog ekondera slike. Mrežnom serveru pristupa se preko mrežnog interfejsa, kako je i prikazano na **Slici 6.3**. Ukoliko na udaljenom računaru postoji instaliran softverski dekomer kompatibilan sa hardverskim enkoderom, moguće je u stvarnom vremenu prikazivati video sekvencu sa senzora slike koji se nalazi na razvojnoj ploči.

# Implementacija sistema za kompresiju slike u boji



Slika 6.2 Zadnja strana razvojne ploče tPad Multimedia Development Kit



Slika 6.3 Eksperimentalni sistem

Mrežni server je implementiran u softveru koji se izvršava na procesoru NIOS II koji je jedna od komponenti u programabilnoj logičkoj komponenti. U komunikaciji

se koriste TCP i UDP mrežni paketi. TCP paketi služe za kontrolu enkodera, dok UDP paketi služe za prenos komprimovane slike.

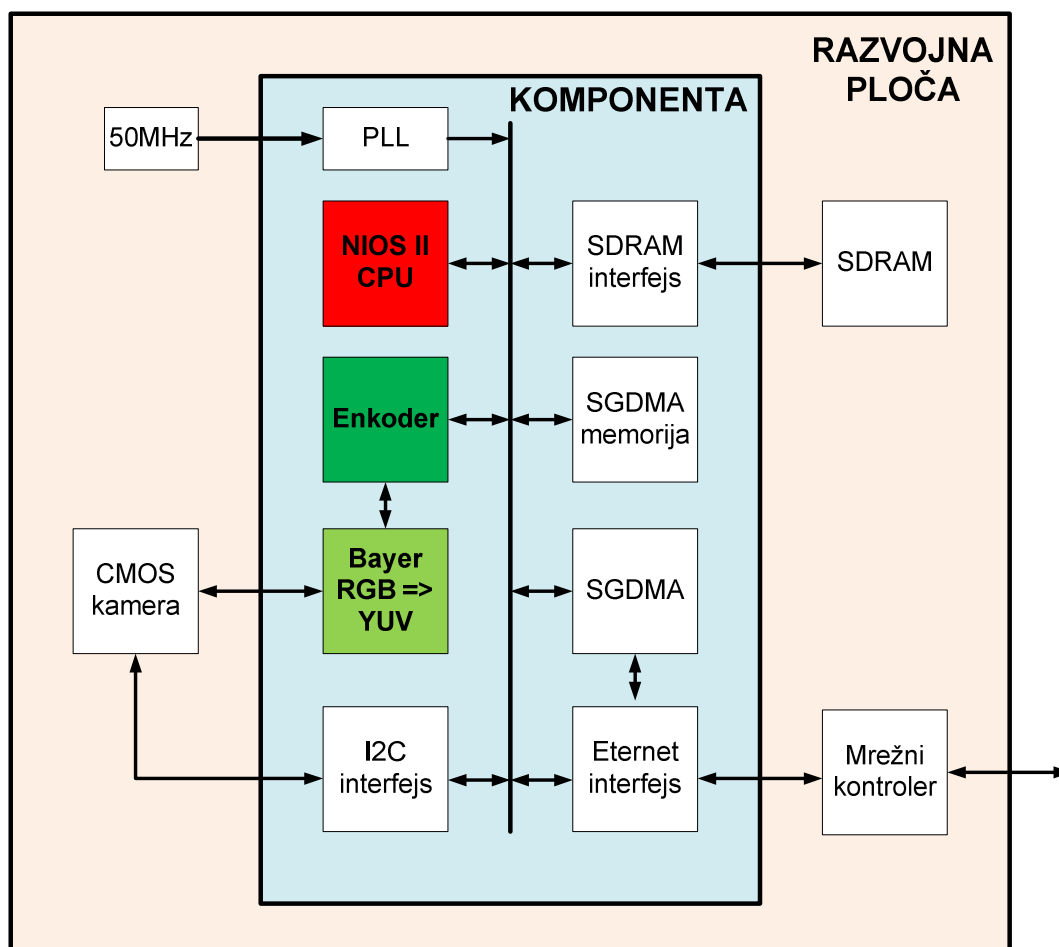
Blok dijagram hardverske realizacije celokupnog sistema prikazan je na **Slici 6.4**. Ulazni takt se obezbeđuje iz izvora takta koji se nalazi na razvojnoj ploči, i u okviru fazno kontrolisane petlje u programabilnoj komponenti se generišu potrebni signali takta za sve komponente u sistemu. Komunikacija između procesora i komponenti sistema obavlja se preko 32-bitne Avalon magistrale [107]. SDRAM memorija na ploči koristi se kao memorija programa i podataka za procesor. SGDMA kontroler obezbeđuje brzi prenos podataka između SDRAM memorije i mrežnog kontrolera na razvojnoj ploči. Za ispravno funkcionisanje SGDMA kontrolera potrebna je memorija za čuvanje liste njegovih poslova, implementirana u okviru komponente.

Kao izvor slike koristi se CMOS senzor slike (kamera) montiran na razvojnoj ploči [108], [109]. Kamera se konfiguriše preko I2C interfejsa. Mogu se podešavati raznovrsni parametri kamere, na primer širina slike, visina slike, brzina sekvence (podešavanjem pauza između linija i između susednih slika), trajanje ekspozicije itd. Ulazni takt kamere obezbeđuje fazno kontrolisana petlja u okviru programabilne komponente. U kameri se nalazi druga fazno kontrolisana petlja koja generiše izlazni takt kamere koji se koristi i za rad enkodera. Izlazni podaci kamere nisu u željenom YUV formatu, već u serijskom RGB Bajerovom formatu. Zato je potreban blok za konverziju slike pre nego što se signal slike dovede na ulaz enkodera.

Principska blok šema komponente enkodera slike prikazana je na **Slici 6.5**. Radi jednostavnosti, nije prikazan interfejs prema magistrali. Enkoder se kontroliše preko skupa registara čiji sadržaj određuje procesor. Pored zajedničkih registara za sve tri ravni boje (go, init, image\_width, image\_height, number\_of\_levels, lossless), postoje i tri skupa registara koja sadrže kontrolne vrednosti za svaki od enkodera jedne ravni boje (horizontal\_filter\_enabled, vertical\_filter\_enabled, band\_removed, Delta\_M, Delta\_E). Očitavanje rezultata enkodovanja omogućeno je preko dve FIFO memorije po enkoderu jedne ravni boje (po jedna za RC i VLC niz podataka),

## Implementacija sistema za kompresiju slike u boji

ukupno šest FIFO memorija za tri ravni boje. Kako signal takta enkodera nije isti kao i signal takta na magistrali, potrebno je obratiti posebnu pažnju na sinhronizaciju između dva domena takta. Iz tog razloga su FIFO memorije implementirane sa različitim taktovima na ulaznom (takt enkodera) odnosno izlaznom (takt magistrale) portu. Što se tiče registara, primenjene su poznate tehnike za sinhronizaciju između dva domena takta [110].

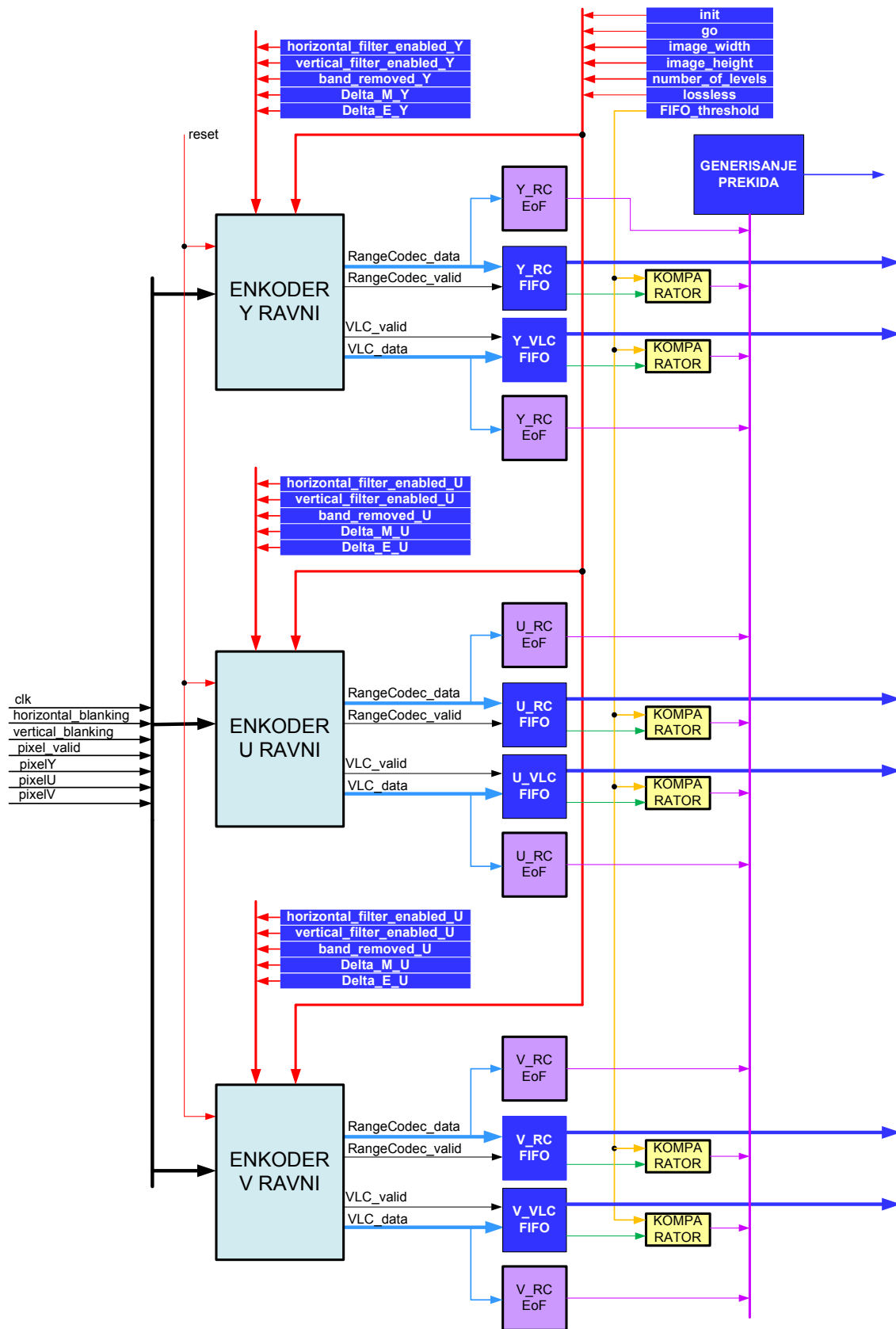


Slika 6.4 Blok dijagram hardverske realizacije mrežnog servera sa enkoderom slike

Procesor se o postojanju izvesne količine spremnih podataka obaveštava preko linije prekida, koja se aktivira ukoliko bilo koja od FIFO memorija sadrži više od određenog broja zapisa (definisano sadržajem registra FIFO\_threshold). Linija prekida se takođe aktivira i kada se svih šest nizova podataka završe za datu ulaznu sliku, što se detektuje u posebnim EoF blokovima.



# Implementacija sistema za kompresiju slike u boji



Slika 6.5 Hardverski enkoder slike

Iskorišćenje hardverskih resursa i performanse hardverskog enkodera slike zavisiće od konkretnog načina povezivanja u širi sistem. Da bi se stekao utisak o redu veličine ovih parametara, iskorišćenje hardverskih resursa i performanse hardverskog enkodera sa **Slike 6.5** u integrisanom kolu Altera EP4CE115F29C7, korišćenjem softverskog alata Quartus II 12.1, prikazani su u **Tabeli 6.1** [111].

**Tabela 6.1**

Hardverski resursi i performanse hardverskog enkodera slike u Altera EP4CE115F29C7

broj logičkih elemenata	<b>54221</b>
količina memorije	<b>2031705 bita</b>
množači (9-bitni)	<b>84</b>
maksimalna radna učestanost @85C	<b>107.08 MHz</b>

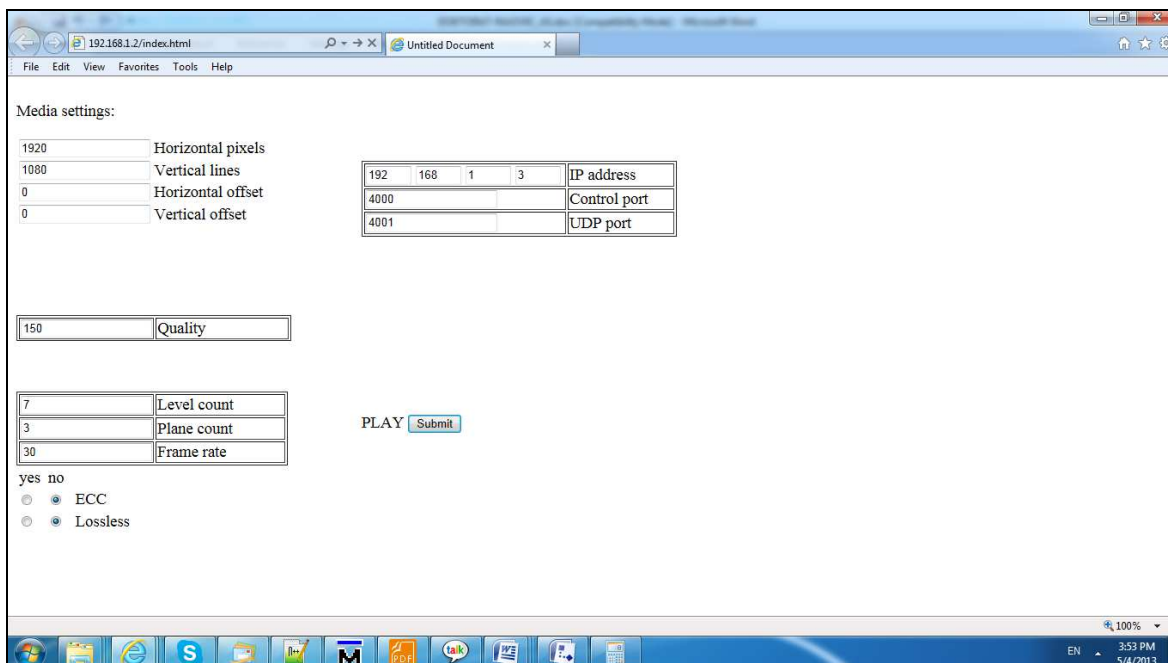
Pojedini specifični parametri koji su korišćeni prilikom testiranja hardverskog enkodera slike u sistemu sa **Slike 6.3** su:

- širina slike 1920 piksela
- visina slike 1080 piksela
- trajanje pauze između linija 900 piksela (minimalna za datu kameru)
- trajanje pauze između slika 50 linija
- brzina sekvence 30 slika u sekundi
- takt kamere i enkodera 96 MHz (maksimalan za datu kameru)
- takt procesora i magistrale 100 MHz
- format ulazne slike 4:2:0

Procedura za korišćenje sistema za kompresiju slike sa **Slike 6.3** započinje otvaranjem adrese hardverskog enkodera u veb pregledaču (*browser*) personalnog računara. Otvara se veb stranica kako je prikazano na **Slici 6.6**. Preko veb stranice se unose bitna podešavanja, kao što su dimenzije slike, broj UDP porta preko koga se šalju komprimovani podaci, mrežna adresa na koju se šalju komprimovani podaci, da li se koristi detekcija greške (implementira je procesor), da li je kompresija sa ili bez gubitaka, stepen kompresije (inverzno proporcionalan parametru Quality) i tako dalje. Pritiskom na taster Submit podešavanja se prebacuju hardverskom enkoderu i započinju komprimovanje slike i slanje komprimovanih podataka slike na izabranu IP adresu preko izabranog UDP porta.

## Implementacija sistema za kompresiju slike u boji

Moguće je u stvarnom vremenu menjati parametre enkodovanja, unosom i ponovnim pritiskom na taster Submit.



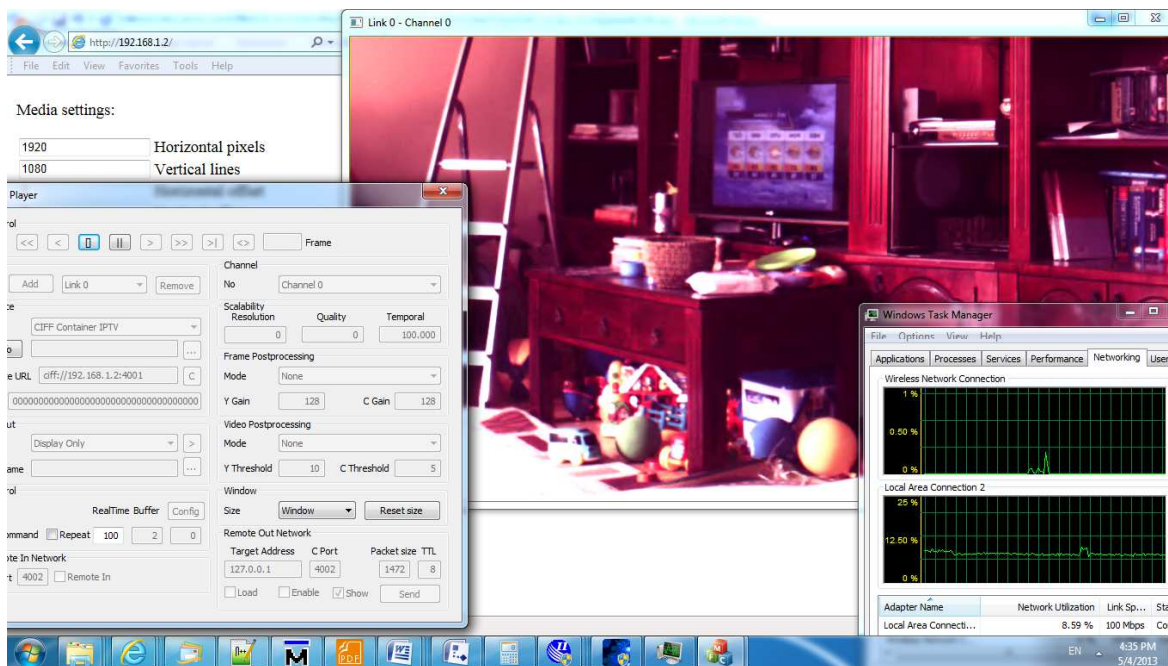
Slika 6.6 Veb stranica hardverskog enkodera slike sa mrežnim interfejsom

Ukoliko na personalnom računaru postoji kompatibilan softverski dekodler, moguće je dekomprimovati pristigle UDP pakete i prikazivati video sekvencu u stvarnom vremenu. Na **Slici 6.7** prikazan je snimak ekrana personalnog računara sa uključenim softverskim dekoderom. Merenja su pokazala da se ovim sistemom postiže bitska brzina reda veličine 11 Mbps (miliona bita u sekundi). Kako je ulazna slika dimenzija 1920x1080 piksela u formatu 4:2:0 i sa brzinom video sekvence 30 fps (slika u sekundi), postignuti stepen kompresije je

$$\frac{11 \text{ Mbps}}{1920 \text{ pixela} \times 1080 \text{ pixela} \times 1.5 \times 8 \text{ b} \times 30 \text{ fps}} = 78.31:1$$

Ova vrednost je ispod realno ostvarive hardverskim enkoderom, kao posledica nemogućnosti mrežnog servera da većom brzinom šalje komprimovane podatke, jer softverske implementacije enkodera imaju stepen kompresije veći od 400:1 [55].

## Implementacija sistema za kompresiju slike u boji



Slika 6.7 Snimak ekrana personalnog računara sa uključenim softverskim dekoderom

Mada je ovaj sistem korišćen samo za testiranje i prezentaciju rada hardverskog enkodera, kao ilustracija hardverskih zahteva, u Tabeli 6.2 dati su iskorišćeni hardverski resursi i performanse hardvera sistema za kompresiju slike sa mrežnim interfejsom u integrisanom kolu EP4CE115F29C7.

Tabela 6.2

Hardverski resursi i performanse sistema za kompresiju slike sa mrežnim interfejsom u Altera EP4CE115F29C7

broj logičkih elemenata	<b>65951</b>
količina memorije	<b>2445904 bita</b>
množači (9-bitni)	<b>88</b>
maksimalna radna učestanost takta enkodera @85C	<b>102.62 MHz</b>

Zaokružen sistem za kompresiju slike u boji sa mrežnim interfejsom prikazan je u ovom poglavlju, kao demonstracija funkcionalnosti razvijenog kodeka.

## Zaključak

**U** okviru hardverske realizacije dvodimenzionalnog enkodera visokih performansi i sa minimalnim zahtevanim memorijskim resursima izvršena je podela realizacije, u skladu sa trenutnim stanjem tehnike, na tri dela. Svaki od delova – modula je posebno teorijski analiziran, dizajniran i istestiran. Prilikom dizajniranja težilo se i uspeo da se minimizuje potrebna količina memorije kao kritičnog resursa u hardverskim realizacijama.

Najznačajnije karakteristike realizovanog rešenja u odnosu na postojeće realizacije su:

- potrebna količina memorije je nekoliko redova veličine manja od referentnih realizacija, što je obezbeđeno novom realizacijom enkoderskog procenjivača verovatnoće i direktnog dvodimenzionalnog podopsežnog transformatora;
- kako je drastično smanjena potrebna količina memorije, nema potrebe za korišćenjem blokova, pločica ili ramova, tako da ne postoje blokovski artefakti;
- postignuto je minimalno moguće kašnjenje, korišćenjem nove realizacije jednoprolaznog enkoderskog procenjivača verovatnoće i direktnog dvodimenzionalnog podopsežnog transformatora;
- potrebna količina hardverskih resursa opšte namene je smanjena, realizacijom hardverskog enkodera opsega i kvantizatora bez operacija deljenja; i
- realizovana je sjedinjena kompresija sa gubicima i bez gubitaka, za 4:4:4, 4:2:2 i 4:2:0 formate ulazne slike.

Naučni doprinos ove disertacije predstavlja optimizovana implementacija novog jednoprolaznog brzog kodeka u programabilnom logičkom kolu, koja omogućava visoki stepen kompresije uz korišćenje minimalnih logičkih i memorijskih hardverskih resursa, što je postignuto sledećim pojedinačnim dostignućima:

- novi algoritam rada direktnog dvodimenzionalnog podopsežnog transformatora,
- novi protok podataka unutar modula enkodera slike,
- razvoj i optimizacija direktnih nestacionarnih filtara za implementaciju u programabilnim logičkim kolima,
- optimizacija izračunavanja konteksta i procenjivača verovatnoća simbola za implementaciju u programabilnim logičkim kolima,
- optimizacija entropijskog enkodera za implementaciju u programabilnim logičkim kolima.

Realizovani brzi enkoder se može uspešno koristiti za kompresiju dvodimenzionalnih signala koje generišu odgovarajući senzori [112-120], kao i za kompresiju videa, što će biti predmet budućih istraživanja.

# Literatura

- [1] I. Pitas, *Digital image processing algorithms and applications*, John Wiley & Sons, Inc., 2000.
- [2] M. Popović, *Digitalna obrada slike*, Akademska misao, Beograd 2006., ISBN 86-7466-272-2
- [3] O. Egger, P. Fleury, T. Ebrahimi, and M. Kunt, "High performance compression of visual information – A tutorial review – Part I: Still pictures," *Proc. IEEE*, vol. 87, no. 6, pp. 976-1011, June 1999.
- [4] Y. Q. Shi and H. Sun, *Image and video compression for multimedia engineering: Fundamentals, algorithms, and Standards, Second edition*, CRC Press, Boca Raton, FL, Mar. 2008.
- [5] S. Wong, L. Zaremba, D. Gooden, and H. K. Huang, "Radiologic image compression – A review," *Proc. IEEE*, vol. 83, no. 2, pp. 194-219, Feb. 1995.
- [6] N. D. Memon, and K. Sayood, "Lossless image compression: A comparative study," in *Proc. SPIE*, vol. 2418, pp. 8-20, 1995.
- [7] T. Q. Nguyen, "A tutorial on filter banks and wavelets," University of Wisconsin, Madison, WI53706, USA, 1995.
- [8] A. Srivastava, A. B. Lee, E. P. Simoncelli and S-C. Zhu, "On Advances in Statistical Modeling of Natural Images," *Journal of Mathematical Imaging and Vision*, vol. 18, pp 17-33, 2003.
- [9] O. Egger and W. Li, "Subband coding of images using asymmetrical filter banks," *IEEE Trans. Image Processing*, vol. 4, no. 4, pp. 478-485, Apr. 1995.
- [10] R. E. Crochiere, S. A. Weber, J. L. Flanagan, "Digital coding of speech in subbands," *Bell Syst. Tech. J.*, vol. 55, no. 8, pp. 1069-1085, Oct. 1976.
- [11] R. E. Crochiere, J. L. Flanagan and S. A. Webber, U.S. Patent 4048443, Sep. 1977.

- [12] C. Chrysafis and A. Ortega, "Efficient context-based entropy coding for lossy wavelet image compression," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1997, pp. 241-250.
- [13] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. Consum. Electr.*, vol. 46, no. 4, pp. 1103-1127, Nov. 2000.
- [14] M. Boliek, C. Christopoulos, and E. Majani (editors), "JPEG2000 Part I Final Draft International Standard," (ISO/IEC FDIS15444-1), ISO/IEC JTC1/SC29/WG1 N1855, Aug. 2000.
- [15] A. Said and W. A. Pearlman, "Image compression using the spatial-orientation tree," *Proc. IEEE Int. Symp. Circuits Systems*, Chicago, IL, pp. 279-282, May 1993.
- [16] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243-250, June 1996.
- [17] W. A. Pearlman and A. Said, U.S. Patent 5764807, June 1998.
- [18] A. Said and W. A. Pearlman, U.S. Patent 5959560, Sep. 1999.
- [19] J. M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.
- [20] R. W. Buccigrossi and E. P. Simoncelli, "Progressive wavelet image coding based on a conditional probability model," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Munich, Germany, vol. 4, Apr. 21-24, 1997, pp. 2597-2600.
- [21] E. P. Simoncelli and R. W. Buccigrossi, "Progressive wavelet image compression using linear inter-band magnitude prediction," *Proc. 4th Int. Conf. Image Processing*, Santa Barbara, CA, Oct. 26-29, 1997.
- [22] C. Chrysafis and A. Ortega, "Line based, reduced memory, wavelet image compression," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1998, pp. 398-407.



- [23] C. Chrysafis and A. Ortega, "An algorithm for low memory wavelet image compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Kobe, Japan, Oct. 1999, vol. 3, pp. 354-358.
- [24] C. Chrysafis and A. Ortega, "Line based, reduced memory, wavelet image compression," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 378-389, Mar. 2000.
- [25] C. Chrysafis, "Wavelet image compression rate distortion optimizations and complexity reductions," *Ph.D. Thesis*, University of Southern California, USA, Mar. 2000.
- [26] D. Santa-Cruz, T. Ebrahimi, J. Askelöf, M. Larsson, and C. Christopoulos, "JPEG2000 still image coding versus other standards," in *Proc. SPIE 45<sup>th</sup> annual meeting, Applications of Digital Image Process. XXIII*, San Diego, CA, vol. 4115, July 2000, pp. 446-454.
- [27] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Denver, CO, Apr. 1980, pp. 291-294.
- [28] M. J. Smith and T. P. Barnwell, III, "A procedure for designing exact reconstruction filter banks for tree structured subband coders," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, San Diego, CA, Mar. 1984, pp. 27.1.1-27.1.4.
- [29] T. A. Ramstad, "Analysis/synthesis filter banks with critical sampling," in *Proc. Int. Conf. Digital Signal Processing*, Florence, Italy, Sep. 1984.
- [30] M. Vetterli, "Filter banks allowing perfect reconstruction," *Signal Process.*, vol. 10, no. 3, pp. 219-244, Apr. 1986.
- [31] M. J. Smith and T. P. Barnwell, III, "Exact reconstruction techniques for tree structured subband coders," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 3, pp. 434-441, June 1986.
- [32] P. P. Vaidyanathan, "Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having perfect reconstruction property," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 4, pp. 476-496, Apr. 1987.

- [33] P. P. Vaidyanathan, "Quadrature mirror filter bank, M-band extensions and perfect reconstruction technique," *IEEE Acoust., Speech, Signal Process. Mag.*, vol. 4, no. 7, pp. 1035-1037, July 1987.
- [34] M. Vetterli and D. Le Gall, "Perfect reconstruction FIR filter banks: Some properties and factorization," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 1057-1071, July 1989.
- [35] K. Nayebi, T. P. Barnwell, III, and M. J. Smith, "Time domain filter bank analysis: A new design theory," *IEEE Trans. Signal Process.*, vol. 40, no. 6, pp. 1412-1429, June 1992.
- [36] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 532-540, Apr. 1983.
- [37] M. Vetterli, "Multi-dimensional sub-band coding: Some theory and algorithms," *Signal Process.*, vol. 6, no. 2, pp. 97-112, Apr. 1984.
- [38] J. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 5, pp. 1278-1288, Oct. 1986.
- [39] E. H. Adelson, E. P. M. Simoncelli, U.S. Patent 4817182, March 1989.
- [40] A. Zandi, M. Boliek, E. L. Schwartz, M. J. Gormish and A. Keith "CREW lossless/lossy medical image compression," Ricoh California Research Center, Menlo Park, CA94025, USA, Sep. 12, 1995
- [41] A. Zandi, E. L. Schwartz, M. J. Gormish, M. Boliek, U.S. Patent 6195465, Feb. 2001.
- [42] D. Le Gall and A. Tabatabai, "Subband coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, New York, NY, Apr. 1988, pp. 761-765.
- [43] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmonic Analysis*, vol. 3, no. 2, pp. 186-200, Apr. 1996.
- [44] W. Sweldens, "The lifting scheme: Construction of second generation wavelets," *SIAM J. Math. Anal.*, vol. 29, no. 2, pp. 511-546, 1997.

- [45] C. Chrysafis and A. Ortega, "Minimum memory implementations of the lifting scheme," in *Proc. SPIE, Int. Symp. Optical Science Tech.*, San Diego, CA, July 2000, pp. 313-324.
- [46] Y. Bao and C. Kuo, "Design of wavelet-based image codec in memory-constrained environment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 642-650, May 2001.
- [47] C.-H. Yang, J.-C. Wang, J.-F. Wang, and C.-W. Chang, "A block-based architecture for lifting scheme discrete wavelet transform," *IEICE Trans. Fundamentals*, vol. E90-A, no. 5, pp. 1062-1071, May 2007.
- [48] J. Oliver and M. Malumbres, "On the design of fast wavelet transform algorithms with low memory requirements," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 237-248, Feb. 2008.
- [49] S. Rein and M. Reisslein, "Low-memory wavelet transforms for wireless sensor networks: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, pp. 1-17, 2010.
- [50] V. Čeperković and M. Prokin, "Non-stationary One-Dimensional Subband Transformer Filters", in *Proc. 2012 Mediterranean Conference on Embedded Computing (MECO)*, Bar, June 2012, pp. 132-135.
- [51] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using the wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 205-220, Apr. 1992.
- [52] M. Martina and G. Maserà, 'Folded multiplierless lifting-based wavelet pipeline', *Electron. Lett.*, 2007, 43, (5), pp. 27-28.
- [53] P. Longa, A. Miri and M. Bolic, 'Modified distributed arithmetic based architecture for discrete wavelet transforms', *Electron. Lett.*, 2008, 44, (4), pp. 270-272.
- [54] V. Čeperković and M. Prokin, "One-Dimensional Codec With Minimized memory Size", in *Proc. Unitech 2011*, Gabrovo.
- [55] V. Čeperković, S. Pavlović, D. Mirković, M. Prokin, U.S. patent 8306340, Nov. 2012.

- [56] V. Rajović, G. Savić, V. Čeperković, M. Prokin, “Combined one-dimensional lowpass and highpass filters for subband transformer”, *Electron. Lett.*, vol. 49, no. 18, pp. 1150-1152, 2013.
- [57] J. Lin and M. J. T. Smith, “New perspectives and improvements on the symmetric extension filter bank for subband/wavelet image compression,” *IEEE Trans. Image Process.*, vol. 17, no. 2, Feb. 2008.
- [58] M. Vrankic, D. Sesic, and V. Sucic, “Adaptive 2-D wavelet transform based on the lifting scheme with preserved vanishing moments,” *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 1987-2004, Aug. 2010.
- [59] “Digital compression and coding of continuous-tone still images,” Int. Org. Standardization ISO/IEC, JTC1 Committee Draft, JPEG 8-R8, 1990.
- [60] G. K. Wallace, “The JPEG still picture compression standard,” *Commun. ACM*, vol. 34, no. 4, pp. 30-44, Apr. 1991.
- [61] D.-U. Lee, H. Kim, M. Rahimi, D. Estrin, and D. Villasenor, “Energy-efficient image compression for resource-costrained platforms,” *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 2100-2113, Sep. 2009.
- [62] S. Rein, S. Lehmann, and C. Gühmann, “Fractional wavelet filter for camera sensor node with external flash and extremely little RAM,” in *Proc. ACM Mobile Multimedia Commun. Conf.*, July 2008.
- [63] S. Rein and M. Reisslein, “Performance evaluation of the fractional wavelet filter: A low-memory image wavelet transform for multimedia sensor networks”, *Ad Hoc Networks*, 2010.
- [64] S. Rein, S. Lehmann, and C. Gühmann, “Wavelet image two line coder for wireless sensor node with extremely little RAM,” in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2009, pp. 252-261.
- [65] V. Čeperković and M. Prokin, “Fast Image Codec With Minimum Required Memory”, in *Proc. Unitech 2011*, Gabrovo.
- [66] I. Daubechies, and W. Sweldens, “Factoring Wavelet Transforms Into Lifting Steps”, *The Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 247-269, 1998.
- [67] K. P. Soman, K. I. Ramachandran and N. G. Resmi, *Insight into Wavelets from Theory to Practice*, PHI Learning, 2010.

- [68] T. Acharya and P.-S. Tsai, *JPEG2000 Standard for Image Compression Concepts, Algorithms and VLSI Architectures*, John Wiley and Sons, 2005.
- [69] D. Goldberg, "What every computer scientist should know about floating-point arithmetic", *ACM Computing Surveys*, vol. 23, no. 1, pp. 5-48, 1991.
- [70] J. Paakkonen, N. Painiven, and P. Toivanen, "The choice of resolution level in lossy wavelet compression of color images", in *Proc. 9<sup>th</sup> International Conference on Signal Processing*, Beijing, China, Oct. 2008, pp. 814-819.
- [71] B. A. Olshausen and D. J. Field, "Natural image statistics and efficient coding," *Network: Computation in Neural Systems*, vol. 7, no. 2, pp. 333-339, July 1996.
- [72] R. W. Buccigrossi and E. P. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," GRASP Laboratory Technical Report 414, University of Pennsylvania, USA, 30 May 1997.
- [73] E. P. Simoncelli and R. W. Buccigrossi, "Embedded wavelet image compression based on a joint probability model," in *Proc. 4th Int. Conf. Image Processing*, Santa Barbara, CA, USA, Oct. 26-29, 1997.
- [74] R. W. Buccigrossi and E. P. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," *IEEE Trans. Image Processing*, vol. 8, no. 12, pp. 1688-1701, Dec. 1999.
- [75] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 11, no. 7, pp. 674-693, July 1989.
- [76] E. P. Simoncelli and E. H. Adelson, "Noise removal via bayesian wavelet coring," in *Proc. 3rd Int. Conf. Image Processing*, Lausanne, Switzerland, vol. 1, pp. 379-383, Sep. 1996.
- [77] S. Todd, G. G. Langdon Jr, and J. Rissanen, "Parameter reduction and context selection for compression of gray-scale images," *IBM J. Res. Develop.*, vol. 29, no. 2, pp. 188-193, Mar. 1985.
- [78] V. R. Algazi and R. R. Estes Jr, "Analysis based coding of image transform and subband coefficients," *SPIE Applications of Digital Image Processing XVIII*, vol. 2564, pp. 11-21, July 1995.
- [79] S. D. Stearns, "Arithmetic coding in lossless waveform compression," *IEEE Trans. Signal Processing*, vol. 43, no. 8, pp. 1874-1879, Aug. 1995.

- [80] A. Zandi, J. D. Allen, E. L. Schwartz, M. Boliek, U.S. Patent 6222941, Apr. 2001.
- [81] B. M. Oliver, "Efficient coding," *Bell Syst. Tech. J.*, vol. 31, no. 4, pp. 724-750, July 1952.
- [82] D. A. Huffman, "A method for the construction of minimum-redundancy codes," in *Proc. IRE*, vol. 40, no. 9, pp. 1098-1101, Sep. 1952.
- [83] E. N. Gilbert, and E. F. Moore, "Variable length binary encodings," *Bell Syst. Tech. J.*, vol. 38, no. 4, pp. 933-967, July 1959.
- [84] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression", *Commun. ACM*, vol. 30, no. 6, pp. 520-540, June 1987.
- [85] A. Moffat, R. M. Neal, and I. H. Witten., "Arithmetic coding revisited", *Proc. Data Compression Conf.*, Snowbird, UT, pp. 202-211, Mar. 1995.
- [86] A. Moffat, R. M. Neal, and I. H. Witten, "Arithmetic coding revisited," *ACM Trans. Inform. Syst.*, vol. 16, no. 3, pp. 256-294, July 1998.
- [87] J. L. Mitchell, and W. B. Pennebaker, "Software implementations of the Q-coder," *IBM J. Res. Develop.*, vol. 21, no. 6, pp. 753-774, Nov. 1988.
- [88] W. B. Pennebaker, J.L. Mitchell, G. G. Langdon, and R. B. Arps., "An overview of the basic principles of the Q-coder adaptive binary arithmetic coder," *IBM J. Res. Develop.*, vol. 32, no. 6, pp. 717-726, Nov. 1988.
- [89] W. B. Pennebaker, J. L. Mitchell, U.S. Patent 4933883, June 1990.
- [90] W. B. Pennebaker, J. L. Mitchell, U.S. Patent 4935882, June 1990.
- [91] L. Bottou, P. G. Howard, and Y. Bengio, "The Z-coder adaptive binary coder," in *Proc. Data Compression Conf.*, Snowbird, UT, pp. 13-22, Mar. 1998.
- [92] Y. Bengio, L. Bottou, and P. G. Howard, U.S. Patent 6188334, Feb. 2001.
- [93] Y. Bengio, L. Bottou, and P. G. Howard, U.S. Patent 6225925, May 2001.
- [94] Y. Bengio, L. Bottou, and P. G. Howard, U.S. Patent 6281817, Aug. 2001.
- [95] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 18-34, Feb. 1992.
- [96] F. Ono, M. Denki, and K. Kaisha, U.S. Patent 5059976, Oct. 1991
- [97] F. Ono, M. Denki, and K. Kaisha, U.S. Patent 5307062, April 1994.

- [98] G. N. N. Martin, "Range encoding: an algorithm for removing redundancy from a digitised message," *Proc. Video & Data Recording Conf.*, Southampton, UK, July 24-27, 1979.
- [99] A. Said and W. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Trans. Image Processing*, vol. 5, no. 9, pp. 1303-1310, Sep. 1996.
- [100] W. A. Pearlman, "High performance, low complexity image compression," *Applications of Digital Image Processing X, Proc. SPIE 3164*, pp. 234-246, July 1997.
- [101] M. Schindler "A fast renormalization for arithmetic coding," *Poster at DDC, Data Compression Conf.*, Snowbird, UT, Mar. 30 – Apr. 1, 1998.
- [102] <http://www.compressconsult.com/rangecoder/>
- [103] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ, 1993.
- [104] *tPad User Manual*, ver. 1.0.3, Terasic Technologies, 2010.
- [105] *DE2-115 User Manual*, ver. 1.0.3, Terasic Technologies, 2010.
- [106] *Cyclone IV Device Handbook – Volume 1*, ver. 1.8, Altera, February 2013.
- [107] *Avalon Interface Specifications*, ver. 11.0, Altera, May 2011.
- [108] *1/2.5-Inch 5Mp CMOS Digital Image Sensor MT9P031*, ver. F, Aptina, May 2011.
- [109] *MT9P031 Register Reference*, ver. A, Aptina, May 2011.
- [110] C. E. Cummings, "Synthesis and Scripting Techniques for Designing Multi-Asynchronous Clock Designs", Synopsys Users Group Conference, San Jose, CA, 2001.
- [111] V. Rajović, M. Prokin, V. Čeperković, D. Prokin , "An image codec with minimum memory size," *Proceedings of 2nd Mediterranean Conference on Embedded Computing (MECO)*, 2013, Budva, Montenegro, pp. 148-151, June 15-20, 2013, ISSN 1800-993X, DOI: 10.1109/MECO.2013.6601342
- [112] T. Price, N. Watson, J. Wilson, V. Rajović, D. Cussans, J. Goldstein, R. Head, S. Nash, R. Page, J. Velthuis, J. Strube, M. Stanitzki, P. Dauncey, R. Gao, A. Nomerotski, R. Coath, J. Crooks, R. Turchetta, M. Tyndel, S. Worm and Z. Zhang, "First radiation hardness results of the TeraPixel Active Calorimeter

- (TPAC) sensor,” *Journal of Instrumentation*, vol. 8, no. 1, pp. 1-5, 2013, ISSN 1748-0221, DOI: 10.1088/1748-0221/8/01/P01007
- [113] M. Stanitzki for Spider collaboration, “Advanced monolithic active pixel sensors for tracking, vertexing and calorimetry with full CMOS capability,” *Nuclear Instruments and Methods in Physics Research Section A*, vol. 650, no. 1, pp. 178-183, Sep. 11, 2011, ISSN 0168-9002, DOI: 10.1016/j.nima.2010.11.166
- [114] J. J. Velthuis, D. Cussans, V. Rajović, J. Goldstein, J. A. Wilson, S. D. Worm, R. E. Coath, J. P. Crooks, R. Page, P. D. Dauncey, R. Gao, R. Head, O. D. Miller, S. Nash, A. Nomerotski, T. Price, M. Stanitzki, J. Strube, R. Turchetta, M. Tyndel, N. K. Watson, Z. Zhang, “Beam test results of FORTIS, a 4T MAPS sensor with a signal-to-noise ratio exceeding 100,” *Journal of Instrumentation*, vol. 6, no. 12, pp. 2-10, Dec. 20, 2011, ISSN 1748-0221, DOI: 10.1088/1748-0221/6/12/P12006
- [115] J. A. Ballin, R. Coath, J. P. Crooks, P. D. Dauncey, A.-M. Magnan, Y. Mikami, O. D. Miller, M. Noy, V. Rajović, M. Stanitzki, K. D. Stefanov, R. Turchetta, M. Tyndel, E. G. Villani, N. K. Watson, J. A. Wilson, Z. Zhang, “Design and performance of a CMOS study sensor for a binary readout electromagnetic calorimeter,” *Journal of Instrumentation*, vol. 6, no. 5, pp. 1-36, 2011, ISSN 1748-0221, DOI: 10.1088/1748-0221/6/05/P05009
- [116] J. A. Ballin, J. P. Crooks, P. D. Dauncey, A.-M. Magnan, Y. Mikami, O. D. Miller, M. Noy, V. Rajović, M. Stanitzki, K. Stefanov, R. Turchetta, M. Tyndel, E. G. Villani, N. K. Watson, J. A. Wilson, “Monolithic Active Pixel Sensors (MAPS) in a Quadruple Well Technology for Nearly 100% Fill Factor and Full CMOS Pixels,” *Sensors* 2008, 8, 5336-5351, 2008, ISSN 1424-8220, DOI: 10.3390/s8095336
- [117] T. Price for **Spider collaboration**, “MAPS Technology for Vertexing, Tracking, and Calorimetry,” *Physics Procedia*, vol. 37, pp. 932–939, 2012, ISSN 1875-3892, DOI: 10.1016/j.phpro.2012.02.437
- [118] P. Dauncey for **Spider collaboration**, “Performance of CMOS sensors for a digital electromagnetic calorimeter,” *PoS(ICHEP 2010)502*, 2010, ISSN 1824-8039



- [119] J. A. Ballin, J. P. Crooks, P. D. Dauncey, B. Levin, M. Lynch, A.-M. Magnan, Y. Mikami, O. D. Miller, M. Noy, **V. Rajović**, M. Stanitzki, K. Stefanov, R. Turchetta, M. Tyndel, E. G. Villani, N. K. Watson, J. A. Wilson, “A MAPS-based readout for a Tera-Pixel electromagnetic calorimeter at the ILC,” *Nucl. Phys. B. Proc. Suppl.* 197, 342-345, 2009, ISSN 0920-5632, DOI: 10.1016/j.nuclphysbps:2009.10.099
- [120] J. P. Crooks, M. Stanitzki, K. D. Stefanov, R. Turchetta, M. Tyndel, E. G. Villani, Y. Mikami, O. Miller, **V. Rajović**, N. K. Watson, J. A. Wilson, J. A. Ballin, P. D. Dauncey, A.-M. Magnan, M. Noy, “A MAPS-based readout of an electromagnetic calorimeter for the ILC,” *J. Phys.: Conf. Ser.* 110 092035, 2008, ISSN 1742-6588, DOI: 10.1088/1742-6596/110/9/092035

# Biografija autora

Vladimir Rajović je rođen u Kosovskoj Mitrovici, 3. jula 1976. godine, gde je završio osnovnu školu i gimnaziju prirodno-matematičkog smera, obe sa odličnim uspehom.

Na Elektrotehničkom fakultetu u Beogradu je diplomirao 2001. godine, na odseku za Elektroniku, Telekomunikacije i Automatiku, sa srednjom ocenom tokom studija 9.56. Diplomski rad pod naslovom „Centralni regulator temperature u centralnom sistemu toplivodnog grejanja“ kod mentora prof. dr Dejana Živkovića je uspešno odbranio sa ocenom 10. Za diplomski rad je nagrađen drugom nagradom Matice Srpske za najbolji diplomski rad u oblasti tehničkih nauka.

U okviru magistarskih studija na Elektrotehničkom fakultetu u Beogradu položio je šest stručnih predmeta, i 2005. godine odbranio magistarsku tezu pod naslovom „Programabilni upravljački sistem za crkvene zvonike“, iz oblasti elektronike, kod mentora prof. dr Dejana Živkovića.

Tokom osnovnih studija kandidat je bio angažovan kao student saradnik na laboratorijskim vežbama iz osnovnih predmeta sa tadašnjeg smera za Elektroniku.

Nakon diplomiranja se zaposlio na Elektrotehničkom fakultetu Univerziteta u Prištini u zvanju asistenta pripravnika, da bi na Elektrotehnički fakultet Univerziteta u Beogradu prešao 2002. godine zadržavši isto zvanje. Unapređen je u zvanje asistenta 2007. godine.

Kandidat ima značajnu međunarodnu saradnju. Tokom 2007. godine proveo je sedam meseci u poseti Fakultetu za fiziku i astronomiju Univerziteta u Birminghamu, Velika Britanija. Posедуje zvanje počasnog naučnog savetnika (*Honorary Research Fellow*)

koje mu je 2007. godine dodelio Univerzitet u Birminghamu. Kontinuirano saraduje sa kolegama iz Velike Britanije, ranije na projektu MAPS, a trenutno na projektu SPIDER.

Kandidat je učestvovao na nekoliko inovacionih i projekata tehnološkog razvoja finansiranih od strane Ministarstva nauke Republike Srbije, nekoliko komercijalnih fakultetskih projekata kao i jednog Tempus projekta. Bio je rukovodilac fakultetskog projekta koji je rađen u saradnji sa Rutherford Appleton Laboratory, Velika Britanija.

Koautor je udžbenika

Milan Ponjavić, **Vladimir Rajović**, Lazar Karbunar, Osnovi digitalne elektronike – Zbirka rešenih zadataka, Beograd 2006. (ISBN 86-7466-258-7)

Kandidat je autor ili koautor sedam radova u međunarodnim časopisima, pet radova na međunarodnim konferencijama i tri rada u domaćim časopisima.

Iz oblasti doktorske disertacije, kandidat je autor ili koautor četiri rada u vrhunskim međunarodnim časopisima (kategorije M21), dva rada u istaknutim međunarodnim časopisima (kategorije M22) i pet radova na međunarodnim konferencijama (kategorije M33).

## **Radovi iz oblasti disertacije kategorije M21**

1. T. Price, N. Watson, J. Wilson, **V. Rajović**, D. Cussans, J. Goldstein, R. Head, S. Nash, R. Page, J. Velthuis, J. Strube, M. Stanitzki, P. Dauncey, R. Gao, A. Nomerotski, R. Coath, J. Crooks, R. Turchetta, M. Tyndel, S. Worm and Z. Zhang, "First radiation hardness results of the TeraPixel Active Calorimeter (TPAC) sensor," *Journal of Instrumentation*, vol. 8, no. 1, pp. 1-5, 2013, ISSN 1748-0221, DOI: 10.1088/1748-0221/8/01/P01007 (**IF=2.504**)
2. J. J. Velthuis, D. Cussans, **V. Rajović**, J. Goldstein, J. A. Wilson, S. D. Worm, R. E. Coath, J. P. Crooks, R. Page, P. D. Dauncey, R. Gao, R. Head, O. D. Miller, S. Nash, A. Nomerotski, T. Price, M. Stanitzki, J. Strube, R. Turchetta, M. Tyndel, N. K. Watson, Z. Zhang, "Beam test results of FORTIS, a 4T MAPS sensor with a signal-to-noise ratio exceeding 100," *Journal of Instrumentation*, vol. 6, no. 12, pp.

2-10, Dec. 20, 2011, ISSN 1748-0221, DOI: 10.1088/1748-0221/6/12/P12006  
(**IF=2.504**)

3. J. A. Ballin, R. Coath, J. P. Crooks, P. D. Dauncey, A-M. Magnan, Y. Mikami, O. D. Miller, M. Noy, **V. Rajović**, M. Stanitzki, K. D. Stefanov, R. Turchetta, M. Tyndel, E. G. Villani, N. K. Watson, J. A. Wilson, Z. Zhang, "Design and performance of a CMOS study sensor for a binary readout electromagnetic calorimeter," *Journal of Instrumentation*, vol. 6, no. 5, pp. 1-36, 2011, ISSN 1748-0221, DOI: 10.1088/1748-0221/6/05/P05009 (**IF=2.504**)
4. J. A. Ballin, J. P. Crooks, P. D. Dauncey, A.-M. Magnan, Y. Mikami, O. D. Miller, M. Noy, **V. Rajović**, M. Stanitzki, K. Stefanov, R. Turchetta, M. Tyndel, E. G. Villani, N. K. Watson, J. A. Wilson, "Monolithic Active Pixel Sensors (MAPS) in a Quadruple Well Technology for Nearly 100% Fill Factor and Full CMOS Pixels," *Sensors* 2008, 8, 5336-5351, 2008, ISSN 1424-8220, DOI: 10.3390/s8095336 (**IF=2.060**)

### **Radovi iz oblasti disertacije kategorije M22**

1. M. Stanitzki for **Spider collaboration**, "Advanced monolithic active pixel sensors for tracking, vertexing and calorimetry with full CMOS capability," *Nuclear Instruments and Methods in Physics Research Section A*, vol. 650, no. 1, pp. 178-183, Sep. 11, 2011, ISSN 0168-9002, DOI: 10.1016/j.nima.2010.11.166 (**IF=1.096**)

### **Radovi iz oblasti disertacije kategorije M23**

1. **V. Rajović**, G. Savić, V. Čeperković, M. Prokin, "Combined one-dimensional lowpass and highpass filters for subband transformer," *Electronics Letters*, vol. 49, no. 18, pp. 1150-1152, Aug. 29, 2013, ISSN 0013-5194, DOI: 10.1049/el.2013.0931 (**IF=0.950**)

## Radovi iz oblasti disertacije kategorije M33

1. **V. Rajović**, M. Prokin, V. Čeperković, D. Prokin , “An image codec with minimum memory size,” *Proceedings of 2nd Mediterranean Conference on Embedded Computing (MECO)*, 2013, Budva, Montenegro, pp. 148-151, June 15-20, 2013, ISSN 1800-993X, DOI: 10.1109/MECO.2013.6601342
2. T. Price for **Spider collaboration**, “MAPS Technology for Vertexing, Tracking, and Calorimetry,” *Physics Procedia*, vol. 37, pp. 932–939, 2012, ISSN 1875-3892, DOI: 10.1016/j.phpro.2012.02.437
3. P. Dauncey for **Spider collaboration**, “Performance of CMOS sensors for a digital electromagnetic calorimeter,” *PoS(ICHEP 2010)502*, 2010, ISSN 1824-8039
4. J. A. Ballin, J. P. Crooks, P. D. Dauncey, B. Levin, M. Lynch, A.-M. Magnan, Y. Mikami, O. D. Miller, M. Noy, **V. Rajović**, M. Stanitzki, K. Stefanov, R. Turchetta, M. Tyndel, E. G. Villani, N. K. Watson, J. A. Wilson, “A MAPS-based readout for a Tera-Pixel electromagnetic calorimeter at the ILC,” *Nucl. Phys. B. Proc. Suppl.* 197, 342-345, 2009, ISSN 0920-5632, DOI: 10.1016/j.nuclphysbps:2009.10.099
5. J. P. Crooks, M. Stanitzki, K. D. Stefanov, R. Turchetta, M. Tyndel, E. G. Villani, Y. Mikami, O. Miller, **V. Rajović**, N. K. Watson, J. A. Wilson, J. A. Ballin, P. D. Dauncey, A.-M. Magnan, M. Noy, “A MAPS-based readout of an electromagnetic calorimeter for the ILC,” *J. Phys.: Conf. Ser.* 110 092035, 2008, ISSN 1742-6588, DOI: 10.1088/1742-6596/110/9/092035

Прилог 1.

## Изјава о ауторству

Потписани-а Владимир М. Рајковић

број индекса \_\_\_\_\_

### Изјављујем

да је докторска дисертација под насловом

ХАРДВЕРСКА РЕАЛИЗАЦИЈА ЈЕДНОПРОЛАЗНОГ БРЗОГ КАДСКА СЪ  
ВИСОКИМ СТЕПЕНОМ КОМПРЕСИЈЕ И МИНИМАЛНИМ ЗАТЕЖЕНИМ РЕСУРСИМА

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, 30.01.2014.

Владимир Рајковић

Прилог 2.

## Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Владимир М. Радовић

Број индекса \_\_\_\_\_

Студијски програм ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Наслов рада ХАРДВЕРСКА РЕАЛИЗАЦИЈА ЈЕДНОПРАЗНОГ БРЗОГ КОДЕКА  
СА ВИСОКИМ СТЕПЕНОМ КОМПРЕСИЈЕ И МИНИМАЛНИМ ЗАПЕВАНИМ РЕСУРСИМА

Ментор ПРОФ. ДР Милош ПРОКИЋ

Потписани/а Владимир М. Радовић

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду**.

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, 30.01.2014.

Владимир Радовић

Прилог 3.

## Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

ХАРДВЕРСКА РЕАЛИЗАЦИЈА ЈЕДНОПРОЈАКТНОГ БРЗОГ КОДЕКА  
СА ВИСОКИМ СТЕПЕНОМ КОМПРЕСИЈЕ И МИНИМАЛНИМ ЗАХТЕВАНИМ РЕСУРСИМА

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство
2. Ауторство - некомерцијално
3. Ауторство – некомерцијално – без прераде
4. Ауторство – некомерцијално – делити под истим условима
5. Ауторство – без прераде
6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

Потпис докторанда

У Београду, 30.01.2014.

Владимир Рајковић