



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
Odsek za elektrotehniku i računarstvo
Institut za računarstvo i automatiku
Katedra za računarsku tehniku i računarske
komunikacije



**Jedno rešenje programskog rukovaoca za
upravljanje procesom dekodovanja i prikazom MPEG
video toka podataka**

- Diplomski - Master rad iz predmeta-
- Programska podrška u TV i obradi slike -

Mentor:
Prof. Dr. Nikola Teslić

Kandidat:
Tihomir Anđelić, e9814

Novi Sad, Novembar 2006.



KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj, RBR:	
Identifikacioni broj, IBR:	
Tip dokumentacije, TD:	Monografska publikacija
Tip zapisa, TZ:	Tekstualni štampani materijal
Vrsta rada, VR:	Diplomski - Master rad
Autor, AU:	Tihomir Anđelić
Mentor, MN:	Prof.Dr. Nikola Teslić
Naslov rada, NR:	Jedno rešenje DND rukovaoca pod Linux OS za digitalne TV prijemnike
Jezik publikacije, JP:	Srpski
Jezik izvoda, JI:	Srpski
Zemlja publikovanja, ZP:	Srbija
Uže geografsko područje, UGP:	Srbija, Vojvodina
Godina, GO:	2006
Izdavač, IZ:	FTN
Mesto i adresa, MA:	Novi Sad, Trg Dositeja Obradovića 6
Fizički opis rada, FO: (poglavlja/strana/citata/tabela/slika/grafika/priloga)	7/47/0/1/0/1
Naučna oblast, NO:	Elektrotehnika i računarska tehnika
Naučna disciplina, ND:	
Predmetna odrednica/Ključne reči, PO:	
UDK	
Čuva se, ČU:	U biblioteci FTN, Novi Sad, Trg Dositeja Obradovića 6
Važna napomena, VN:	
Izvod, IZ:	
Datum prihvatanja teme, DP:	
Datum odbrane, DO:	
Članovi komisije, KO:	Predsednik: Član: Član, mentor:
	Potpis mentora



KEY WORDS DOCUMENTATION

Accession number, ANO :		
Identification number, INO :		
Document type, DT :	Monographic's publication	
Type of record, TR :	Word printed record	
Contents code, CC :	Master work	
Author, AU :	Tihomir Anđelić	
Mentor, MN :	Ph. D.E.E. Nikola Teslić	
Title, TI :		
Language of text, LT :	Serbian	
Language of abstract, LA :	Serbian	
Country of publication, CP :	Serbia	
Locality of publication, LP :	Serbia, Vojvodina	
Publication year, PY :	2005	
Publisher, PB :	FTN	
Publication place, PP :	FTN	
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	7/47/0/1/0/1	
Scientific field, SF :	Electrical engineering and computer science	
Scientific discipline, SD :		
Subject/Key words, S/KW :		
UC		
Holding data, HD :	Library of FTN, Novi Sad, Trg Dositeja Obradovića 6	
Note, N :		
Abstract, AB :		
Accepted by the Scientific Board on, ASB :		
Defended on, DE :		
Defended Board, DB :	President:	
	Member:	Mentor's sign
	Member, Mentor:	

SKRAĆENICE

- API – Aplikativna programska sprega (eng. *Application Programming Interface*)
- ATSC – Digitalni TV standard za Severnu Ameriku (eng. *Advanced Television Systems Committee*)
- CPU – Centralni procesor (eng. *Central Processing Unit*)
- DTV – Digitalna televizija (eng. *Digital TV*)
- DVB – Emitovanje digitalnog videa (eng. *Digital Video Broadcast*)
- DVD – Digitalni video disk (eng. *Digital Video Disc*)
- DND – Dekoduj i prikaži (eng. *Decode and Display*)
- ES – Elementarni tok podataka (eng. *Elementary Streams*)
- GOP – Grupa slika (eng. *Group Of Pictures*)
- HD – Visoka rezolucija (eng. *High Definition*)
- HDTV – Televizija visoke rezolucije (eng. *High Definition TV*)
- HCUS – (eng. *Horizontal Color Up Sampling*)
- VCUS – (eng. *Vertical Color Up Sampling*)
- VD – Video dekodier (programska simulacija modula za izdvajanje i dekodovanje slike)
- OS – Operativni sistem
- Kbit/s – Hiljada bita u sekundi (eng. *Kilo bit per second*)
- Mbit/s – Miliona bita u sekundi (eng. *Mega bit per second*)
- MPEG – Standard za prenos video i audio signala za digitalnu televiziju (eng. *Moving Picture Experts Group*)
- MR – Rukovalac prikazom slike (programska simulacija modula za prikaz slike)
- PES – Paketski elementarni tok podataka (eng. *Packetised Elementary Stream*)
- PS – Programski tok podataka (eng. *Program Stream*)
- TS – Transportni tok za prenos podataka (eng. *Transport Stream*)
- SVCD – CD disk (ent. *Super Video Compact Disc*)
- SD – Standardna digitalna televizija (*Standard Definition television*)
- SDRAM– Sinhronizovana dinamička memorija sa slučajnim pristupom (eng. *Synchronous Dynamic Random Access Memory*)

SADRŽAJ

1.	Zadatak	1
2.	Teorijske osnove rada.....	2
2.1	Uvod u digitalnu televiziju	2
2.2	Arhitektura digitalnih TV prijemnika.....	2
2.3	Standardi za emitovanje digitalne televizije	3
2.4	MPEG-2 standard.....	4
2.5	Osnove MPEG-2 kompresije video podataka.....	4
2.6	Korisnički podaci	6
2.7	AFD deskriptor.....	6
2.8	Linux operativni sistem	7
3.	Analiza problema	9
3.1	Tok podataka u sistemu	9
3.2	Dekodovanje jednog toka video podataka.....	10
3.3	Dekodovanje dva toka video podataka.....	10
3.4	Kontrola dekodovanja.....	11
3.5	Arhitektura DND rukovaoca.....	12
3.6	Upravljanje memorijskim resursima	13
3.7	Rukovanje dekodovanjem dva toka video podataka	14
3.8	Skaliranje slike	15
3.9	Kompresija podataka	15
3.10	Kontrola prikaza.....	15
3.11	Rukovanje korisničkim podacima.....	16
3.12	Modovi rada DND rukovaoca.....	16
3.13	Rukovanje greškama	16
4.	Realizacija.....	17
4.1	Video dekodер	17
4.2	Programski modul za prikaz slike (MR).....	22
4.3	DND rukovaoc	24
4.3.1	DND sprežne funkcije.....	27
4.3.2	Rukovanje memorijskim baferima	28
4.3.3	Funkcije za obradu prekida	28
4.3.4	Upravljanje procesom dekodovanje.....	29
4.3.5	Rukovanje korisničkim podacima	30
4.3.6	Rukovanje greškama	30
5.	Testiranje i verifikacija.....	31
5.1	Test okruženje	31
5.2	Proces testiranja.....	32
5.3	Rezultati testiranja.....	33
6.	Zaključak	34
6.1	Pravci daljeg razvoja	34
7.	Literatura	36

1. Zadatak

Cilj diplomskog (master) rada je realizacija programskog rukovaoca za dekodovanje i prikaz MPEG video toka podataka (DND rukovaoc) za digitalne TV prijemnike u programskom jeziku C na Linux OS. Potrebno je realizovati programsku simulaciju modula za prikaz slike i modula za dekodovanje video podataka. DND je potpuno zatvoren programski modul i nema vidljivih programskih sprema prema aplikacionim slojevima sistema. Osnovni zadaci DND rukovaoca su :

- Upravljanje dekodovanjem slika od strane programskog video dekodera
- Upravljanje prikazom slike od strane programskog modula za prikaz slike
- Rukovanje video PTS/STC sinhronizacijom
- Rukovanje dekodovanjem jednog i dva toka podataka
- Realizacija obrade nastalih grešaka i podrška za oporavak od detektovanih grešaka
- Podrška osnovnih modova rada, inicijalna podrška za trik modove

DND rukovaoc treba da ima četiri osnovna stanja rada:

- Stop
Podaci se ne procesiraju, VD nije pokrenut, podaci nisu prosleđeni prema programskom modulu za prikaz slike
- Idle
Svi ulazni podaci se preskaču od strane VD, poslednja ispravna slika je prosleđena prema programskom modulu za prikaz slike
- Sync
Svi podaci se preskaču dok se ne detektuje zaglavlje sa sekvencom ili I slika. Posle toga se prelazi u stanje Play
- Play
Ulazni podaci se dekoduju i prikazuju u zavisnosti od izabranog DND moda rada.

Stanja rada DND rukovaoca zavise od tri ulazna parametra:

1. Broja dostupnih enkodovanih slika u ulaznom baferu
2. Broja slobodnih bafera za prikaz
3. Stanja VD (Spreman, Parsira, Dekoduje)

U sistemu treba da postoji samo jedan programski video dekod器和 stvarno paralelno dekodovanje se postiže vremenskom podelom procesa dekodovanja na dekodovanje dva zasebna toka podataka. Prebacivanje sa jednog na drugi aktivan tok podataka se dešava nakon dekodovanja svake slike. Iz tog razloga DND treba da bude implementiran kao jedna programska nit, ali koja radi sa dva nezavisna skupa parametara.

2. Teorijske osnove rada

2.1 Uvod u digitalnu televiziju

Razvojem informacionih tehnologija i usavršavanjem prenosa digitalnih podataka razvijaju se nove proizvodne tehnologije koje nalaze svoju primenu u svim oblastima tehnike. Ovaj razvoj nije zaobišao ni televizijsku industriju. Sredinom devedesetih godina prošlog veka razvijeni su standardi za prenos televizijskog signala u digitalnom obliku. Osnovne prednosti digitalnog prenosa televizijskog signala, u odnosu na analogni, su:

- Digitalni signal se može komprimovati, što dovodi do prenosa, približno, deset digitalnih kanala, umesto jednog analognog na istom frekventnom opsegu
- Digitalni signal se može prenositi raznim komunikacionim putevima (Ethernet, Internet, ...)
- Digitalni prenos daje mogućnost prenosa kvalitetnijeg zvuka i slike (televizija visoke rezolucije - HDTV)
- Velika otpornost na šum (postoje metode provere ispravnosti primljenih podataka)
- Proširenje usluga (izbor jezika prevoda i zvuka, interaktivni i multimedijalni sadržaji, pristup Internetu, ...)

Da bi se ovakav digitalan signal prikazao na ekranu i iz njega izdvojili i obradili svi dodatni podaci, TV prijemnici su transformisani iz analognih elektronskih sklopova u računarski sistem.

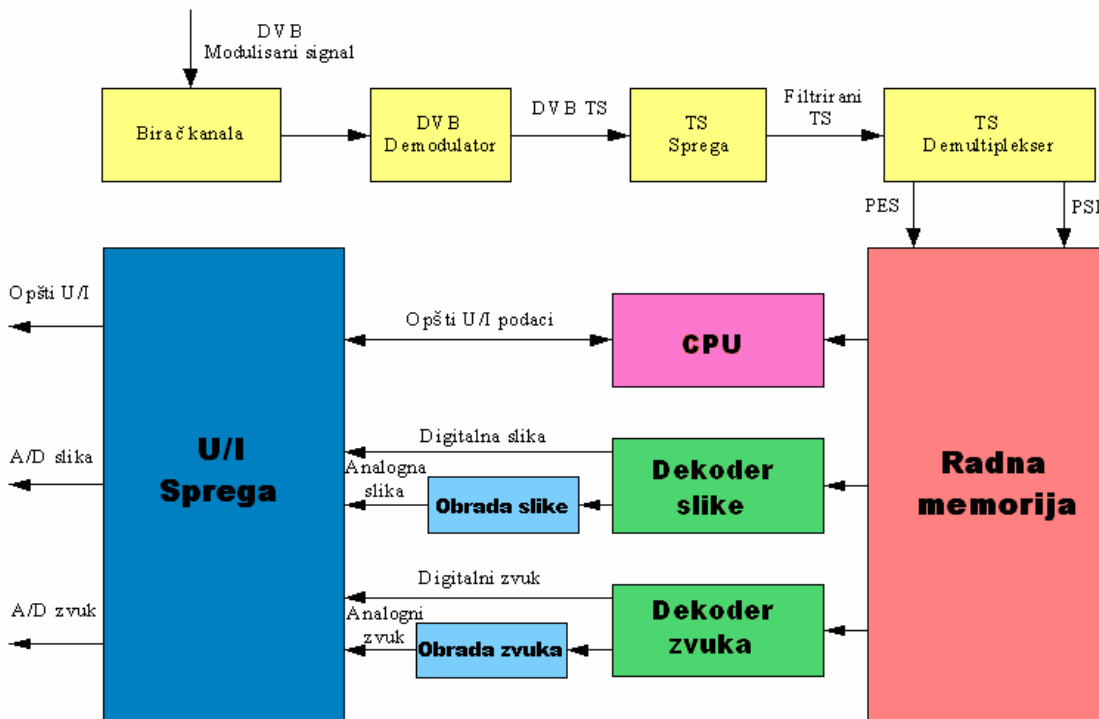
Prenos digitalnog signala se može obaviti putem raznih komunikacionih medija, kao što su radio veza, digitalna satelitska veza, kablovska TV mreža, zemaljska komunikaciona veza, mikrotalasne veze (wireless, mobile), DSL (ADSL). DVB standard propisuje nekoliko načina distribucije signala:

- DVB-S – DVB Satellite - definiše prenos DTV putem satelita
- DVB-C – DVB Cable - definiše prenos DTV putem kablovske mreže
- DVB-T – DVB Terrestrial- definiše zemaljski prenos putem VHF/UHF radio talasa
- DVB-H – DVB Handheld- definiše DTV prenos za prenosne uređaje (mobilni telefoni)

2.2 Arhitektura digitalnih TV prijemnika

Razvojna platforma svakog digitalnog TV uređaja (Slika 1) se sastoji od skupa, međusobno spregnutih, integrisanih kola, kojima upravlja centralni procesor. Ovakav sistem je složen i može da izvrši veliki broj operacija nad primljenim TV signalom. Modulisan signal stiže u prijemnik kroz birač kanala (eng. *tuner*). Dalje se vrši demodulacija i izdvaja se prenosni tok podataka (eng. *Transport Stream*). Iz njega se, u demultipleksu, izdvajaju paketski elementarni tokovi podataka (eng. *Packetised Elementary Stream*), kao i kontrolni podaci (eng. *Program Specific Information*), tabele servisnih informacija (eng. *Service Information tables*), privatni podaci itd. Izdvojeni podaci se smeštaju u radnu memoriju odakle se dalje obrađuju u procesoru, audio ili

video dekoderima. Osnova arhitekture ovog sistema je osnova bilo kog računarskog sistema (procesor, radna memorija i ulazno/izlazni podsistem). Centralni procesor upravlja procesima koji se izvršavaju u sistemu (izdvajanje podataka, dekodovanje, prikaz, ...) po unapred zadatom programu (programskoj podršci). Programska podrška se čuva u stalnoj memoriji (flash memorija).

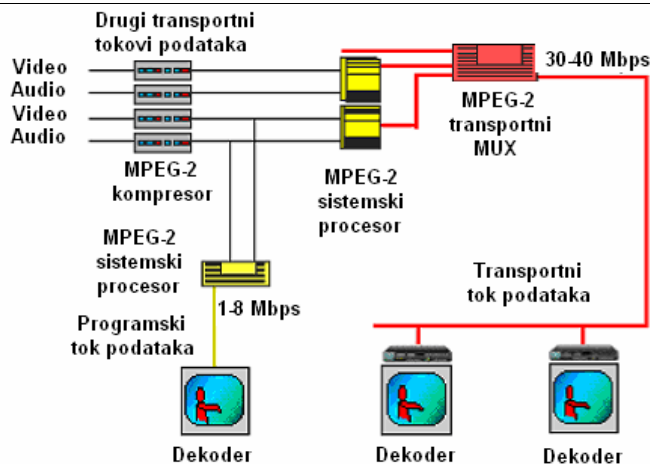


Slika 1 Arhitektura digitalnih TV prijemnika

2.3 Standardi za emitovanje digitalne televizije

Trenutno se u digitalnoj televiziji koristi MPEG-2 standard koji definiše algoritme za kodovanje i dekodovanje audio/video podataka. Tokovi MPEG2 audio i video podataka koji se emituju nazivaju se elementarni tok podataka (ES) i prenose se kao zasebni tokovi podataka. Kodovani podaci se prenose korištenjem paketski orijentisanih protokola u kombinaciji sa odgovarajućim mehanizmima za detekciju i obradu grešaka. Da bi se na prijemnoj strani ES mogli povezati u logičke celine (npr. koji audio ES pripada odgovarajućem video ES) kao i da bi se obezbedila odgovarajuća sinhronizacija audio i video podataka, pored ES moraju se prenositi i dodatni podaci. Dodatni podaci u sebi nose i druge informacije o TV uslugama kao što su naziv TV kanala, vreme emitovanja, raspored programa i slično.

Ovakvo kompleksna struktura prenosa podataka zahteva postojanje posebnog transportnog sloja/protokola koji se takođe definiše MPEG-2 standardom ali i odgovarajućim **Digital TV standardom**. U zavisnosti od prenosnog medijuma MPEG-2 standard definiše različite formate za kombinovanje MPEG-2 elementarnih tokova u jedinstven informacioni tok podataka. Ukoliko prenosni medijum nije podložan pojavi grešaka u toku prenosa, MPEG-2 elementarni tokovi se kombinuju u tzv. programski tok podataka (PS). Ovakav format omogućava jednostavnije programsko rukovanje podacima i koristi se za audio/video reprodukciju (npr. reprodukcija sa CD-a ili DVD-a) kao i u nekim mrežnim aplikacijama. Ukoliko je međutim prenosni put podložan pojavi grešaka (npr. eng. *broadcasting*), elementarni tokovi se kombinuju i prenose kao transportni tok podataka (TS). Sem toga TS format je pogodan za kombinovanje više TV programa u jedinstven informacioni tok. Ovaj proces kombinovanja MPEG-2 elementarnih tokova u jedinstveni informacioni tok u skladu sa definisanim formatom naziva se **multipleksiranje** MPEG-2 toka podataka. U digitalnoj televiziji se za prenos podataka koristi **MPEG-2 Transport Stream** (Slika 2).



Slika 2 Struktura MPEG-2 TS

U svetu se koriste različiti DTV standardi. Najzastupljeniji su *ATSC* standardi koji se koriste u Severnoj Americi i grupa *DVB* standarda koji se koriste kako u Americi tako i u ostatku sveta.

2.4 MPEG-2 standard

MPEG je standard za kodovanje i kompresiju digitalnih audio i video podataka objavljen od strane MPEG (eng. *Moving Picture Experts Group*) organizacije. MPEG-2 je nastavak osnovnog MPEG sistema sa proširenjima za kompresiju i prenos digitalnih TV video podataka. Modifikovani MPEG-2 se kao standard takođe koristi za kodovanje komercijalnih DVD filmova. Korišćenje MPEG-2 standarda nije besplatno i podleže plaćanju MPEG licence. U okviru MPEG standarda nije definisan proces kodovanja slike već sintaksa kodovanog toka podataka i proces dekodovanja.

MPEG-2 uključuje dva sistematska dela koja definišu dva odvojena (ali srodna) formata. Jedan je TS (eng. *Transport Stream*) koji se obično koristi za emitovanje digitalnih podataka, kao što su ATSC i DVB. Standard takođe definiše i programski tok podataka, format podržava emitovanje podataka sa optičkih i drugih memorijskih medija kao što je čvrsti disk. MPEG-2 programski tok podataka se koristi za DVD i SVCD standarde.

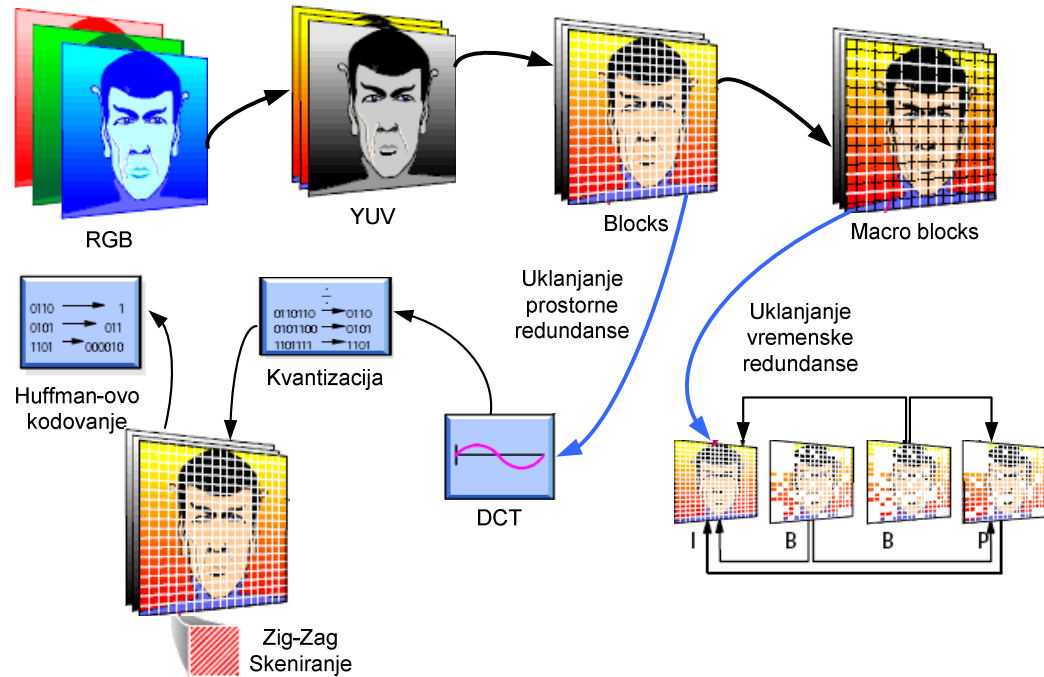
Video deo MPEG-2 standarda je sličan MPEG-1 standardu, ali uključuje i podršku za isprepleten (eng. *interlaced*) video (format slike koj se koristi u analognim TV sistemima). MPEG-2 video nije optimizovan za male bitske brzine (manje od 1 Mbit/s), ali prevazilazi MPEG-1 na 3 Mbit/s i više. Svi standardi zasnovani na MPEG-2 video standardu su potpuno komatibilni sa MPEG-1 standardom. Sa nekim unapređenjima MPEG-2 se takođe koristi i u većini HDTV sistema.

2.5 Osnove MPEG-2 kompresije video podataka

MPEG-2 generički koduje pokretne slike i prpratni zvuk i specifira format toka video podataka koji mogu da se rekonstruišu od tri tipa frejmova (eng. *intra frames*, *forward predicted frames* and *bidirectionally predicted frames*) koji mogu biti složeni u specifične nizove koji se nazivaju GOP strukture. Standard podržava *interlaced* i *progressive* video podatke. U *progressive* video podacima osnovna jedinica enkodovanja je frejm dok u *interlaced* podacima osnovna jedinica može da bude polje ili frejm. MPEG-2 ima tri načina enkodovanja slika: intra kodovana slika (I slike), prediktovana slika od prethodne I ili P slike (P slike) i bi-direkciono prediktovna (B slike).

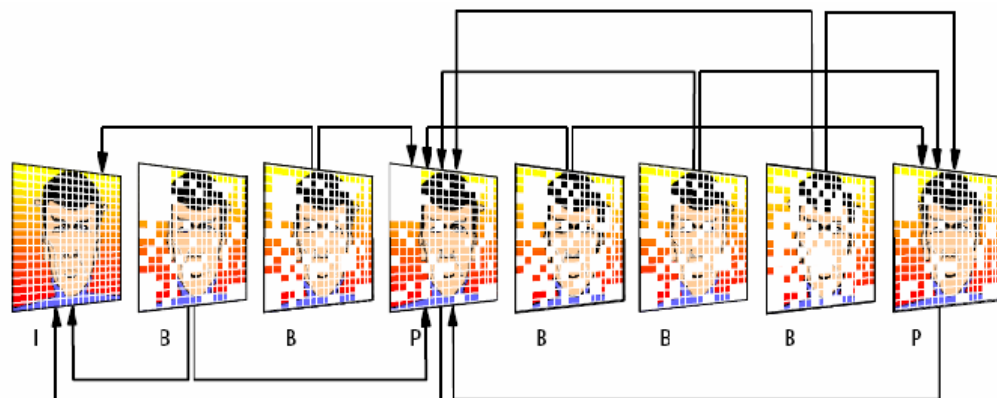
Video slika je podeljena na jedan luma (Y) i dva hroma (Cb i Cr komponente) kanala. Blokovi luma i hroma nizova su organizovani u makroblokove, koji su osnovna jedinica kodovanja slike. Svaki makro blok je podeljen u četiri 8x8 luma bloka. Broj 8x8 hroma blokova po jednom makro bloku zavisi od hroma formata izvorne slike. Na primer, u opštem 4:2:0

formatu, ima jedan hroma blok po jednom makro bloku za svaki od dva hroma kanala, što ukupno čini šest blokova po jednom makro bloku.



Slika 3 MPEG kompresija videa

Za slučaj I slike, stvarni podaci se enkoduju u prenose u okviru transportnog toka podataka. P i B slike enkoduju tako što se određuje vektor pomeraja (eng. *motion estimation*) na osnovu sličnosti sa prethodnom referentnom slikom (za slučaj B slike, i sa sledećom referentnom slikom) u vremenskom redosledu. Za svaki makro blok, enkoder traži dobar referentni odmerak sa kojim je najbliži u prethodnoj i/ili sledećoj slici. Vektor pomeraja je enkodovan tako da opiše odnos između trenutnih makro blokova i referentnih odmeraka iz kojih je prediktovan. Obično, nakon određivanja vektora pomeraja još uvek postoje razlike između makro bloka i referentnog odmerka. Ako je razlika veoma mala verovatno ne može da se uoči tokom gledanja i enkoder neće enkodovati taj makro blok. U tom slučaju, ulazni makro blok je opisan samo sa dva broja: dve komponente vektora pomeraja.



- I-slika: Sadrži informaciju o celoj slici
- P-slika: Prediktovana slika od prethodne I ili P
- B-slika: Koristi prethodnu i sledeću sliku

Slika 4 Redosled enkodovanja slika

Svaki blok se transformiše sa 8x8 diskretnom kosinusnom transformacijom. Dobijeni DCT koeficijenti se kvantizuju i preslože tako da se može primeniti run-length kodovanje. Konačno se proces završava primenom hafmanovog kodovanja sa fiksnom tabelom. Ovim se postiže da usled velike sličnosti uzastopnih slika P slike mogu da budu veličine 10% od veličine I slike, i B slike oko 2% veličine.

Sekvenca različitih tipova slika, počinje sa I slikom i završava se pre sledeće I slike i naziva se GOP (eng. *Group of Pictures*). Broj i tip slika u okviru jednog GOP-a nije određen ali se preporučuje da jedan GOP sadrži 15 frejmova i da redosled slika bude I_BB_P_BB_P_BB_P_BB_P_BB_. Za najefikasnije kodovanje postavljanje I, P i B slika u okviru jedne GOP strukture treba da bude određena prirodom toka ulaznih video podataka. Scene sa većim brojem pokreta i većom dinamikom se enkoduju sa sa manjim brojem B slika. Ovaj slučaj je veoma čest u realnim sistemima koji imaju ograničene fizičke resurse. Izlazna bitska brzina MPEG-2 enkodera može da bude konstantne ili promenljive dužine sa maksimalnom gornjom dozvoljenom granicom.

2.6 Korisnički podaci

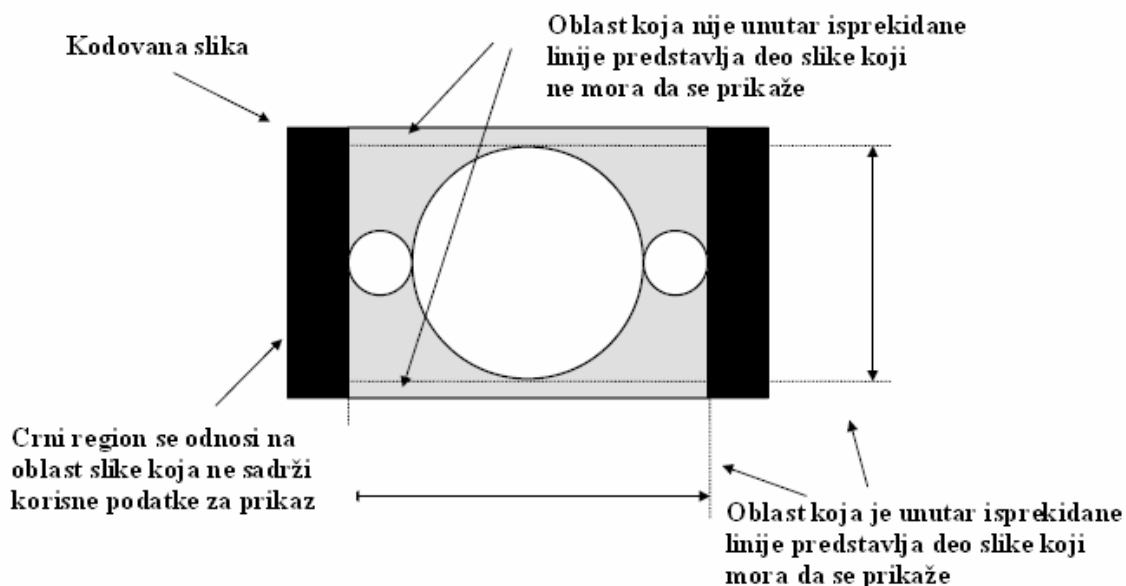
Korisnički podaci (eng. *user data*) su podaci koji nemaju jasno definisane parametre u okviru MPEG-2 standarda. Nije definisana dužina podataka koji mogu da se prenesu ni tip podataka koji se prenose. Korisnički podaci mogu da se uključe na tri mesta u okviru toka podataka: na nivou sekvence, nivou grupe slika (GOP) i na nivou slike. Aplikacija koja obrađuje MPEG podatke ne mora da obrađuje korisničke podatke ali mora da bude u stanju da ih sačuva.

Informacije o formatu slike mogu biti sačuvane u okviru korisničkih podataka. DVB i ATSC standardi koriste AFD deskriptore za čuvanje podataka o preporučenom prikazu slike.

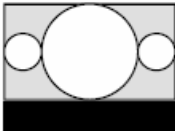
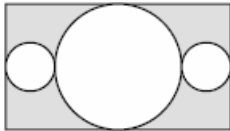
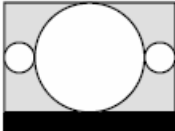
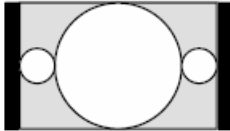
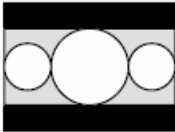
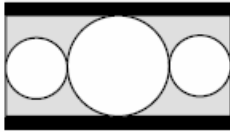
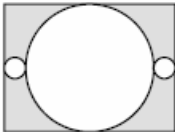
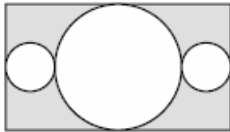
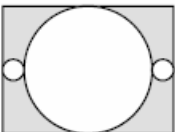
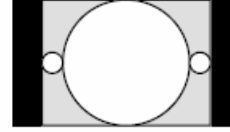
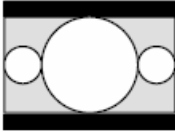
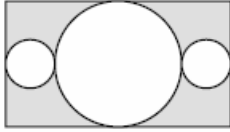
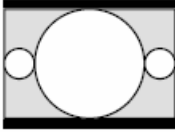
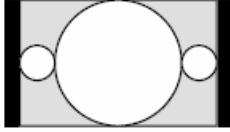
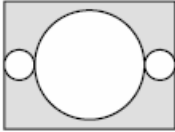
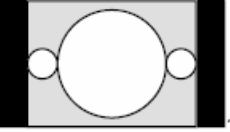
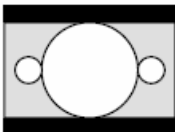
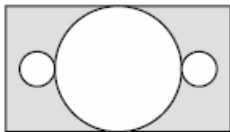
2.7 AFD deskriptor

AFD deskriptor se prenosi u okviru korisničkih podataka. Nakon početka svake sekvence (i ponovljene sekvence) u zaglavlju sekvence se signaliziraju preporučeni format slike i ostali parametri bitni za prikaz. Primljeni AFD deskriptor ima vreme važenja do prijema sledeće sekvence ili dok se ne primi drugi AFD.

AFD deskriptor može biti postavljen na svakom mestu gde se nalaze korisnički podaci u okviru ES (nakon zaglavlja sekvence, GOP zaglavlja ili zaglavlja slike). AFD može da se menja za svaku sliku koja se prenosi i samo poslednji primljeni je validan. Kombinovanje dobijene vrednosti za AFD i aktivnog formata određuje oblast od "interesa" za prikaz. Sledeće slike prikazuju interpretaciju AFD deskriptora:



Slika 5 Delovi slike od interesa za prikaz

AFD		Interpretacija	
vrednost	opis	4:3 kodovana slika	16:9 kodovana slika
0000-0001	rezervisano		
0010	box 16:9 (top)		
0011	box 14:9 (top)		
0100	box > 16:9 (centre)		
0101-0111	rezervisano		
1000	Kao kodovana slika		
1001	4:3 (centre)		
1010	16:9 (centre)		
1011	14:9 (centre)		
1100	rezervisano		
1101	4:3 "shut and protect" 14:9 centre		
1110	16:9 "shut and protect" 14:9 centre		

Slika 6 Interpretacija AFD deskriptora

2.8 Linux operativni sistem

Linux je operativni sisem koji se uglavnom koristi na personalnim računarima, ali je prisutan i na serverima i mnogim prenosnim uređajuma. Linux je delo studenta računarstva Linusa Torvaldsa. Nastao je 1991. godine kao posledica Linusovog hobija, u nadi se da će stvoriti robusnu distribuciju UNIX-a za Minix korisnike. Operativni sistem Minix je napisao profesor računarskih nauka Andrew Tannebaurn da bi pokazao nekoliko naučnih koncepata o

operativnim sistemima. Torvalds je te koncepte objedinio u samostalan operativni sistem nalik UNIX-u. Do programa su mogli doći studenti računarstva širom sveta, tako da su ubrzo nastale zasebne razvojne grupe. Operativni sistem su razvijali i gradili programeri iz čitavog sveta. Njihov cilj je bio razvijati operativni sistem bez bilo kakvog komercijalnog softvera koji se koristi u svetlu.

Linux operativni sistem je zasnovan na GNU alatima (eng. *Gnu's Not UNIX*), koji obezbeđuju skup standardnih načina rukovanja i korišćenja operativnog sistema. Svi GNU alati su otvorenog koda, tako da mogu biti instalirani na bilo koji sistem. Mnoge distribucije nude pakete najčešće korišćenih alata, kao što su RPM paketi, tako da korisnik ne mora biti programer da bi mogao instalirati paket na svoj sistem. Linux se koristi u okviru GPL dozvole (eng. *GNU Public Licence*). Ova dozvola omogućuje svakom korisniku da menja Linux i eventualno da izvrši distribuciju promenjene verzije, pod jednim uslovom da je omogućen uvid u kod posle distribucije.

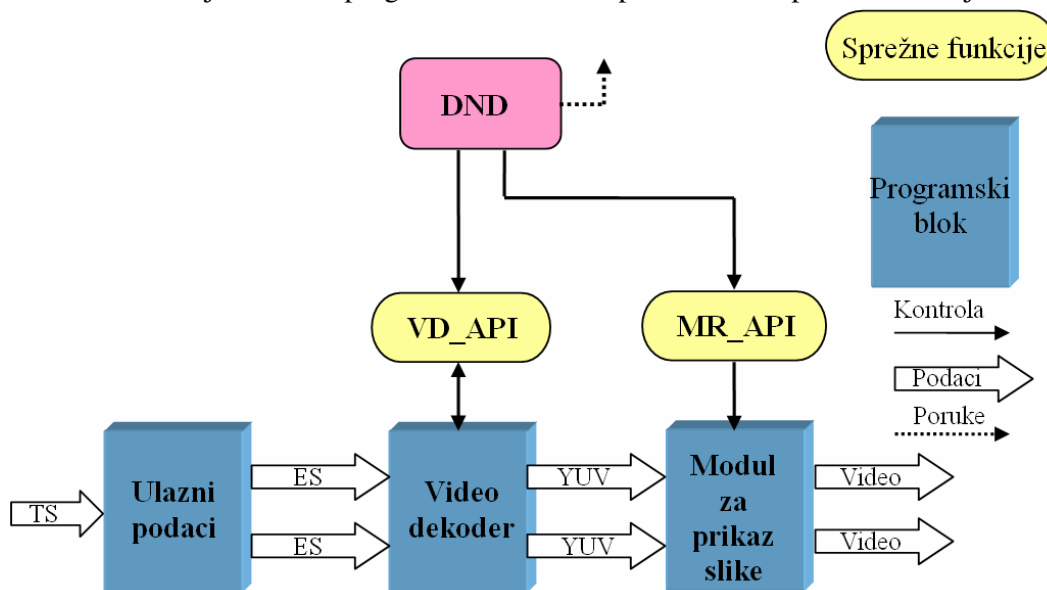
Osnovna komponenta ili najniži programski sloj Linux operativnog sistema predstavlja jezgro (eng. *kernel*). On omogućava spregu sa fizičkom arhitekturom i upravlja rukovaocima fizičke arhitekture. Takođe kernel upravlja svim fizičkim resursima sistema: memorijom, sistemom podataka (eng. *filesystem*), mrežom, procesima i dozvolama pristupa.

Linux omogućava tekstualni oblik sprege sa korisnikom. Korisnik iz komandne linije (eng. *command line*) zadaje instrukcije. Softverski sloj koji ovo omogućava sa naziva školjka (eng. *shell*). On se sastoji iz komandnog interpretera koji pokreće aplikacije i skupa komandi ili malih programa koji omogućavaju pretragu i editovanje datoteka, kontrolu izvršavanja programa, prikaz sistemskih informacija, itd.

3. Analiza problema

3.1 Tok podataka u sistemu

Sa aspekta razmene poruka u odnosu na DND rukovaoc sistem se grubo može podeliti u tri celine (Slika 7). Prvu celinu čini DND rukovaoc koji posredstvom odgovarajućih sprežnih funkcija komunicira i upravlja sa ostatkom sistema. Drugu celinu čine dva skupa sprežnih funkcija koje služe kao sprega programskih blokova sistema i viših programskih slojeva. Treća celina su programski blokovi sistema za simulaciju fizičkih jedinica koji interno razmenjuju podatke i komuniciraju sa višim programskim nivoima posredstvom sprežnih funkcija.



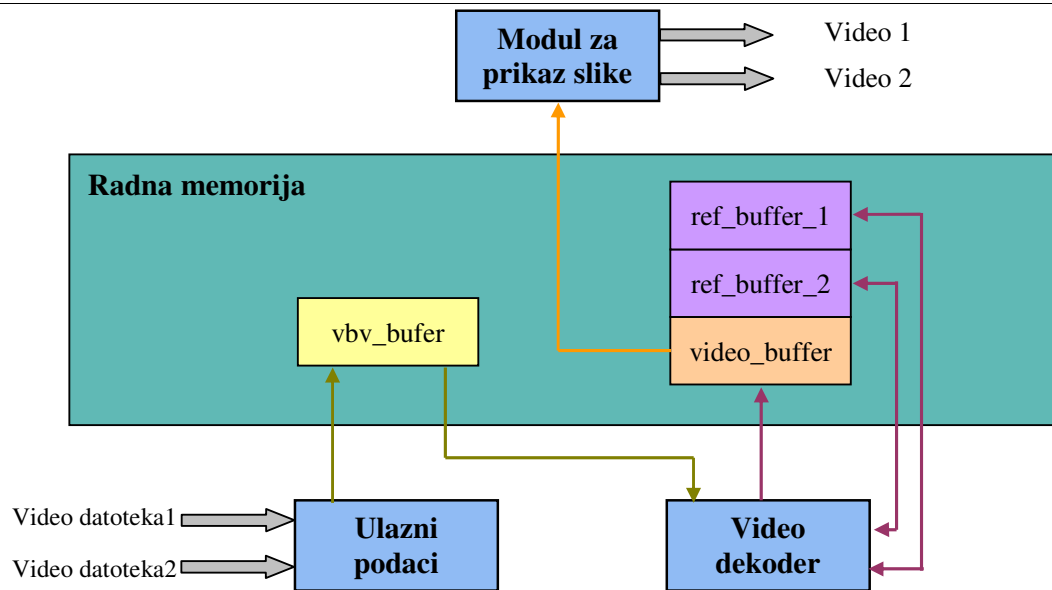
Slika 7 Razmena podataka u sistemu

Tok podataka sa aspekta korišćenja memorijskih resursa predstavljen je na slici 13. DND rukovaoc upravlja radom modula za prijem ulaznih podataka, video dekodierom i modulom za prikaz slike.

Modul za prijem podataka se koristi kao ulazna sprega za bilo koju vrstu MPEG video toka podataka. Kao ulazni tok podataka može da se koristi video datoteka ili podaci sa TV kartice.

Video dekodier (eng. *Video Decoder*) modul dekoduje MPEG video podatke visoke rezolucije (HD) ili standardne (SD) rezolucije sa podrškom za dekodovanje u paraleli. Izdekodovane slike se čuvaju u odgovarajućim video baferima.

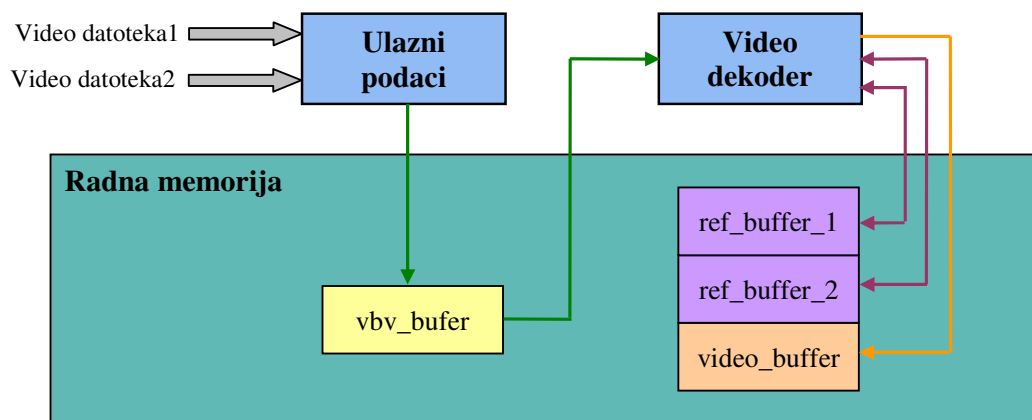
Osnovni zadatak modula za prikaz slike je konverzija MPEG videa u YUV 4:2:0 i YUV 4:4:4 format i obezbeđivanje odgovarajućih signala za prikaz i obradu slike. Svi relevantni parametri koji se odnose na sliku su programibilni.



Slika 8 Tok podataka sa aspekta korištenja radne memorije

3.2 Dekodovanje jednog toka video podataka

Modul za prijem ulaznih podataka služi kao sprega sa ulaznim video tokovima podataka iz video datoteke ili TV kartice. Podržan je prijem dva video toka podataka u paraleli s tim da se za slučaj dekodovanja jednog toka podataka drugi tok podataka ne obrađuje. Primljeni podaci se dalje prosleđuju u bafer za smeštanje video sadržaja.



Slika 9 Proces dekodovanja jednog toka video podataka

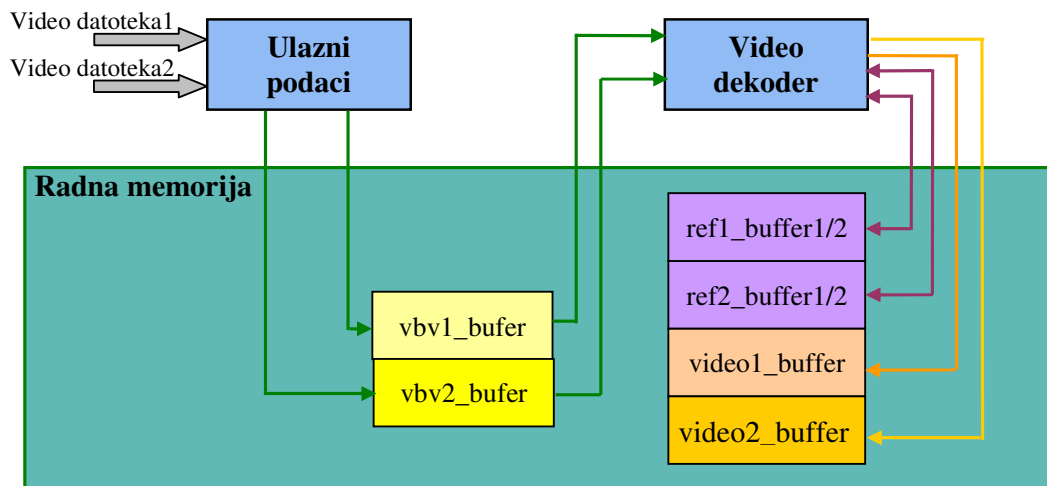
MPEG video tok podataka je spakovan kao paketizovani tok video podataka (PES) u TS paketima. Programski video dekodier odvaja PES zaglavlja iz korisnog sadržaja i upisuje ih u bafer za smeštanje video podataka (*vbw_bufer*). Bafer za smeštanje je cikličan i pokazivač za upis sadržaja se vraća na početak bafera kada dođe do njegovog kraja.

VD modul čita kompresovane video podatke iz *vbw_bufer* i vrši njihovo dekodovanje u 4:2:0 format. U procesu dekodovanja VD pristupa podacima u nekoliko bafera, dva za smeštanje referentnih slika (*ref_buffer_1*, *ref_buffer_2*) i baferu za smeštanje izdekodovanih slika (*video_bufer*). Ovi baferi se dinamički izmenjuju zavisno od tipa slike koja se dekoduje.

3.3 Dekodovanje dva toka video podataka

Za potrebe konkurentnog dekodovanja dva nezavisna TS izvora, modul za prijem ulaznih podataka spaja dva izlazna TS. TS se spajaju zavisno od bitske brzine dolaznih paketa. Tako

spojeni TS se smeštaju u odvojene bafere u radnoj memoriji. Broj odredišnog bafera je određen svojim PID-om i identifikatorom TS-a.



Slika 10 Proces dekodovanja dva toka video podataka u paraleli

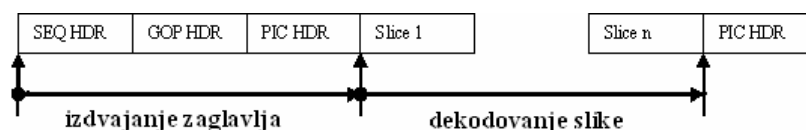
U ovom slučaju dekodovanja koriste se odvojeni video bafere za smeštanje podataka za svaki ulazni TS (*vbv_buffer1*, *vbv_buffer2*). Smeštanje izdekodovanih slika u oba video bafere se vrši sekvencijalno. VD modul mora dodatno da postavlja bafere za smeštanje referentnih slika. Prilikom izmene aktivnog toka podataka za dekodovanje potrebno je sačuvati kontekst prethodno dekodovanog toka podataka svaki put kada se promeni odredišni video bafer.

3.4 Kontrola dekodovanja

Video dekodier dekoduje sliku po sliku i iz tog razloga je potrebno izmeniti odredišni bafer nakon svake izdekodovane slike. Da bi se odredila pozicija slike u ulaznom video bafer, šalju se MPEG start kodove slike (00 00 00 01) zajedno sa njihovom adresom DND rukovaocu. Naredni MPEG start kodovi mogu biti takođe sačuvani da bi se omogućila analiza ostalih polja MPEG zaglavlja.

Za sinhronizaciju između audio i video podataka neophodni su PTS kodovi. Kada je PTS kod prisutan vrši se njegovo izdvajanje i izdvojena vrednost se prosleđuje DND rukovaocu. PTS se dodeljuje za svaku MPEG sliku. U ovom konceptu svaka slika mora biti obeležena sa svojim PTS kodom, ali PTS se ne šalje za svaku sliku. U tom slučaju programski se interpolira PTS na osnovu vrednosti poslednjeg primljenog PTS koda.

Prilikom inicijalizacije procesa dekodovanja neophodno je da bude detektovan jedan start kod za I sliku i start kod za sledeću sliku pre nego što se pokrene VD modul. Iz tog razloga je potrebno da se u memoriji nalazi smeštena jedna cela I slika. U principu VD se pokreće nakon prijema podataka u *vbv_buffer* ali se inicijalno preskaču sve slike dok se ne primi ispravna vrednost MPEG zaglavlja sekvence (Slike 12), odnosno dok se ne pronađe prva I slika.



Slika 11 Proces dekodovanja

Proces dekodovanja slike može biti startovan samo ako su zadovoljeni sledeći uslovi:

- izdvojena su odgovarajuća zaglavlja
- video dekodier se nalazi u stanju spreman
- poslednja cela slika je dostupna u ulaznom baferu
- dostupan je bafer u koji se dekoduje slika

Za slučaj kada se dekoduju dva ulazna toka podataka prebacivanje sa jednog na drugi tok podataka za dekodovanje može se uraditi samo nakon što je izdekodovana prethodna slika a pre slanja komande za izdvajanje slike.

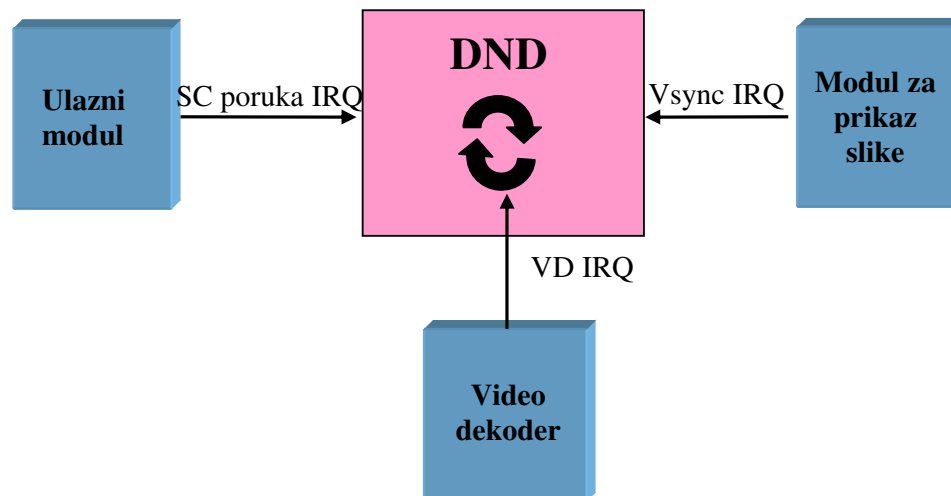
3.5 Arhitektura DND rukovaoca

DND rukovaoc treba da bude realizovan kao programska nit koja u sebi sadrži automat sa konačnim brojem stanja. Zavisno od stanja u kome se nalazi i primljenih poruka od strane sprežnih funkcija programskih blokova rukovaoc treba da upravlja radom sistema (Slika 13). Kao odgovor na primljene poruke neophodno je da se implementiraju programske procedure za obradu prekidnih poruka. Potrebne su tri programske procedure: za obradu prekida primljenih od jedinice za prijem video toka podataka (ulazni modul), za obradu prekida od modula za prikaz slike i procedura za obradu prekida primljenih od video dekodek jedinice.

Za slučaj kada se dekoduju dva ulazna toka podataka potrebno je inicijalizovati dva skupa struktura nad kojima radi nit DND rukovaoca.

Poruke koje se primaju od strane jedinice za prijem video toka podataka prvenstveno služe da se detektuju greške u video toku podata. Proces detekcije grešaka na ovom nivou se zasniva na poređenju izdvojenih parametara zaglavlja primljenih od strane ulaznog i video dekodek modula. Greške koje mogu da se detektuju se manifestuju u prijemu različitih vrednosti za parametre zaglavlja slika koje se prenose a neophodni su za ispravno dekodovanje. Parametri iz odgovarajućih zaglavlja koji se posmatraju su:

- tip slike koja se dekoduje (I, P ili B slika)
- tip slike ("frame" ili "field")
- trenutna referenca (eng. *Temporal reference*)
- PTS vrednost za sliku
- trenutna bitska brzina



Slika 12 Arhitektura DND rukovaoca

Poruke koje se razmenjuju između DND rukovaoca i video dekodek bloka obuhvataju komandne poruke za dekodovanje i izdvajanje slike koje šalje rukovaoc i poruke koje šalje video dekodek kao potvrdu da je operacija uspešno obavljena ili da je nastupila greška u toku dekodovanja.

Modul za prikaz slike prima poruke neophodne za prikaz. Parametri koji se prosleđuju MR bloku sadrže informacije o lokaciji slike, rezoluciji, trenutnoj bitskoj brzini, formatu slike i druge parametre bitne za odgovarajući prikaz.

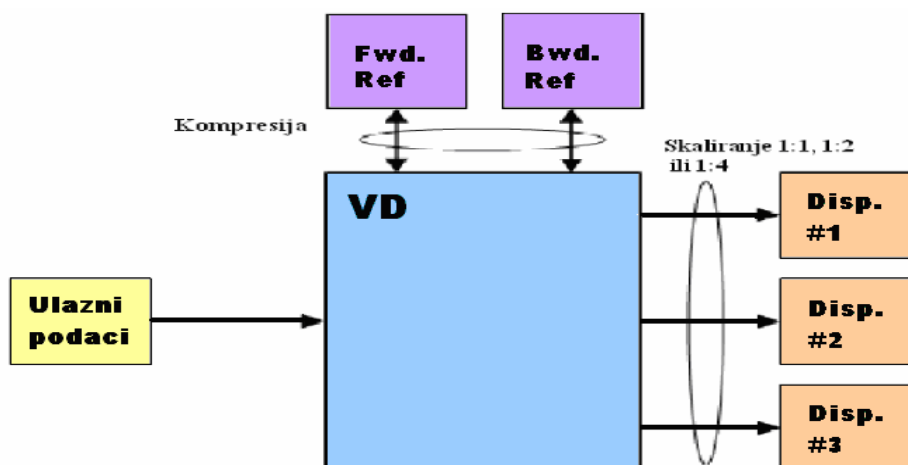
DND automat može da se nalazi u četiri stanja:

- **Stop**
Podaci se ne procesiraju, VD nije pokrenut, podaci nisu prosleđeni prema modulu za prikaz slike.
- **Idle**
Svi ulazni podaci se preskaču od strane VD, poslednja ispravna slika je prosleđena prema modulu za prikaz slike.
- **Sync**
Svi podaci se preskaču dok se ne detektuje zaglavlje sa sekvencom ili I slika. Posle toga se prelazi u stanje Play.
- **Play**
Ulazni podaci se dekoduju i prikazuju u zavisnosti od izabranog DND moda rada.

3.6 Upravljanje memorijskim resursima

Za smeštanje izdekodovanih slika, na nivou DND rukovaoce, koristi se "2+3" organizacija bafera (Slika 14). Za smeštanje i čuvanje referentnih slika koriste se dva fiksna bafera (po jedan za svaku referentnu sliku) i tri bafera za smeštanje slika za prikaz.

Svrha ovakve organizacije bafera je da se omogući istovremeno dekodovanje i prikaz slike (eng. *Decode During Display*) tako što video dekodler može da dekoduje sliku u isti bafer kome pristupa i jedinica za prikaz slike. Baferi za smeštanje referentnih slika se pune nakon dekodovanja svake I ili P slike. Punjenje bafera za prikaz se vrši po algoritmu "sledeći slobodan" (slika se upisuje u prvi slobodan bafer). Rezultat ovakvog rukovanja baferima obezbeđuje nezavisnost od trenutnog broja korištenih bafera za prikaz. Za slučaj kada se dekoduju dva toka podataka u paraleli svi navedeni baferi su udvojeni.

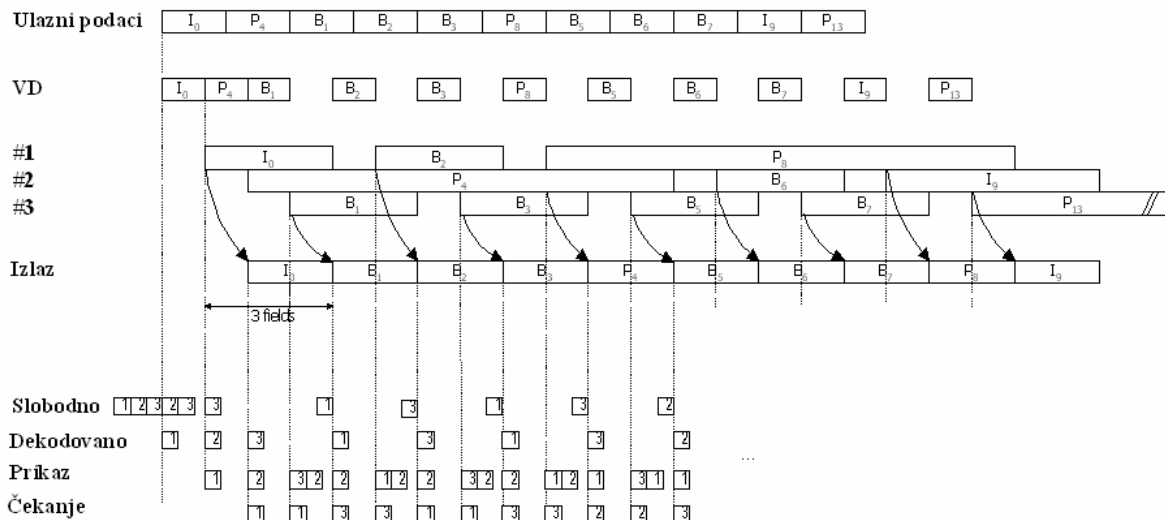


Slika 13 Memorijski resursi video dekodera

Za rukovanje baferima za smeštanje slika za prikaz koriste se četiri jednostruko spregnute liste:

1. Lista slobodnih bafera (eng. *Free List*); sadrži trenutno slobodne bafere
2. Lista za dekodovanje (eng. *Decode List*); može da ima samo jedan element i sadrži određeni bafer u koji se trenutno dekoduje slika
3. Lista bafera za prikaz (eng. *Display List*); sadrži bafere sa izdekodovanim slikama koje su spremne za prikaz. Ova lista treba da bude sortirana na osnovu trenutne reference (eng. *temporal reference*)
4. Lista čekanja (eng. *Delay List*); može da sadrži samo jedan element i sadrži bafer koji se trenutno čita od strane MR bloka

Sortiranje liste bafera za prikaz vrši se na osnovu trenutne reference (eng. *temporal reference*) prilikom uvezivanja elementa iz liste dekodovanih bafera u listu za prikaz. Sledeća slika (Slika 15) prikazuje način rukovanja elementima navedenih listi.



Slika 14 Rukovanje listom bafera

Gornja polovina slike prikazuje redosled slika u ulaznom video toku podataka, zatim trenutno dekodovane slike od strane video dekodera i redosled upisa slika u baferu za prikaz. Donja polovina slike prikazuje četiri jednostruko spregnute liste gde se broj odnosi na redni broj bafera za prikaz, tako da može da se vidi redosled povezivanja elemenata u navedene liste.

3.7 Rukovanje dekodovanjem dva toka video podataka

Kao što je ranije navedeno DND rukovaoc treba da bude implementiran kao jedna programska nit. Za slučaj dekodovanja dva toka podataka nit DND rukovaoca radi nad dva skupa parametara, za svaki ulazni tok podataka po jedan, pri čemu se vodi računa o trenutno aktivnom toku podataka (informacija o trenutno aktivnom skupu se čuva u globalnoj promenljivoj). U principu DND rukovaoc uvek proverava ulazno stanje oba toka podataka i dekoduje onaj koji je aktivan. Globalna promenljiva u kojoj se čuva informacija o trenutno aktivnom toku podataka za dekodovanje odnosi se na to koji će tok podataka prvi da se proverava.

Na ovaj način ne dolazi do međusobnog uticaja dekodovanja dva toka video podataka i ako je jedan u stanju neaktivan drugi se dekoduje bez prekida. U isto vreme, u normalnim uslovima, slike mogu da se dekoduju i kada imaju različitu frekvenciju.

Promena aktivnog toka podataka može se izvršiti samo nakon izdekodovane cele slike i pre izdvajanje sledećeg toka podataka. Za slučaj kada je enkodovana slika sastavljena od dve slike (*interlaced*) uvek je potrebno izdekodovati obe slike pre nego što se promeni aktivni tok video podataka.

Prilikom dekodovanja dva toka video podataka vremenski se dele resursi video dekodera jedinice. Za dekodovanje u realnom vremenu dva toka podataka, osnovni zadatak treba da bude ravnomerna preraspodela resursa video dekodera. To zahteva da se kontekst video dekodera mora promeniti svaki put kada je VD slobodan (spreman za parsiranje). Inače, VD može neograničeno da bude zauzet na dekodovanju trenutno aktivnog toka podataka prouzrokujući da dekodovanje drugog toka podataka bude zaustavljeno.

Nakon promene toka podataka koji se dekoduju DND je odgovoran za obnavljanje prethodnog statusa korisničkih podataka i sadržaja video bafera. Izmena konteksta prouzrokuje gubljenje nekih podataka iz ulaznog bafera. Stoga, DND rukovaoc treba da "premoti" ulazni bafer za 128 bajta u nazad i obezbedi obradu svih podataka. "Premotavanjem" ulaznog bafera, neke slike mogu biti obrađene dva puta (ako je slika manja od 128 bajta). Iz tog razloga DND rukovaoc treba da prepozna navedenu situaciju i ignoriše ponovljene podatke (preskoči dekodovanje te slike i izdvajanje korisničkih podataka).

3.8 Skaliranje slike

U cilju uštede potrošnje memorijskih resursa video dekodera ima mogućnost skaliranja dekodovanih slika sa faktorom 1:2 ili 1:4. Ova opcija se uvek koristi kod dekodovanja dva HD video toka podataka koji se obično prikazuju u PiP (Slika u slici) ili PaP (Slika i slika) modu.

Teoretski, video dekodera može da skalira cele slike i da ih smesti u odvojene bafere ali ovakav tip skaliranja nema smisla na polju uštede memorijskih resursa. Stoga DND rukovaoc treba uvek da koristi tri ista bafera za prikaz bez obzira da li se dekoduje skalirana ili cela slika.

Skaliranje nije kompletno transparentna operacija za DND rukovaoc. Generalno, nije potrebno dodatno rukovanje od strane DND rukovaoca da bi se podržalo skaliranje izuzev podešavanja konfiguracije video dekodera i modula za prikaz slike. Da ne bi dolazilo do promene horizontalne rezolucije prikaza za slučaj kada se često manja faktora skaliranja potrebno je odbaciti sve slike koje imaju veći faktor skaliranja od aktivne i tako očuvati kvalitet slike.

DND rukovaoc treba da obezbedi programsku spregu dostupnu višim programskim slojevima koji treba da vode računa o skaliranju dekodovanih slika.

3.9 Kompresija podataka

Drugi način da se sačuva potrošnja memorijskih resursa je mogućnost video dekodera da čuva referentne slike u kompresovanom formatu. Ova opcija može da bude dozvoljena ili zabranjena na početku svakog zatvorenog GOP-a. Pošto VD ne vodi računa o načinu smeštanja slika u memoriji, promena kompresije podataka će prouzrokovati da neke slike budu izobličene (zbog neodgovarajuće interpretacije formata referentnih slika).

Promena podešavanja kompresije podataka u toku rada (kada je dekodovanje aktivno) će prouzrokovati da se ponovo izvrši sinhronizacija sa ulaznim tokom podataka sihrnoizuje (i stoga će neke slike biti izgubljene).

DND treba da podrži programsku spregu za kontroisanje ove opcije.

3.10 Kontrola prikaza

Modul za prikaz slike može biti podešen da čita "field" ili "frame" dekodovane slike. U ovom rešenju rukovaoca modul za prikaz slike treba da bude podešen za čitanje "frame" (celih) slika. DND rukovaoc treba da obezbedi da kontrola prikaza bude nezavisna od procesa dekodovanja. Jedina veza sa modulom za prikaz slike treba da bude preko listi bafera za prikaz u kojima se nalazi izdekodovane slike:

- Lista bafera za prikaz (eng. *Display List*)
- Lista čekanja (eng. *Delay List*)

Nakon što je slika izdekodovana od strane VD jedinice slika se smešta u listu bafera za prikaz. Modul za prikaz slike može da bude podešen da čitanje *interlaced* ili *progressive* video podatka. U svakom slučaju uzima se da je rezultat proces dekodovanja kompletna slika (sa gornjim i donjim poljem za slučaj *interlaced* slike).

MPEG standard dozvoljava postavljanje različite pozicije prikaza za svaku sliku posebno. U ovom slučaju DND mora da podesi modul za prikaz slike (ili poslednji memorijski bafer u kome se nalazi izdekodovana slika) za svaki "Vsync" period.

DND rukovaoc prosleđuje sve neophodne parametre modulu za prikaz slike nezavisno za svaku sliku:

- Bafer za prikaz (eng. *Buffer to display*)
- Rezolucija za sekvencu (eng. *Sequence resolution*)
- Rezolucija za prikaz (eng. *Display resolution*)
- Brzina potpune slike (eng. *Frame Rate*)
- "Scan mode" (interlaced or progressive)

Pre nego što se postave svi navedeni parametri, DND mora da konfiguriše memorijski bafer u kome se nalazi slika za prikaz. Ovo treba da se izvrši svaki put kada je prisutan "Pan scan" vektor u zaglavlju slike.

- Horizontalni "Pan-Scan Offset"

Takođe je predviđeno da se proslede neophodne informacije modulu za prikaz slike da bi se obezbedila odgovarajuća korekcija formata prikaza. Parametri koji se prosleđuju su:

- Format slike (eng. *Aspect Ratio*)
- Trenutni AFD (ako postoji)
- Trenutna bitska brzina

3.11 Rukovanje korisničkim podacima

Korisničke podatke izdvojene iz zaglavlja sekvence, GOP-a ili slike video dekodera upisuje u odvojene memorijske bafere fiksne veličine. DND rukovaoc treba da proverava bafere i sakuplja podatke nakon svake izdekodovane slike. Sakupljeni podaci se prosleđuju višim programskim nivoima.

Veoma je važno da se sakupljeni podaci pročitaju iz bafera pre nego što se on prepuni. DND je odogovoran za sakupljanje svih izdvojenih korisničkih podataka i njihovo čuvanje za slučaj kada je njihova veličina veća od memorijskog bafera. Veličina korisničkih podataka nije ograničena po MPEG-2 standardu i DND treba da čuva podatke u odgovarajućoj listi iz koje oni mogu biti pročitani od strane viših programskih slojeva. Na ovaj način se obezbeđuje da ne dođe do njihovog gubljenja i dozvoljava se njihovo čitanje bez ometanja procesa dekodovanja.

Sam DND rukovaoc ne treba da podržava obradu korisničkih podataka izuzev AFD deskriptora. Ovaj deskriptor treba da se izdvoji na nivo DND rukovaoca i da se izdvojeni parametri proslede modulu za prikaz slike.

3.12 Modovi rada DND rukovaoca

DND rukovaoc treba da podrži sledeće modove rada:

- Normalno dekodovanje i prikaz (audio/video sinhronizovani): Dekoduju se cele I P B slike
- Zamrznuta slika: Prikazuje se poslednja izdekodovana slika, sve ostale slike se preskaču
- Prikaz i dekodovanje samo I i P slika (bez audio/video sinhronizacije): I i P slike se dekoduju i prikazuju, B slike se preskaču
- Prikaz i dekodovanje samo I slika (bez audio/video sinhronizacije): I slike se dekoduju i prikazuju, P i B se preskaču
- Prikaz i dekodovanje I slika i sinhronizacija sa narednim GOP-om (bez audio/video sinhronizacije): I slike se dekoduju a zatim se vrši sinhronizacija sa narednom grupom slika

3.13 Rukovanje greškama

DND rukovaoc može da podrži samo osnovno rukovanje greškama i prikrije greške koje su obrađene od strane VD bloka na nivou dekodovanja. Generalno najbolje je prepustiti video dekodera modulu rukovanje greškama u najvećoj mogućoj meri pošto samo na nivou dekodovanja mogu da se obrade greške koje dese na u okviru jednog makro bloka. DND rukovaoc može samo da obradi greške koje se dese na nivou jedne slike ili grupe slika.

DND rukovaoc ima tri načina za detekciju grešaka:

- Greške detektovane na nivo video dekodera bez mogućnosti oporavka
U tom slučaju je potrebno resetovati video dekodera i vratiti se u stanje sinhronizacije ako je greška otkrivena na nivou referentne slike ili preskočiti sliku ako je greška detektovana u B slici.
- Pogrešne vrednosti parametara u okviru zaglavlja:
DND treba da preventivno proverava vrednost parametara u zaglavlju i ako se desi greška da pokuša da je otkloni i ako to nije moguće da se vrati u stanje sinhronizacije.
- Razlika u zaglavlju prosleđenim od strane ulaznog modula i izdvojenih zaglavlja od strane video dekodera:
U ovom slučaju DND treba da ispravi grešku koja je verovatno nastala zato što video dekodera i ulazni modul nisu sinhronizovani.

4. Realizacija

Realizacija DND rukovaoca se može podeliti u više razvojnih faza. Podela na razvojne faze je izvršena na osnovu same strukture DND rukovaoca.

Prva faza realizacije obuhvata programsku simulaciju VD (video dekodera) i MR (modul za prikaz slike) jedinica. U ovoj fazi programski je simulirano ponašanje navedenih jedinica i implementirane su odgovarajuće programske sprege. Pored simulacije fizičke platforme u ovoj fazi razvoja realizovane su i dodatne funkcije za potrebe testiranja ponašanja sistema u slučaju pojave grešaka u toku dekodovanja i prikaza slike.

U drugoj fazi razvoja implementiran je DND rukovaoc za slučaj dekodovanja i prikaza jednog toka video podataka.

Treća faza razvoja se odnosi na implementaciju DND rukovoca za slučaj dekodovanja dva toka video podataka.

Četvrta faza obuhvata realizaciju funkcija za rukovanje korisničkim podacima i obradu AFD deskriptora.

U petoj fazi je vršeno detaljno testiranje rada DND rukovaoca.

4.1 Video dekodera

Programski video dekodera je realizovan tako da simulira rad fizičke video dekodera jedinice. Za razliku od realnog sistema, programski video dekodera za ulaz ima video tok podataka koji se čuva u internoj datoteci. Umesto fizički odvojenih memorijskih blokova za potrebe simulacije koriste se memorijski blokovi u radnoj memoriji.

Generalno programski video dekodera čini jedna programska nit koja u beskonačnoj petlji čita podatke iz ulazne datoteke i vrši njihovo dekodovanje. Svi relevantni parametri video dekodera čuvaju se u globalnoj strukturi *decoder* (Tabela 1). Za slučaj dekodovanja dva ulazna toka podataka postoje dve globalne *decoder* strukture za svaki tok podataka posebno.

Struktura	Opis	
decoder	Sadrži relevantne parametre za VD	
Polja	Opis	Tip
dc_fbuf	Adresa odredišnog bafera za dekodovanje.	uint8*
fp_fbuf	Adresa prvog referentnog bafera.	uint8*
bp_fbuf	Adresa prvog referentnog bafera.	uint8*
frameLSize	Veličina luma komponente.	int
frameCSize	Veličina hroma komponente.	int
frameYCrCbSize	Ukupna veličina luma i obe hroma komponente.	int
fieldSelect	Obeleženo polje.	struct bs
ld	Struktura sa parametrima neophodnim za rukovanje sa ulaznom datotekom.	struct bs
hd	Struktura koja sadrži sva zaglavlja iz	struct mpeg_head

	MPEG-2 toka podataka.	
headerFlags	Indikatori za sva izdvojena zaglavlja.	int

Tabela 1 decoder struktura

Najvažnija funkcija VD modula je *decodeLoop*, čiji je blok dijagram algoritma prikazan na slici 15. U njoj se vrši dekodovanje video toka podataka iz ulazne datoteke. Ova funkcija se proziva iz *decoderThread* niti i izvršava se u beskonačnoj petlji. Proces dekodovanja započinje prijemom komande *VD_PARSE_STREAM* od DND rukovaoca nakon čega se vrši izdvajanje odgovarajućih zaglavlja iz MPEG-2 toka podataka i korisničkih podataka. Strukture zaglavlja u kojima se čuvaju izdvojeni parametri navedene su u tabeli 3. Kada su sva zaglavlja izdvojena dekodier šalje *VD_CMD_READY* poruku DND rukovaocu nakon čega DND može da izda komandu *VD_DECODE_PICTURE* čime se započinje proces dekodovanje slike. Kada je slika izdekodovana šalje se *VD_PROC_READY* poruka DND rukovaocu kao signal da je dekodovanje izvršeno i da može da se pokrene izdvajanje novog zaglavlja. U okviru *decodeLoop* funkcije podržano je dekodovanje kompresovanih i horizontalno skaliranih slika.

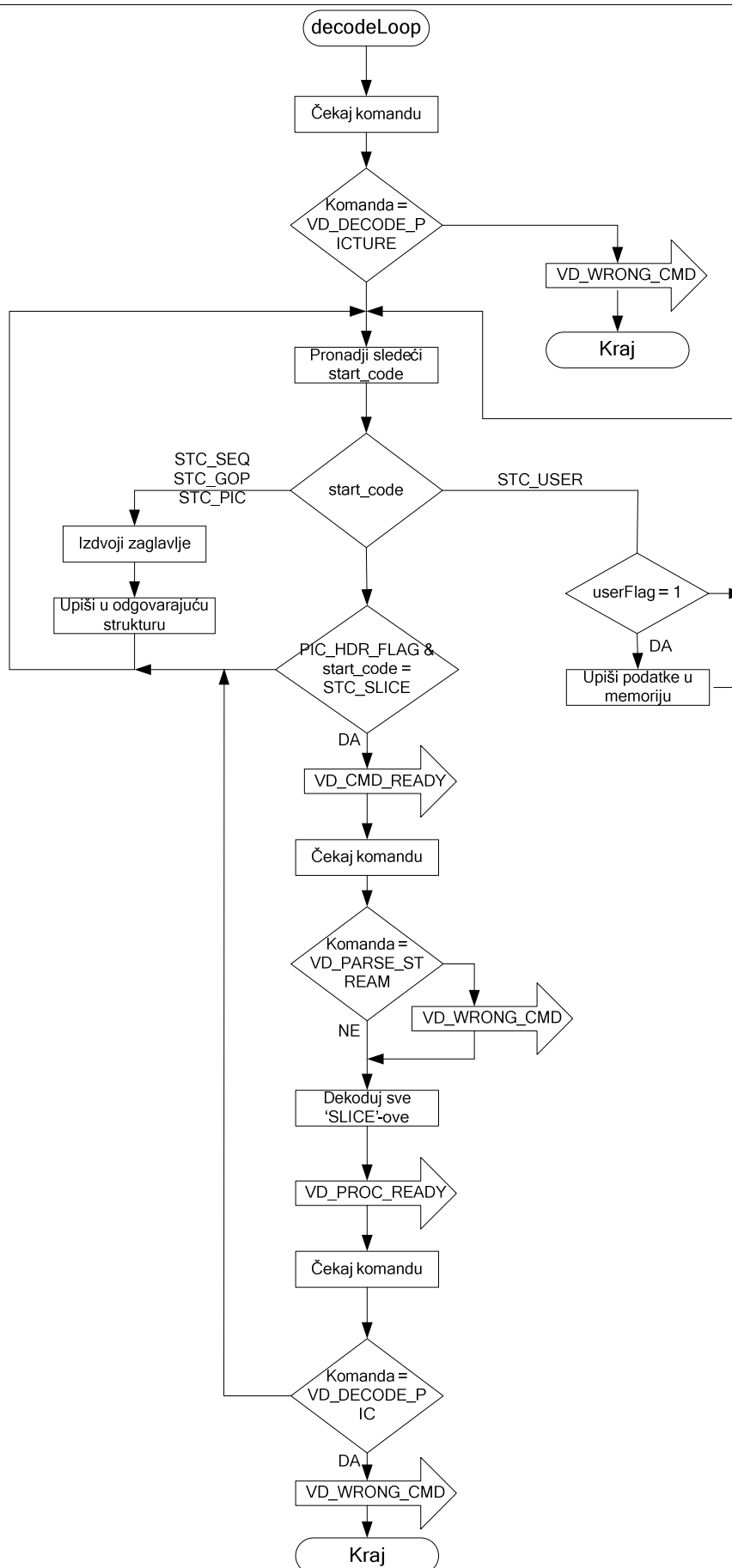
Naziv strukture	Komentar
VD_MPEG_sequ_header_t	Sadrži polja zaglavlja sekvence.
VD_MPEG_group_header_t	Polja zaglavlja grupe slika (GOP).
VD_MPEG_pic_header_t	Polja zaglavlja slike.
VD_MPEG_pan_scan_info_t	Polja "pan scan" vektora.

Tabela 2 Strukture MPEG-2 zaglavlja

Za komunikciju VD modula sa DND rukovaocem realizovan je skup sprežnih funkcija (Tabela 4) koje treba da zamene sprežne funkcije rukovaoca fizičkim uređajem (eng. *driver*).

Funkcija	Parametri	Komentar
<i>VD_Init</i>	uint32_t size, uint32_t bss1, int32_t bss1_offset, uint32_t ud1, uint32_t ref1, uint32_t ref2	Inicijalizacija video dekodera.
<i>VD_RegisterIsrCB</i>	void (*cb)(int)	Registrowanje "callback" funkcije.
<i>VD_SetDecodeInfo</i>	int frame_cnt, int field_cnt, int decode_buffer	Postavljanje parametara izdekodovane slike.
<i>VD_SendCommand</i>	int command	Slanje komandi.
<i>VD_GetSeqInfo</i>	VD_MPEG_sequ_header_t *	Izdvajanje zaglavlja sekvence.
<i>VD_GetGOPInfo</i>	VD_MPEG_group_header_t*	Izdvajanje GOP zaglavlja.
<i>VD_GetPicInfo</i>	VD_MPEG_pic_header_t *	Izdvajanje zaglavlja slike.
<i>VD_GetPanScanInfo</i>	VD_MPEG_pan_scan_info_t *	Izdvajanje "pan scan" vektora.
<i>VD_SetPara</i>	int para, int value	Funkcija za postavljanje konfiguracionih parametara.
<i>VD_GetPara</i>	int para	Funkcija za čitanje konfiguracionih parametara.
<i>VD_SwitchStream</i>	int bss, int bref, int fref, int ud	Promena aktivnog toka podataka koji se trenutno dekoduje

Tabela 3 Sprežne funkcije video dekodera

Slika 15 Blok dijagram algoritma funkcije *decodeLoop*

Funkcija *VD_Init* služi za inicijalizaciju programskog video dekodera. U njoj se vrši inicijalizacija parametara *decoder* strukture (Tabela 2), memorijskih bafera kao i kreiranje niti *decoderThread*. Za slučaj dekodovanja dva ulazna toka podataka inicijalizuju se dve *decoder* strukture.

VD_GetSeqInfo, *VD_GetGOPInfo*, *VD_GetPicInfo*, *VD_GetPanScanInfo* funkcije služe za čitanje parametara izdvojenih iz MPEG-2 zaglavlja. Kada je odgovarajuće zaglavlje izdvojeno funkcije kao povratnu vrednost vraćaju 0 i popunjavaju prosleđenu strukturu u suprotnom slučaju vraća se -1. Strukture zaglavlja koja se izdvajaju su date u nastavku (Tabela 4, ...).

Struktura	Opis	
VD_MPEG_sequ_header_t	Sadrži parametre zaglavlja sekvence	
Polja	Opis	Tip
bit_rate	Bitska brzina.	uint32_t
hor_size	Horizontalna veličina.	uint16_t
ver_size	Vertikalna veličina.	uint16_t
hor_disp_size	Horizontalna veličina za prikaz.	uint16_t
ver_disp_size	Vertikalna veličina za prikaz.	uint32_t
vbv_bsize	Veličina vbv bafera.	uint8_t
pel_asp	"Pel" format slike.	uint8_t
pic_rate	Format slike.	uint8_t
csprm_flag	Indikator graničnog parametra.	uint8_t
ld_iqntmtx	Indikator intra kvantizacije matrice	uint8_t
iqntmtx[64]	Intra kvantizaciona matrica.	uint8_t
ld_nqntmtx	Indikator non-intra kvant. matrice	uint8_t
nqntmtx[64]	Non-Intra kvantizaciona matrica.	uint8_t
MPEG2	MPEG-2 tok podataka.	uint8_t
profile_level_id	"profile_and_level" indikator	uint8_t
progre_seq	Progresivna sekvenca.	uint8_t
chroma_format	Hroma format.	uint8_t
low_delay	Slabo kašnjenje.	uint8_t
frame_rate_ext_n	Bitska brzina frejma n.	uint8_t
frame_rate_ext_d	Bitska brzina frejma d.	uint8_t
horizontal_size_ext	Horizontalna veličina (produžena).	uint8_t
vertical_size_ext	Vertikalna veličina (produžena).	uint8_t
mpeg_1_flag	Mpeg-1 indikator.	uint8_t
video_format	Video format.	uint8_t
color_descript	Deskriptor boje.	uint8_t
col_prim	Glavna boja.	uint8_t
transfer_char	Prenos karakteristika.	uint8_t
matrix_coeff	Matrični koeficijenti.	uint8_t

Tabela 4 Struktura zaglavlja sekvence

Struktura	Opis	
VD_MPEG_group_header_t	Sadrži parametre GOP zaglavlja	
Polja	Opis	Tip
time_code_hour	Vremenski kod za sate.	uint8_t
time_code_minute	Vremenski kod za minute.	uint8_t
time_code_sec	Vremenski kod za sekunde.	uint8_t
time_code_pict	Vremenski kod slike.	uint8_t
drop_frame	Odbačen frejm.	uint8_t
clsd_gop	Zatvoren GOP.	uint16_t
brkn_link	Prekinuta veza.	uint16_t

Tabela 5 Struktura GOP zaglavlja

Struktura	Opis	
VD_MPEG_pic_header_t	Sadrži parametre zaglavlja slike	
Polja	Opis	Tip
temp_ref	Privremena referenca.	uint8_t
vbv_delay	Vbv kašnjenje.	uint8_t
type	Tip slike.	uint8_t
intra_dc_prec	Intra dc tačnost.	uint8_t
picture_struct	Struktura slike.	uint8_t
top_field_first	Gornje polje prvo.	uint8_t
frame_pred_frame_dct	Indikator predikcije frejma.	uint8_t
conc_mv	Indikator kodovanja vektora pokreta	uint8_t
q_scale_type	Indikator inverzne kvantizacije.	uint8_t
intra_vlc_format	Indikator dekodovanja transformisanih koeficijenata.	uint8_t
alternate_scan	Alternativno skeniranje.	uint8_t
repeat_first_field	Indikator ponavljanja prvog polja.	uint8_t
progressive_frame	Progresiv frejm.	uint8_t
composite_display_flag	Indikator kompozitnog prikaza.	uint8_t
v_axis	V osa.	uint8_t
field_seq	Sekvenca polja.	uint8_t
sub_carrier	"sub-carrier/line" frekvencioni odnos	uint8_t
burst_amplitude	Burst amplituda.	uint8_t
sub_carrer_phase	Faza "sub carrer" reference.	uint8_t
frame_centre_hor_0	Vertikalni centralni opseg slike 0.	uint16_t
frame_centre_vert_0	Horizontalni centralni opseg slike 0.	uint16_t
frame_centre_hor_1	Vertikalni centralni opseg slike 1.	uint16_t
frame_centre_vert_1	Horizontalni centralni opseg slike 1.	uint16_t
frame_centre_hor_2	Vertikalni centralni opseg slike 1.	uint16_t
frame_centre_vert_2	Horizontalni centralni opseg slike 1.	uint16_t
number_of_fco	Broj opsega.	uint8_t

Tabela 6 Struktura zaglavlja slike

Struktura	Opis	
VD_MPEG_pan_scan_info_t	Sadrži parametre "pan scan" vektora.	
Polja	Opis	Tip
numPanScanOffsets	Broj "pan scan" opsega.	int32_t
PanScanOffsetX[3]	Horizontalni opsezi.	int32_t
PanScanOffsetY[3]	Vertikalni opsezi.	int32_t

Tabela 7 Struktura "pan scan" zaglavlja

Funkcija *VD_SetDecodeInfo* se koristi za postavljanje adresa memorijskih bafera pre početka dekodovanja slike. Postavljaju se adresa bafera za smeštanje referentnih slika i bafera za dekodovanje slike.

VD_SwitchStream funkcija služi za promenu aktivnog toka podataka koji se dekoduje. U okviru ove funkcije se menja ulazna datoteka koja se dekoduje i menjaju se adrese memorijskih bafera za smeštanje izdekodovane i referentnih slika.

Funkcija *VD_RegisterIsrCB* se koristi da se registruje *callback* funkcija preko koje video dekođer može da komunicira sa DND rukovaocem.

VD_SetPara i *VD_GetPara* funkcije se koriste za postavljanje i čitanje parametara video dekođera koji se odnose na njegovu internu konfiguraciju. Parametri koji se koriste i njihov opis su navedeni u sledećoj tabeli (Tabela 8):

Naziv parametra	Opis
VD_LAST_PIC_ADDRESS	Adresa poslednje dekodovane slike.
VD_HSCALE	Horizontalno skaliranje
VD_COMPRESSION	Kompresovanje slike.
VD_UD_CONTROL	Dozvola čitanja korisničkih podataka.
VD_UD_WR_PTR_POS	Pozicija na koju se trenutno upisuju korisnički podaci.

Tabela 8 Konfiguracioni parametri video dekodera

VD_SendCommand funkcija obezbeđuje slanje komandi od strane DND rukovaoca video dekodera. Komande, sa odgovarajućim opisom, koje se šalju su navedene u sledećoj tabeli (Tabela 9):

Naziv parametra	Opis
VD_DECODE_STOP	Zaustavlja rad vide dekodera.
VD_PARSE_STREAM	Izdvajanje zaglavlja iz MPEG-2 toka podataka.
VD_DECODE_PICTURE	Dekodovanje slike.
VD_VBV_DATA_AVIL	Dostupni podaci u ulaznom baferu.

Tabela 9 Komande koje šalje DND rukovaoc

4.2 Programski modul za prikaz slike (MR)

Ovaj programski modul je zadužen za prikazivanje izdekodovanih slika. Implementiran je kao jedna programska nit. Za slučaj prikazivanja izdekodovanih slika iz dva različita toka podataka (dva bafera za prikaz) inicijalizuju se dve programske niti. Svi relevantni parametri programske MR jedinice se čuvaju u *mpr_struct_t* strukturi (Tabela 11, svaki inicijalizovani MR ima svoju strukturu).

Za prikaz izdekodovanih slika MR se oslanja na funkcije iz linux X11 biblioteke. U principu nakon inicijalizacije i pokretanja niti MR u beskonačnoj petlji čita podatke iz ulaznog bafera i vrši njihovo prikazivanje. Nakon prikaza slike MR javlja DND rukovaocu da je slika uspešno prikazana i DND može da upiše novu sliku u bafer za prikaz.

Struktura	Opis	
<i>mpr_struct_t</i>	Parametri MR modula.	
Polja	Opis	Tip
thread	Nit za koju je vezana struktura.	pthread_t
w, h, rate	Širina, visina i bitska brzina slike za prikaz.	int
topY, botY, topC, botC	Adrese bafera: gornje luma, donje luma, gornje hroma, donje hroma komponente.	int
int (*cb)(int)	Adresa "callback" funkcije.	int
mode	Mod rada.	enum
field	Polje.	int
invfield	Invertovano polje.	int
hofposc	Horizontalni opseg slike (pan scan).	int
skip_pixel	Indikator preskakanja "pixel-a".	int
*display	Parametri neophodni za korištenje funkcija iz X11 biblioteke.	Display
win		Window
gc		GC
gcv		XGCValues
screen		int
*ximage		XImage
*buffer		unsigned char
depth, bpl		int

Tabela 10 *mpr_struct_t* struktura

Za komunikaciju sa DND rukovaocem realizovan je skup sprežnih funkcija (Tabela 12) koje treba da zamene sprežne funkcije rukovaoca fizičkim uređajem. Funkcije koje su implementirane su navedene u tabeli:

Funkcija	Parametri	Komentar
<i>MPR_Init</i>	<i>int</i> ID, <i>int</i> w, <i>int</i> h, <i>int</i> rate, <i>int</i> topY, <i>int</i> botY, <i>int</i> topC, <i>int</i> botC	Inicijalizacija parametara.
<i>MPR_RegisterVSyncCB</i>	<i>void</i> (*cb)(<i>int</i>)	Registrowanje "callback" funkcije.
<i>MPR_SetFrameInfo</i>	<i>int</i> ID, <i>struct frame_info</i> *f	Postavljanje parametara slike za prikaz.
<i>MPR_SetPara</i>	<i>int</i> ID, <i>int</i> para, <i>int</i> value	Funkcija za postavljanje konfiguracionih parametara.
<i>MPR_GetPara</i>	<i>int</i> ID, <i>int</i> para	Funkcija za čitanje konfiguracionih parametara.

Tabela 11 Sprežne funkcije MR modula

Funkcija *MPR_Init* služi za inicijalizaciju programskog MR modula. U njoj se vrši inicijalizacija parametara *mpr_struct_t* strukture i prozora za prikazivanje izdekodovanih slika. U slučaju prikazivanja slika iz dva različita ulazna bafera inicijalizuju se dve odvojene programske niti.

Funkcija *MPR_RegisterVSyncCB* se koristi da se registruje *callback* funkcija preko koje video dekodier može da komunicira sa DND rukovaocem. *Callback* funkcija se proziva svaki put kada istekne programski časovnik (eng. *timer*) u programskoj niti MR modula a nakon što se prikaže poslednja slika iz bafera za prikaz. Na ovaj način se omogućuje sinhronizuje rad programskog video dekodera i MR modula od strane DND rukovalaca.

Funkcija *MPR_SetFrameInfo* se koristi za postavljanje parametara slike za prikaz. Parametri se postavljaju promenom polja u okviru *frame_info* strukture (Tabela 12) koja sadrži relevantne parametre slike koja se prikazuje.

Struktura	Opis	
<i>frame_info</i>	Parametri vezani za sliku koja se prikazuje.	
Polja	Opis	Tip
mode	Mod rada.	uint32_t
horSize, verSize	Horizontalna i vertikalna veličina slike.	int32_t
horDisp, verDisp	Horizontalna i vertikalna veličina prikaza slike.	int32_t
hofposc	Horizontalni opseg slike (pan scan).	int32_t
aspect_ratio	Format slike.	int32_t
afd	AFD deskriptor.	int32_t
field	Polje.	int32_t

Tabela 12 *frame_info* struktura

MPR_SetPara i *MPR_GetPara* funkcije se koriste za postavljanje i čitanje parametara MR modula koji se odnose na njegovu internu konfiguraciju. Parametri koji se koriste i njihov opis su navedeni u sledećoj tabeli (Tabela 13):

Naziv parametra	Opis
MPR_FIELD_MODE	Mod rada .
MPR_CURR_FIELD	Trenutno polje za prikaz.
MPR_INV_FIELD	Invertovano polje.
MPR_HOFPOSC	Horizontalni opseg slike.
MPR_SKIP_PIXEL	Preskakanje "piksela".

Tabela 13 Konfiguracioni parametri video dekodera

4.3 DND rukovaoc

Arhitekturu DND rukovaoca čine:

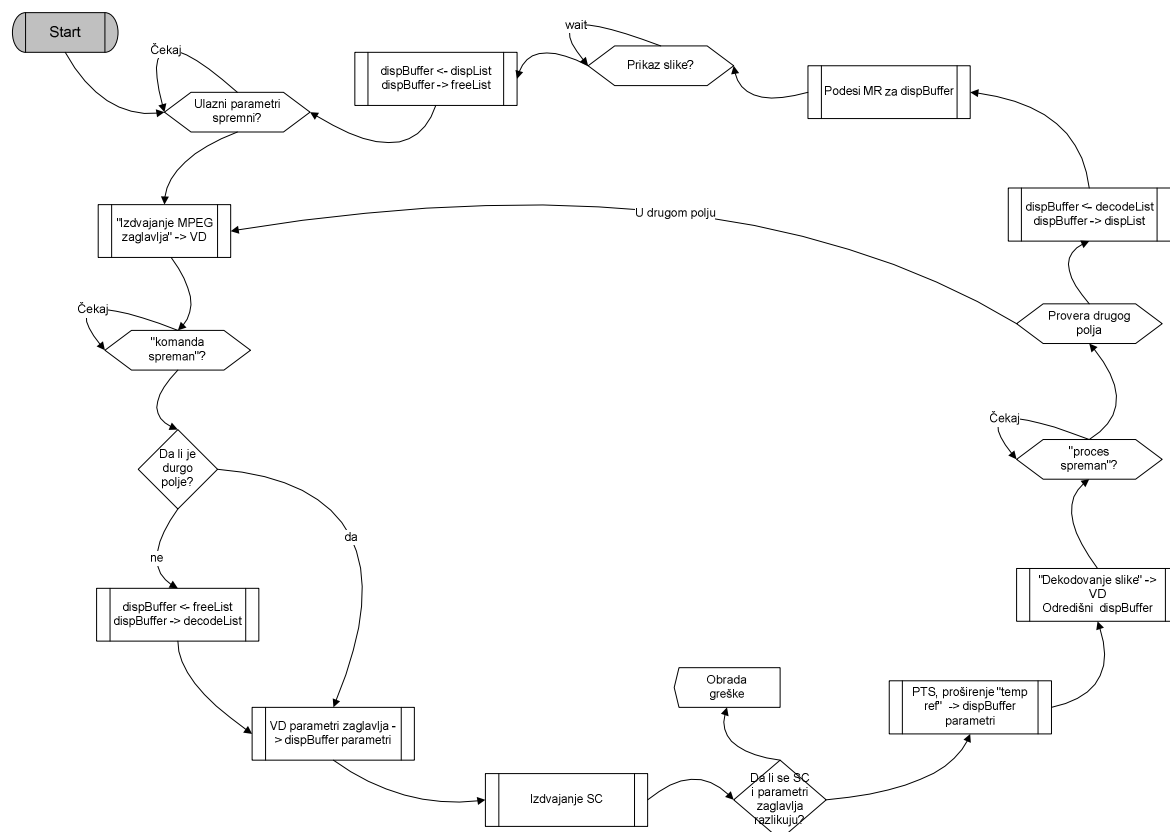
- Jedna programska nit raspoređena za upravljanje dekodovanjem dva toka video podataka.
- Vsync ISR callback funkcija za komunikaciju sa slojem za rukovanje modulom za prikaz slike; rukovanje procesom prikazivanja slike.
- VD ISR callback funkcija za komunikaciju sa slojem za rukovanje VD jedinicom; rukovanje procesom dekodovanja slike.
- SC ISR callback funkcija za komunikaciju sa modulom za prijem podataka; sakupljanje SC podataka.

DND rukovalac podržava rukovanje dekodovanjem dva ulazna toka video podataka. Za svaki tok podataka vezuje se jedna *dnd_stream_t* struktura (Tabela 14) u kojoj se čuvaju svi relevantni parametri. U globalnoj promenljivoj *currentStream* (0 ili 1) se čuva informacija o trenutno aktivnom toku podataka. Za slučaj kada se dekoduje jedan tok podataka *currentStream* promenjiva uvek je postavljena na inicijalnu vrednost i DND koristi samo jednu *dnd_stream_t* strukturu. Kada se obrađuju dva toka podataka aktivna struktura se menja nakon svake izdekodovane slike.

Programska nit DND rukovaoca vodi računa o bezbednoj promeni konteksta prilikom dekodovanja dva toka podataka. Promena aktivnog toka podataka koji se dekoduje može da se izvrši samo ako su zadovoljeni sledeći uslovi:

- Postoje dostupni ulazni baferi sa enkodovanim slikama
- Postoje slobodni baferi za smeštanje izdekodovanih slika
- Status VD je spreman

Promena toka podataka koji se dekoduje se izmenjuje slanjem komande za izdvajanje zaglavlja VD modulu. Rukovanje procesom dekodovanja je opisan dijagramom na slici 17.



Slika 16 Proces dekodovanja video toka podataka

Parametri koji se nalaze u *dnd_stream_t* strukturi mogu da se grupišu u parametre za:

- čuvanje statusa ulaznog toka podatka
- fazu dekodovanja slike
- upravljanje memorijskim baferima
- MPEG informacije izdvojene iz toka podataka
- rukovanje korisničkim podacima
- kontrolu video dekodera
- konfiguraciju video dekodera

Struktura	Opis	
<i>dnd_stream_t</i>	Parametri DND rukovaoca.	
Polja	Opis	Tip
state	Trenutno stanje(stop,play,sync...).	int
mode	Mod rada(normal,freeze,pause...).	int
vdReady	Indikator da VD spreman.	int
mask_field		int
lastFwd	Poslednja "prednja" referentna slika.	buf_info_t
ref	Indikator referentne slike.	int
n	Broj bafera.	int
lastPicAddr	Poslnja dekodovana slika(adres).	int
bssID	Identifikator ulaznog bafera.	int32_t
frefID	Identifikator prednjeg referentnog bafera.	int32_t
brefID	Identifikator zadnjeg referentnog bafera.	int32_t
udID	Identifikator bafera sa korisničkim podacima.	int32_t
mprID	Identifikator MPR bafera za prikaz slke.	int32_t
freeList	Lista slobodnih bafera.	list_t
decodeList	Lista bafera za dekodovanje.	list_t
displayList	Lista bafera za prikaz.	list_t
delayList	Lista bafera koji se trenutno čita od MR	list_t
pic_info	Zaglavlje slike.	VD_MPEG_pic_header_t
seq_info	Zaglavlje sekvence.	VD_MPEG_sequ_header_t
gop_info	GOP zaglavlje.	VD_MPEG_group_header_t
pan_scan_info	Zaglavlje "pan scan" vektora.	VD_MPEG_pan_scan_info_t
seqHeaderReceived	Indikator da je zaglavlje sekvence pročitano.	int
firstGOP	Indikator prviog GOP-a.	uint32_t
user_data_len	Dužina korisničkih podataka.	uint32_t
user_data_info	Korisnički podaci.	uint8_t
udQueue	Lista korisničkih podataka.	list_t
udQueueFree	Lista slobodnih bafera za smeštanje korisničkih podataka.	list_t
active_ud_block	Trenutno aktivni blok korisničkih podataka.	user_data_block_t
afd	AFD deskriptor.	uint32_t
skipFrame	Indikator za preskakanje sledeće slike.	uint32_t
resync	Indikator za ponovnu sinhronizaciju.	uint32_t
picType	Tip slike.	uint32_t
skipNextB	Indikator preskakanja sledeće B slike.	uint32_t
pts	PTS.	uint32_t
pts_valid	Indikator ispravnosti za PTS.	uint32_t

scQueue	Lista za čuvanje start kodova.	list_t
scQueueFree	Lista slobodnih bafera za start kodove.	list_t
correct_seq_info	Poslednje izdvojeno zaglavlje sekvence.	VD_MPEG_sequ_header_t
erLevel	Nivo dozvoljenih grešaka.	int
mismatchCount	Broj pogrešno dekodovanih slika.	int
ptsCheck	Indikator za proveru PTS-a.	int
ptsOffset	Opseg za PTS.	int
hscale	Horizontalno skaliranje slike.	int

Tabela 14 DND_stream_t struktura

DND rukovalac može da se nalazi u četiri režima rada. Promena režima rada vrši se pozivanjem lokalne funkcije *locChangeState* koja kao parametre prima identifikator toka podataka za koji se vrši promena i novi režim rada. Za slučaj dekodovanja dva toka podataka promena režima rada može nezavisno da se vrši za svaki tok podataka. Mogući režimi rada sa odgovarajućim opisom su navedeni u sledećoj tabeli:

Režim rada	Opis
DND_STATE_STOP	Podaci se ne obrađuju, VD je zaustavljen, podaci se ne prosleđuju MR jedinici.
DND_STATE_IDLE	Ulazni podaci se ne dekoduju, poslednja ispravna slika je prosleđen prema MR modulu.
DND_SYNC_MODE	Ulazni podaci se preskaču dok se ne detektuje zaglavlje sa sekvencom ili I slika. Posle toga se prelazi u DND_STATE_PLAY stanje.
DND_STATE_PLAY	Ulazni podaci se dekoduju i prikazuju u zavisnosti od izabranog DND moda rada.

Tabela 15 Režimi rada DND rukovaoca

DND rukovaoc podržava različite modove dekodovanja slika (Tabela 18). Promena moda dekodovanja vrši se pozivanjem lokalne funkcije *locChangeState* koja kao parametre prima identifikator toka podataka za koji se vrši promena i novi mod dekodovanja.

U okviru *dnd_stream_t* strukture nalaze se identifikatori (*bssID*, *frefID*, *brefID*, *udID*) za svaki memorijski bafer. Za upravljanje memorijskim resursima za smeštanje i prikaz dekodovanih slika koristi se *buf_info_t* struktura i četiri jednostruko spregnute liste:

1. Lista slobodnih bafera (*freeList*); sadrži inicijalizovane *buf_info_t* strukture.
2. Lista za dekodovanje (*decodeList*); može da ima samo jedan element i sadrži određeni bafer u koji se trenutno dekoduje slika
3. Lista bafera za prikaz (*displayList*); sadrži bafere sa izdekodovanim slikama koje su spremne za prikaz. Ova lista je sortirana na osnovu trenutne reference (*temp_ref*)
4. Lista čekanja (*delayList*); može da sadrži samo jedan element i sadrži bafer koji se trenutno čita od strane MR bloka

Parametri koji opisuju memorijske bafer su opisani *buf_info_t* strukturom (Tabela 17).

Struktura	Opis	
<i>buf_info_t</i>	Parametri bafera za smeštanje izdekodovanih slika.	
Polja	Opis	Tip
q	Zaglavlje liste kojoj pripada.	listMember_t
n	Broj bafera za prikaz (0-2).	int
pic_type	Tip slike.	int
temp_ref	Trenutna referenca.	int
gopStart	Početak GOP-a.	int
topFirst	Gornje polje prvo.	int

numFields	Broj polja koja treba da se prikažu.	int
fieldCount	Brojač prikazanih polja.	int
pts	PTS	uint32_t
pts_valid	Indikator ispravnosti za PTS.	uint32_t
psi	Zaglavlje "pan scan" vektora.	VD_MPEG_pan_scan_info_t
frame	Frame struktura.	frame_info
display	Display struktura.	display_info

Tabela 16 buf_info_t struktura

MPEG informacije izdvojene iz toka podataka čuvaju se u posebnim strukturama u okviru *DND_stream_t* strukture.

Izdvojeni korisnički podaci se čuvaju u *udQueue* listi do trenutka dok se ne pročitaju od strane viših programskih nivoa. Da bi se obezbedilo da se u listu uvezuju samo ispravni korisnički podaci u *active_ud_block* parametru se čuva poslednji blok izdvojenih podataka i tek ako su svi uslovi zadovoljeni on se uvezuje u *udQueue* listu.

U okviru *DND_stream_t* strukture se nalaze i parametri neophodni za kontrolisanje ispravnog dekodovanja slika (*resync*, *picType*, *skipNextB*).

4.3.1 DND sprežne funkcije

Za upravljanje radom DND rukovaoca od strane viših programskih slojeva realizovan je skup sprežnih funkcija (Tabela 17). Funkcije koje se koriste za upravljanje DND rukovaocem su navedene u sledećoj tabeli:

Funkcija	Parametri	Komentar
<i>DND_Init</i>	nema parametre	Inicijalizacija DND rukovaoca.
<i>DND_Stop</i>	int streamID	Zaustavljanje DND rukovaoca.
<i>DND_Play</i>	int stream, int mode	Pokretanje DND-a.
<i>DND_SetMode</i>	int command	Postavljanje moda rada.
<i>DND_Control</i>	int stream, int mode	Postavljanje kontrolnih parametara.
<i>DND_PictureSC</i>	int stream, int32_t tempRef, int32_t picType, uint32_t pts, int32_t ptsValid	Rukovanje SC (start kodov).
<i>DND_GetUserData</i>	int stream, uint8_t* data	Izdvajanje korisničkih podataka.

Tabela 17 Sprežne funkcije DND rukovaoca

Funkcija *DND_Init* služi za inicijalizaciju DND rukovaoca. U okviru ove funkcije postavljaju se početne vrednosti za *dnd_stream_t* strukturu (za slučaj dekodovanja dva toka podataka inicijalizuju se dve *dnd_stream_t* strukture), zauzimaju se memorijski resursi i pozivaju funkcije za inicijalizaciju VD i MPR modula.

DND_Stop funkcija zaustavlja proces dekodovanja video dekoader jedinice i postavlja stanje DND rukovaoca u stanje "zaustavljen".

Funkcija *DND_Play* pokreće proces dekodovanja i prevodi DND rukovaoc u stanje sinhronizacije. Kao ulazne parametre funkcija prima identifikator toka podataka čije dekodovanje se započinje i mod rada DND rukovaoca (Tabela 18).

Mod rada	Opis
DND_DECODE_NORMAL	Normalan mod dekodovanja.
DND_DECODE_FREEZE	Prikazuje zadnju izdekodovanu sliku i preskače ostale.
DND_DECODE_PAUSE	Prikazuje se zadnja izdekodovana slika i ne dekoduju se naredne.
DND_DECODE_IP	Dekoduju se samo I i P slike.

DND_DECODE_I	Dekoduju se samo I slike.
DND_DECODE_I_RECYN	Dekoduje se I slika i sinhronizuje sa sledećim GOP-om.

Tabela 18 Modovi rada DND rukovaoca

Funkcije *DND_Control* i *DND_SetMode* služe za postavljanje kontrolnih parametara DND rukovaca i promenu moda rada.

DND_GetUserData služi za izdvajanje sačuvanih korisničkih podataka iz *udQueue* liste.

4.3.2 Rukovanje memorijskim baferima

Za rukovanje listama memorijskih bafera implementirane su četiri funkcije:

- *bufFreeToDecode*

Ova funkcija uzima jedan element iz liste slobodnih bafera i dodaje u listu za dekodovanje. Ovo se dešava pre slanje "VD_DECODE_PICTURE" komande video dekomoderu.

- *bufDecodeToDisplay*

U ovoj funkciji se uzima prvi element iz liste za dekodovanje, vrše se proširenje trenutne reference na 32 bitnu vrednost i dodaje se u listu bafera za prikaz. Ova funkcija se poziva nakon što se desi "VD_PROC_READY" prekid svaki put kada se izdekoduje cela slika.

- *bufDisplayToDelay*

Ova funkcija uzima prvi element iz liste bafera za prikaz i dodaje u listu bafera koji čekaju da se prikažu. Ovo se dešava u okviru *mprISR* prekidne funkcije svaki put kada treba da se prikaže sledeća slika.

- *bufDelayToFree*

Ova funkcija se poziva svaki put kada se signalizira da je programski modul za prikaz slike završio sa prikazivanjem poslednje slike iz bafera za prikaz a pre poziva funkcije *bufDisplayToDelay*. U okviru ove funkcije se uklanja prvi element liste bafera koji se prikazuju i dodaje u listu slobodnih.

Da bi se izdekodovane slike uvezale u listu bafera za prikaz u ispravnom redosledu koristi se polje trenutne reference (eng. *temporal reference*). Trenutna referenca je brojač koji počinje od nule i uvećava se za svaku izdekodovanu sliku. Za enkodovane slike koje se sastoje iz gornjeg i donjeg polja trenutna referenca je ista. Na početku svake grupe slika vrednost trenutne reference se postavlja na nulu.

U okviru svakog GOP-a čuva se maksimalna vrednost trenutne reference. Ova maksimalna vrednost se zatim koristi kao opseg trenutne reference za sledeći GOP. Ovo se dešava svaki put kada kada se *buf_info_t* struktura prebacuje iz liste za dekodovanje u listu za prikaz. Proširena trenutna referenca se pre uvezivanja novog elementa u listu za prikaz postavlja u polje "prioritet" (skup funkcija za rad sa jednostrukom spregnutom listom podržava uvezivanje elemenata liste na osnovu zadatog prioriteta) kako bi se obezbedio prikaz slika u odgovarajućem redosledu.

4.3.3 Funkcije za obradu prekida

U okviru DND rukovaoca su implementirane reakcije na tri osnovna tipa događaja (Tabela 20). Prekid može da bude signaliziran od strane MR, VD i ulaznog modula.

Funkcija	Parametri	Komentar
<i>vdISR</i>	<i>int</i> type	Obrada prekida signaliziranih od strane VD jedinice.
<i>mprISR</i>	<i>int</i> streamID	Obrada prekida signaliziranih od strane modula za prikaz slike.
<i>scISR</i>	<i>int</i> stream <i>startcode_t*</i> tsd_sc	Obrada prekida signaliziranih od strane ulaznog modula.

Tabela 19 Funkcije za obradu prekida DND rukovaoca

vdISR prekidna rutina kao parametar prima prekidnu poruku od VD modula. U okviru ove rutine se kontroliše najveći deo operacija vezanih za proces dekodovanja. Zavisno od trenutnog stanja DND rukovaoca i primljene poruke vrši se:

- slanje odgovarajućih poruka VD jedinici
- pozivanje funkcija za obradu greške
- pozivanje funkcija zaduženih za kontrolu procesa dekodovanja

mprISR prekidna rutina kao parametar prima prekidnu poruku od modula za prikaz slike. Za vreme trajanja prekida signaliziranog od strane MR modula potrebno je da se postavi nova slika za prikaz. Zavisno od moda rada modula za prikaz slike (prikaz *frame* ili *field* slika) nova slika za prikaz se postavlja svaki ili svaki drugi prekid. Sve informacije neophodne MR modulu se prosleđuju preko *buf_info_t* strukture iz liste za prikaz. U funkciji se takođe vodi računa o ispravnosti podataka koji se prosleđuju.

scISR prekidna rutina rukuje prekidima primljenim od strane ulaznog modula. Kao ulazni parametar prima se *vid_startcode_t* struktura (Tabela 20). Primljena struktura se zatim uvezuju u *scQueue* listu i budi se DND nit.

Struktura	Opis	
<i>vid_startcode_t</i>	SC parametri.	
Polja	Opis	Tip
<i>pic_type</i>	Tip slike.	uint32_t
<i>pic_struct</i>	Struktura slike.	int32_t
<i>pts</i>	PTS	int32_t
<i>pts_valid</i>	Indikator ispravnosti za PTS	int
<i>temp_ref</i>	Trenutna referenca.	int
<i>bitrate</i>	Bitska brzina.	int32_t

Tabela 20 *vid_startcode_t* struktura

4.3.4 Upravljanje procesom dekodovanja

U procesu dekodovanja DND rukovaoc može da se nalazi u četiri režima rada (Tabela 15). Zavisno od trenutnog režima rada DND na odgovarajući način upravlja procesom dekodovanja video podataka i radom VD i MR programskih modula.

U "DND_STATE_IDLE" režimu DND osigurava da se na bezbedan način preskaču sve slike iz aktivnog toka podataka. Jedini očekivani prekid u ovom režimu rada je "VD_CMD_READY". U ovom slučaju se jednostavno izdvajaju podaci iz SC reda (da se podrži poravnavanje sa ulazom), i vraća se kontrola DND niti koja zatim šalje novu komandu VD modulu.

"DND_SYNC_MODE" režimu rada se vrši sinhronizacija sa ulaznim tokom podataka nakon čega može da se pokrene proces dekodovanja. Jedini očekivani prekid u ovom režimu je "VD_CMD_READY". VD proverava status izdvojenih zaglavlja i ako su sva potrebna zaglavlja izdvojena, posebno zaglavlje sekvence, izdvojeni podaci se čuvaju u promenjivih *DND_stream_t* strukture. Za svaku sliku uzima se SC iz *scQueue* liste i porede se neophodna polja sa zaglavlja izdvojenim od strane VD modula. Ako se desi da se poređena polja razlikuju više od jednog puta za redom, DND se vraća u "DND_STATE_IDLE" režim i postavlja "DND_PANIC" događaj. Ako se uspešno izdvoje potrebna zaglavlja i ako je sledeća slika za dekodovanje I slika, VD modulu se šalje "VD_DECODE_PICTURE" komanda i menja se režim rada DND rukovaca u "DND_STATE_PLAY".

"DND_STATE_PLAY" režim rada je osnovni režim rada u procesu dekodovanja i prikaza slike. U njemu se dešavaju "VD_CMD_READY" i "VD_PROC_DONE" prekidi. Kao i u "DND_SYNC_MODE" režimu izdvaja se SC i upoređuje sa odgovarajućim poljima zaglavlja izdvojenih od VD modula. Ako je sve u redu šalje se "VD_DECODE_PICTURE" komanda.

4.3.5 Rukovanje korisničkim podacima

Sakupljanje korisničkih podataka izdvojenih od strane VD modula se vrši nakon završetka procesa dekodovanja slike u slučaju kada nije došlo do prepunjavanja VD memorijskog bafera za smeštanje podataka. Za slučaj kada se VD bafer prepuni dešava se "VD_UDATA_EXCEEDS" prekid i tada se prekida proces dekodovanja da bi se sakupili izdvojeni podaci.

Sakupljeni podaci se čuvaju u kružnom baferu dok se ne pročitaju od strane višeg programskog nivoa. Da bi se obezbedilo da svaki blok korisničkih podataka bude dostupan koristi se *udQueue* lista *user_data_block_t* struktura (Tabela 21) gde se u strukturi nalaze parametri koji opisuju svaki blok izdvojenih podataka. U slučaju kada se podaci ne pročitaju od strane višeg programskog sloja odbacuju se "najstariji" upisani podaci u listi i na njihovo mesto se upisuju novi.

U okviru DND rukovaca ne vrši se dodatna obrada korisničkih podataka osim izdvajanje AFD deskriptora. Vrednost izdvojenog AFD deskriptora se čuva u okviru *DND_stream_t* strukture i poništavaju se pre početka svake nove sekvence.

Struktura	Opis	
<i>user_data_block_t</i>	Parametri korisničkih podataka.	
Polja	Opis	Tip
q	Zaglavlje liste kojoj pripada.	listMember_t
ud_address	Adresa u baferu za smeštanje.	uint32_t
ud_len	Dužina podataka.	uint32_t

Tabela 21 *user_data_block_t* struktura

4.3.6 Rukovanje greškama

Za slučaj kada se greška desi u toku procesa dekodovanja VD modul javlja prekid. Moguća su dva prekida:

- VD_WRONG_CMD

Ovaj prekid se dešava ako VD modul primi komandu od strane DND rukovaoca koju ne očekuje.

- VD_SET_ERROR

Dešava se ako se otkrije greška u samom procesu dekodovanja.

U oba slučaja se preskače slika u toku čijeg dekodovanja se desila greška i sakupljaju se korisnički podaci s tim da se ne vrši obrada AFD deskriptora.

Drugi tip greške koji može da se desi odnosi se na pogrešno postavljene vrednosti izdvojenih zaglavlja. Provera ispravnosti izdvojenih zaglavlja se vrši pre početka dekodovanja svake slike. Ako je otkrivena greška na ovom nivou DND rukovalac se prevodi u "DND_STATE_SYNC" režim rada za slučaj da je prag tolerancije greške mali u suprotnom greška se ignoriše.

5. Testiranje i verifikacija

5.1 Test okruženje

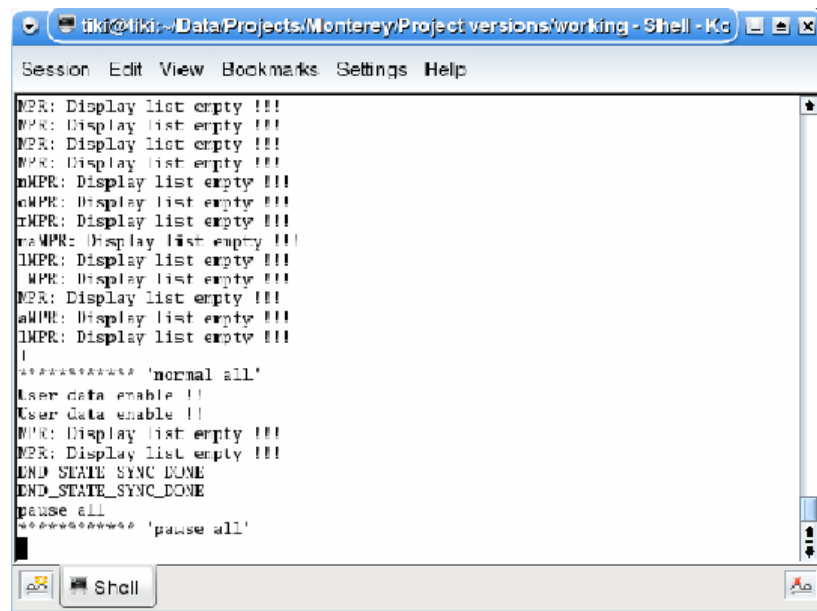
PC platforma na kojoj je vršeno testiranje ima sledeće karakteristike:

- Procesor Intel® Pentium® 4, sa taktom na 2GHz
- Matična ploča bazirana SiS® SiS648 integrisanom kolu
- 1GB DDR SDRAM memorije, sa taktom na 266MHz

Programska podrška za razvoj i testiranje:

- Fedora distribucija Linux OS kernel 2.6.17-1.2174_FC5
- X Window System Version 7.0.0
- gcc version 4.1.1
- GNU gdb Red Hat Linux (6.3.0.0-1.122rh)

Da bi se podržali različiti modovi rada DND rukovaoca i omogućile promene konfiguracije realizovan je mehanizam upravljanja radom DND rukovaca zadavanjem tekstualnih komandi. Komande se zadaju u Linux konzoli (Slika 17).



Slika 17 Linux konzola za unos tekstualnih komandi

Spisak podržanih komandi je naveden u tabeli 22. Prilikom zadavanja svake komande unosi se identifikator za tok podataka na koji se komanda odnosi.

Komanda	Opis
play	Pokretanje procesa dekodovanja.
stop	Zaustavljanje procesa dekodovanja.
data	Izdvajanje korisničkih podataka.
mode	Postavljanje moda rada.
ctrl	Postavljanje kontrolnih parametara.
normal	Normalan mod dekodovanja.
freeze	Prikazuje zadnju izdekodovanu sliku i preskače ostale.
pause	Prikazuje zadnju izdekodovanu sliku i preskače ostale.
IP	Dekoduju se samo I i P slike.
I	Dekoduju se samo I slike.
resync	Dekoduje se I slika i sinhronizuje sa sledećim GOP-om.

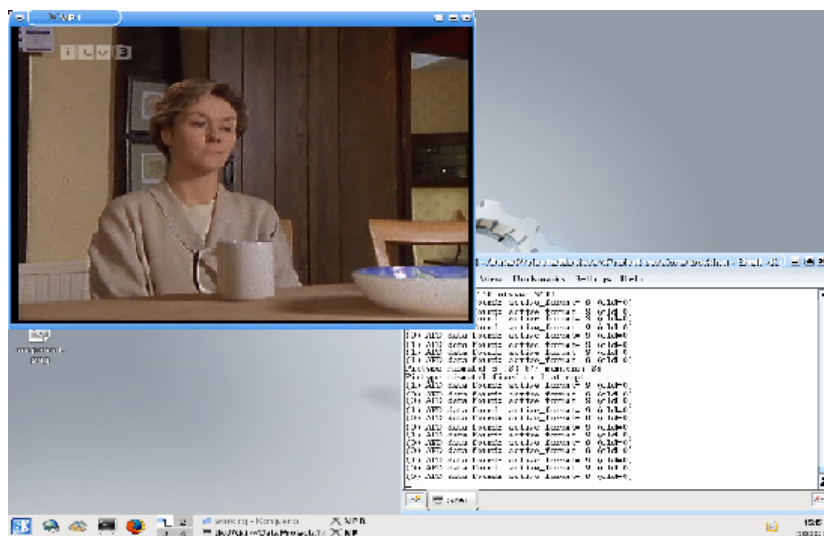
Tabela 22 Komande za testiranje

5.2 Proces testiranja

Proces testiranja se odvijao u dve faze:

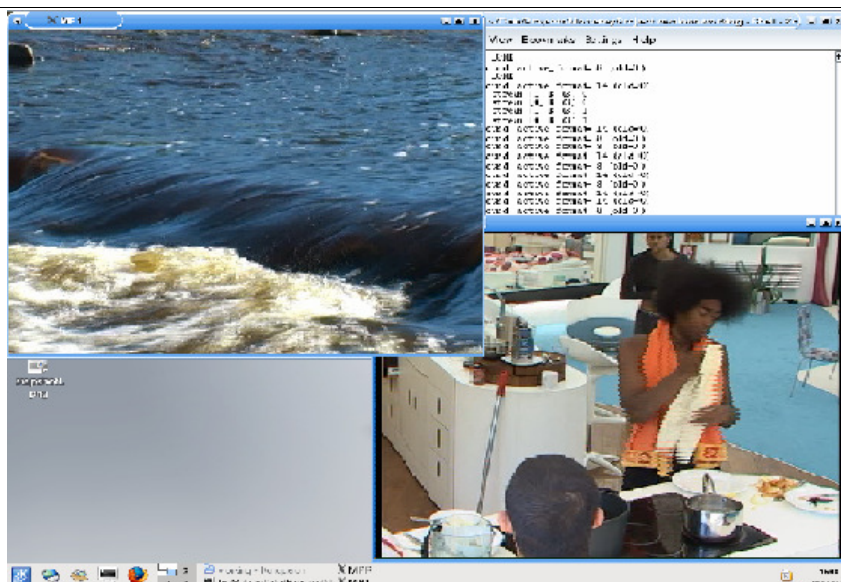
- Testiranje dekodovanja i prikaza jednog toka podataka (Slika 18)
- Testiranje dekodovanja i prikaza dva toka podataka (Slika 19)

Testiranja dekodovanja i prikaza jednog toka podataka vršeno je nad skupom ulaznih datoteka. U toku svakog testa menjani su modovi rada DND rukovaoca. Za potrebe testiranja oporavka od grešaka nastalih u procesu dekodovanja u okviru VD modula namerno su programski generisane greške.



Slika 18 Dekodovanje i prikaz jednog toka podataka

Testiranja dekodovanja i prikaza dva toka podataka vršeno je korištenjem istih datoteka za oba ulazna toka i korištenjem različitih ulaznih datoteka za svaki tok podatka. U toku svakog testa menjani su modovi rada DND rukovaoca. Postavljani su različiti modovi rada za svaki tok podataka posebno i menjani su konfiguracioni parametri DND rukovaoca.



Slika 19 Dekodovanje i prikaz dva toka podataka

5.3 Rezultati testiranja

Rezultati testiranja sa spiskom korištenih datoteka i kratkim opisom testa su navedeni u sledećoj tabeli:

Naziv ulazne datoteke	Opis testa	Jedan tok podataka	Dva toka podataka
afd.m2v	Promena vrednosti AFD deskriptora (sve moguće vrednosti za AFD)	Prošao	Prošao
udafd.m2v		Prošao	Prošao
DYNPICTS.m1v	Dinamička promena veličine slike.	Prošao	Prošao
MPEG12.m1v	MPEG-1 i MPEG-2 video.	Prošao	Prošao
PAN2HOR.m2v	"pan scan" vektori.	Prošao	Prošao
PULLDOWN.m2v	"top field first" i "repeat first fild"	Prošao	Prošao
s1.m2v	"interlaced" video.	Prošao	Prošao
s2.m2v		Prošao	Prošao
s3.m2v		Prošao	Prošao
s4.m2v		Prošao	Prošao
c.m2v	Različiti MPEG-2 video zapisi.	Prošao	Prošao
r.m2v		Prošao	Prošao
w.m2v		Prošao	Prošao
ch2.m2v		Prošao	Prošao
ch3.m2v		Prošao	Prošao
ch4.m2v		Prošao	Prošao
ch5.m2v		Prošao	Prošao

Tabela 23 Rezultati testiranja

6. Zaključak

DND rukovaoc je realizovan da bude sastavni deo programske podrške za rukovanje dekodovanjem i prikazom MPEG-2 video podataka visoke rezolucije (HD) ili standardnih digitalnih podataka niske rezolucije (SD) sa podrškom dekodovanja dva toka podataka u paraleli.

U osnovi DND vrši sinhronizaciju rada VD, MR i ulaznog programskog modula i obezbeđuje spregu sa višim programskim nivoima. Osnovni zadaci se svode na:

- Upravljanje radom VD modulom.
- Upravljanje radom MR modulom.
- Rukovanje video PTS/STC sinhronizacijom.
- Rukovanje dekodovanjem i prikazom dva toka podataka u paraleli.
- Upravljanje obradom grešaka i podrška za oporavak od nekih tipova grešaka.

6.1 Pravci daljeg razvoja

Prilikom relazacije DND rukovaoca vodilo se računa da rešenje bude primenjivo na fizičkoj platformi i stoga ova realizacija DND rukovaoca ostavlja dovoljno prostora za dalja usavršavanja i nadogradnju. Generalno pravci daljeg razvoja se odnose na:

- Implementaciju rukovaoca (eng. *driver*) fizičkim VD i MR jedinicama.
- Usaglašavanje sprežnih funkcija programskih VD i MR jedinica sa funkcijama rukovaoca fizičkim jedinicama.
- Usaglašavanje sprežnih funkcija DND rukovoca u odnosu na više programske slojeve.
- Integraciju DND rukovaoca u Linux OS.
- Testiranje u realnim uslovima rada na fizičkoj platformi.
- Implementacija naprednih algoritama za oporavak od grešaka.

Trenutna realizacija DND rukovaoca se oslanja na programske VD i MR jedinice tako da implementacija rukovaoca fizičkim jedinicama je preduslov za dalji razvoj. Prilikom realizacije rukovaoca fizičkim VD i MR jedinicama potrebno je voditi računa da sprežne funkcije rukovaoca fizičkim jedinicama budu u skladu sa funkcijama korištenim u programskoj simulaciji čime bi se obezbedilo da trenutna arhitektura DND rukovaoca ostane nepromenjena.

Nakon realizacije rukovaoca fizičkim VD i MR jedinicama neophodno je prilagoditi DND rukovaoc da koristi nove sprežne funkcije. Prilikom implementacije programskih jedinica vodilo se računa da se što približnije opiše rad fizičke platforme tako da ne bi trebalo da bude većih promena u ovom koraku razvoja.

Prilikom razvoja i testiranja DND rukovoca, DND sprežne funkcije nisu korištene od strane viših programskih nivoa. Iz tog razloga je potrebno nakon implementacije funkcija viših programskih slojeva, koje treba da upravljaju radom DND rukovaoca, izvršiti korekciju DND sprežnih funkcija i njihovo testiranje u skladu sa višim programskim slojevima.

Veoma bitan korak u daljem razvoju predstavlja integraciju DND rukovaoca u operativni sistem koji će da se izvršava na ciljnoj platformu.

Jedan od završnih koraka daljeg razvoja je integracija sa fizičkom platformom i testiranje rada u realnim uslovima. U ovom koraku razvoja najveći akcenat će biti stavljen na testiranje za različite režime rada i za različite ulazne tokove video podataka.

Trenutna realizacija rukovanja oporavkom od grešaka nastalih u procesu dekodovanja se zasniva na preskakanju slika na kojima je uočena greška ili ponovnoj sinhronizaciji sa ulaznim tokom podataka. Ovakav pristup rukovanju greškama nije dovoljan iz razloga što VD modul pruža dovoljno informacija o nastalim greškama tako da je ostavljena mogućnost ispravljanja greškaka na nivou DND rukovaoca korištenjem odgovarajućih algoritama.

7. Literatura

- [1] Information Technology – Generic Coding of Moving Pictures and Associated Audio, recommendation H.222.0, ISO-IEC 13818-1, Draft International Standard, 10. Jun 1994.
- [2] Chris Mlynarczyk, MPEG2 – The basics of how it works (MPEG tutorial 1)
- [3] A Guide to MPEG Fundamentals and Protocol Analysis (25W_11418_3), www.tektronix.com
- [4] Literatura za predavanja iz predmeta Programska podrška u televiziji i obradi slike, April 2005
- [5] Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Digital_television
- [6] Jedno rešenje implementacije sprege VoIP aplikacije sa TV uređajem, XIV Telekomunikacioni forum, Novembar 2006, Beograd, Tihomir Anđelić, Nikola Teslić, Branislav Čurčić, Zoran Marčeta, Vladimir Đurković, Mile Davidović

Jedno rešenje implementacije sprege VoIP aplikacije sa TV uređajem

Tihomir Anđelić, Nikola Teslić, Branislav Čurčić, Zoran Marčeta, Vladimir Đurković, Mile Davidović

Sadržaj — Prikazano je jedno rešenje nadogradnje programske podrške za analogni TV prijemnik zasnovan na VCT-i integrisanom kolu sa VoIP aplikacijom (u daljem tekstu TV Skype aplikacija) korišćenjem UACB razvojne platforme i SkypeAPI programske sprege za Windows operativni sistem. Date su osnove karakteristike sistema i kratak opis realizovane aplikacije.

Ključne reči — I²C, MINTS, Skype, TVSkype, UACB, VCT.

I. Uvod

A bi ostali konkurentni na tržištu proizvođači televizora moraju konstantno unapređivati svoje proizvode usavršavanjem kvaliteta slike i nadogradnjom sa novim funkcijama. Ovaj rad predstavlja implementaciju jedne nove funkcionalnosti TV uređaja koja obezbeđuje korisniku da pored standardnih funkcija na TV uređaju koristi VoIP tehnologiju.

Ograničenja pri implementaciji ove funkcionalnosti se odnose na tehnička ograničenja TV uređaja, na stabilnost nove TV aplikacije i pre svega na prilagođavanje jedne standardne PC aplikacije prosečnom TV korisniku.

Rad je delimično podržan u okviru projekta TR-6136B Ministarstva za nauku i zaštitu životne sredine Republike Srbije.

Tihomir Anđelić, Autor, Fakultet tehničkih nauka u Novom Sadu, Srbija; e-mail: (e-mail: tihomir.andjelic@micronasnit.com).

Nikola Teslić, Koautor, Fakultet tehničkih nauka u Novom Sadu, Srbija; e-mail: nikola.teslic@micronasnit.com.

Branislav Čurčić, Koautor, Institut za Informacione Tehnologije, Novi Sad; e-mail: branislav.curcic@micronasnit.com.

Problemi koji nastaju iz ograničenja TV uređaja su nedostatak radne memorije, brzine procesora i značajno različite korisničke sprege ka aplikaciji kod TV uređaja i personalnog računara. Takođe dodavanje nove funkcije TV uređaju koja podrazumeva komunikaciju sa drugim uređajima stvara se potencijalni izvor nestabilnosti što je za sistem kao što je TV neprihvatljivo i stoga je tokom implementacije neophodno voditi računa o ovom segmentu.

Na kraju prilagođavanje PC aplikacije TV korisniku predstavlja prepreku zbog toga što se profil prosečnog korisnika TV uređaja značajno razlikuje od prosečnog korisnika PC računara. To znači da se grafička sprega za korišćenje TV Skype aplikacije mora integrisati u sistem TV menija bez narušavanja postojećeg koncepta.

Dodatna prepreka za realizaciju TV Skype aplikacije je tehničke prirode i odnosi se na pojavu akustičnog eha. Problem akustičnog eha i smetnji koja se javljaju usled udaljenosti korisnika od mikrofona rešavaju se posebnim audio algoritmima koji nisu tema ovog rada.

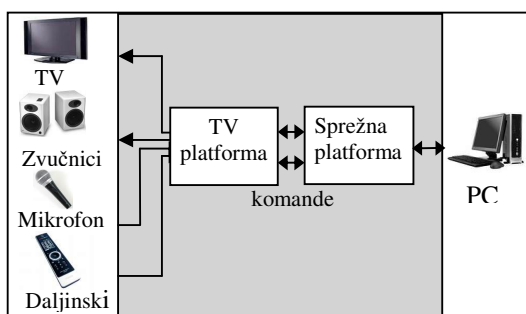
II. Opis sistema

TV Skype sistem podržava VoIP funkcionalnost Skype programskog rešenja namenjenog za PC. Rešenje obezbeđuje jednostavnu spregu korisnika sa aplikacijom koji će korišćenjem daljinskog upravljača kontrolisati aplikaciju kao deo funkcionalnosti TV seta.

Fizičku arhitekturu sistema čine:

- TV razvojna platforma.

- Fizička sprega između TV-a i PC-a.
- Personalni računar.



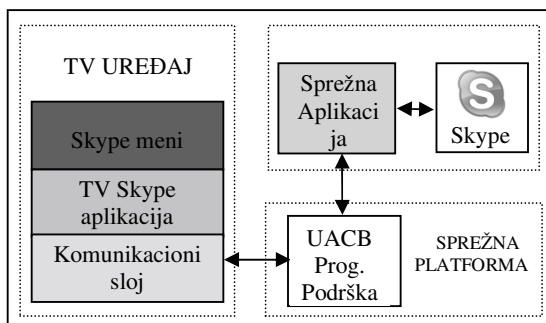
Sl. 1. TV Skype sistem

Za TV platformu u ovoj implementaciji korištena je VCT razvojna platforma (TV razvojna platforma za analognu televiziju, VCTi verzija integrisanog kola).

Programsku podršku se sastoji iz dve celine:

- Programska podrška na PC strani koju čine Skype aplikacija i sprežna aplikacija za povezivanje sa TV platformom.
- Proširena programska podrška za TV platformu.

Svaki segment rešenja realizovan je posebno i omogućeno je testiranje njegove funkcionalnosti nezavisno od ostalih segmenata koji sačinjavaju sistem.



Sl. 2. Šema programske podrške

III. Komunikacija

Za komunikaciju između aplikacije koja se izvršava na ciljnoj platformi i PC dela aplikacije projektovan je komunikacioni protokol koji zadovoljava neophodne kriterijume komunikacije posredstvom UACB razvojne ploče a uz korišćenje USB komunikacije i I²C magistrale.

Komunikacioni protokol

Protokol obuhvata dvosmerni prenos komandnih poruka od PC-a prema VCT platformi. Analogni signali se prenose nezavisno od komandi a obrađuju se na samoj UACB platformi.

Procedura za inicijalizaciju i režimi rada UACB platforme

Prilikom inicijalizacije sistema neophodno je da UACB čip bude vodeći uređaj na magistrali. Tokom normalnog režima rada VCT je vodeći uređaj na

magistrali ali tek nakon što UACB signalizira da je spreman da startuje u normalnom modu kada radi kao prateći uređaj. Nakon inicijalizacije, UACB treba da signalizira da je inicijalizacija uspešno izvršena. Posle uspešno izvršene inicijalizacije dozvoljava se normalan režim rada ostalim uređajima u sistemu.

UACB čip ima dva režima rada:

- Režim prenosa analognih signala (zvuka), u kome se na UACB spregu povezuju analogni ulazi i izlazi. U ovom režimu se obrađuju signali primljeni od PC-a i prosleđuju na VCT platformu.
- USB režim prenosa komandi, u kome se prosleđuju VCT platformi komande primljene od strane PC-a preko I²C magistrale.

Komunikacija od VCT prema UACB

Pošto je VCT vodeći uređaj na I²C magistrali, samo on može da inicira komunikaciju kada je to potrebno. Da bi se izbegla mogućnost gubitka podataka, neophodno je poštovati sledeća pravila:

- Pre slanja nove komande VCT mora da proveriti komandni bajt. Ako nije postavljena nova vrednost, to znači da je UACB spreman da obradi sledeću komandu, i VCT može da startuje sa upisom nove komande. Ovo je neophodno zato što je vreme potrebno da se podaci pošalju prema PC-u promenljivog trajanja i može se desiti da UACB nije na vreme završio slanje podataka. Da bi se izbegao konflikt ovo pravilo mora da se primenjuje za sve komande koje se šalju od VCT-a.

- Komanda će od UACB biti prosleđivana u dva koraka. U prvom koraku potrebno je upisati podatke u bafer za podatke. Ovaj korak će biti preskočen ako se šalje samo komanda bez podataka (kao što je promena režima rada). U drugom koraku se upisuje komandni bajt.

- Nakon upisa komandnog bajta UACB će početi sa obradom. Komandni bajt neće bit promenjen sve dok se ne završi proces slanja. Kada UACB završi sa obradom isprazniće komandni bajt i dozvoliti da VCT upiše sledeću komandu.

Ovaj komunikacioni metod omogućava minimalnu potrošnju resursa i zahteva od VCT-a da proverava komandni bajt pre samog slanja. Komande sa odgovarajućim komentarom su prikazane u tabeli 1.

TABELA 1: PRIMERI KOMANDI.

Komanda	Komentar
NAME	Slanje imena uređaja sa kojim se komunicira.
PROTOCOL	Verzija Skype protokola koji se koristi.
SET USERSTATUS OFFLINE	OFFLINE režim rada Skype aplikacije.
SET USERSTATUS ONLINE	ONLINE režim rada.
SET USERSTATUS AWAY	AWAY režim rada.
SET USERSTATUS BUSY	BUSY režim rada.
SEARCH USERS	Zahtev za listom korisnika.
CALL user	Pozivanja korisnika.
SET CALL user STATUS FINISHED	Kraj razgovora.

Komunikacija od UACB prema VCT

UACB radi kao prateći uređaj na I²C magistrali i ne može neposredno da započne razmenu podataka. Iz tog razloga potrebno je zahtevati razmenu podataka od strane VCT platforme tako što će se postavljati vrednost odgovarajuće nožice VCT kontrolera (u nastavku teksta UACB INT). Kada VCT otkrije da je UACB INT postavljen započinje sesiju I²C razmene prema sledećem redosledu:

- UACB upisuje podatke u bafer i postavlja statusni bajt. UACB ne može izmeniti jednom postavljen statusni bajt sve dok VCT ne pročita podatke i postavi ispravnu vrednost u statusni bajt.
- UACB INT se postavlja na logičku jedinicu.
- VCT, kada otkrije da je UACB INT postavljen čita podatke iz statusnog bajta.
- Zavisno od statusnog bajta postoje dva slučaja:
 - Podaci u statusnom bajtu su neispravni, što znači da UACB nije promenila podatke od zadnje obrade. U ovom slučaju VCT će ignorisati podatke.
 - Podaci u statusnom bajtu su ispravni VCT će ih obraditi i zavisno od podataka pročitane informaciju primljenu od UACB.
 - Nakon završetka obrade podataka, VCT upisuje status bajt i postavlja vrednost sedmog bita.
 - Kada UACB otkrije da je postavljena vrednost odgovarajućeg bita statusnog bajta može da se postavi UACB INT na logičku nulu.

Komunikacija od UACB prema PC

UACB koristi USB standard za komunikaciju sa PC-om. Sve komande definisane prema UACB standardu mogu da se koriste za komunikaciju. Komande koje nisu definisane u standardu obrađuju se u zavisnosti od koda a u okviru aplikacije koja komunicira sa Skype programskom spregom na PC-u.

Komunikacija od PC prema UACB

Podaci koji dođu od PC-a preko UACB sprege biće prosledjeni VCT strani bez obrade UACB platformi. Kada je razmena podataka prema VCT obavljena, UACB izveštaj "Output report processing done" postavlja bit signal PC-u tako da aplikacija može da pređe na slanje sledećeg dela podataka. Izlazni izveštaj je veličine 34 bajta, i proziva se svakih 10ms.

Format podataka koji se prenose

Za oba smera prenosa format podataka je isti radi njihove lakše obrade i jednostavnijeg kodovanja. Dužina paketa koji se šalje je 34 bajta s tim da aplikacija koja šalje podatke sa strane PC-a zbog formata poruke koja se razmenjuje dodaje na početak paketa još jedan bajt čija je vrednost "0".

Paket sadrži tri dela sa podacima. Prvi deo je kod poruke koja se prenosi i njegova dužina je jedan bajt, drugi parametar poruke jeste dužina sadržaja izražena u bajtovima, dužine jedan bajt, a odnosi se na korisni sadržaj koji se prenosi, treći parametar poruke su sami podaci dužine 32 bajta što odgovara najvećoj mogućoj dužini Skype korisničkog imena.

IV. Opis realizacije

Programsko rešenje TV Skype aplikacije sačinjavaju tri segmenta od kojih se svaki izvršava na zasebnoj platformi i svi međusobno komuniciraju preko definisanog protokola. Prilikom realizacije vođeno je računa da svaki segment rešenja može da radi samostalno i da pri tome ne ugrožava ostatak sistema. Ovim je postignuto da otkazom ili isključenjem nekog od segmenata sistema ne dolazi do otkaza celog sistema nego samo do prekida komunikacije. Nakon vraćanja u sistem nekog od segmenata, sistem nastavlja sa radom bez potrebe za ponovnom inicijalizacijom ili nekom drugom vrstom akcije od strane korisnika.

Sprežna aplikacija za povezivanje Skype aplikacije sa UACB platformom

Za pokretanje sprežne aplikacije nije potrebna njena instalacija već samo pokretanje izvršne datoteke (ovaj deo može po želji korisnika i da sa automatizuje). Podrazumeva se da na računaru postoji instaliran Skype i da je korisnik prijavljen na isti.

Skype API je spregra realizovana za pristup Skype funkcijama od drugih aplikacija, namenjenih za poziv Skype kontakata, slanje tekstualnih poruka, rukovanje korisničkim profilom, itd. U ovoj sprezi treća aplikacija se ponaša kao kontrolna strana i posredno rukuje funkcijama Skype aplikacije (podržanim u Skype API). API je realizovan na principu jednostavne razmene tekstualnih poruka između Skype-a i uređaja.

Kada klijentska aplikacija otpočne komunikaciju sa Skype aplikacijom posredstvom API-a, tada Skype postavlja UACB kao audio uređaj. Kada komunikacioni nivo između klijentskog uređaja i Skype aplikacije prestane sa radom, Skype menja audio uređaj na prethodno korišteni. Svaki put kada neka aplikacija pokuša da pristupi Skype API delu pojavljuje se dijalog u okviru koga korisnik treba da dozvoli aplikaciji upotrebu Skype API-a.

TV Skype aplikacija na VCT platformi

Zbog ograničenih resursa VCT platforme u pogledu raspoložive memorije, brzine procesora i drugih parametara, realizacija aplikacije je u skladu sa sistemom i ne narušava njegovu funkcionalnost. Realizovan je mehanizam koji dozvoljava jednostavno dodavanje i uklanjanje novih funkcija i modula neophodnih za realizaciju aplikacije.

Za potrebe komunikacije između VCT i UACB platforme konfigurisani su odgovarajući ulazni i izlazni konektori. Od konektora inicijalizovane su dve ulazne nožice od kojih se jedna koristi kao signal reseta, aktivan na logičkoj nuli a drugi kao signal za ulazni GPIO kontrolisan od UACB platforme. Konfigurisan je i jedan izlazni konektor za signal reseta koji se koristi za resetovanje UACB čipa. Konektori za I²C magistralu su aktivni nakon inicijalizacije platforme i ne zahtevaju dodatnu inicijalizaciju. U okviru postojećih modula MINTS operativnog sistema dodate su funkcije za rukovanje i konfiguraciju navedenih konektora. Za implementaciju grafičke korisničke sprege modifikovani su postojeći meniji dodavanjem novih menija za

rukovanje TV Skype funkcijama. Za ispis liste korisnika sa aktivnim statusom realizovan je dinamički meni koji prikazuje listu od 10 kontakata dok se ostali kontakti čuvaju u lokalnoj listi na PC strani aplikacije. Sve promenljive i funkcije koje se neposredno koriste za TV Skype aplikaciju odvojene su u poseban modul koji se po potrebi može isključivati ili uključivati. Za rukovanje dolaznim porukama sa UACB platforme, što se pre svega odnosi na promenu stanja ulaznog GPIO konektora, implementiran je mehanizam za njegovo kontinuirano vremensko proveravanje i pozivanje funkcije za obradu odgovarajuće promene.

Inicijalizacija sistema

EPROM memorije koje se nalaze na obe platforme imaju istu fizičku adresu na I²C magistrali što predstavlja problem kada neki od uređaja pokuša da pristupi navedenoj adresi. Drugi problem je što UACB čip mora da bude vodeći uređaj na I²C magistrali za vreme inicijalizacije što nije izvodljivo ako se VCT platforma inicijalizuje prva.

Problem je rešen uvođenjem mehanizma dvojnog reseta. Nakon što započne inicijalizaciju, VCT platforma resetuje UACB čip i drži ga u resetu za vreme inicijalizacije. Nakon uspešne inicijalizacije VCT dozvoljava inicijalizaciju UACB platforme i za to vreme ne radi ništa na I²C magistrali dok UACB čip ne završi inicijalizaciju i postavi signal reseta na logičku jedinicu nakon čega je sistem uspešno inicijalizovan. Ovim je rešen problem za sve moguće redoslede inicijalizacije radnih platformi bez opasnosti za otkaz sistema ako se neki od segmenata isključi i ponovo uključi u sistem.

V. Testiranje

Svaka celina sistema je testirana posebno i zavisno od njene prirode primenjene su odgovarajuće metode za njeno testiranje. Nakon sklapanje svih delova sistem je testiran u celini.

U toku razvoja aplikacije na PC strani koja povezuje SkypeAPI sa UACB platformom preko HID nivoa simulirana su ostala dva nivoa. U okviru same aplikacije implementirana je korisnička sprega pomoću koje je bilo dozvoljeno ručno unošenje komandi za simulaciju realne situacije i ispis poruka primljenih od odgovarajućeg nivoa. Pored testiranja uspešnosti razmene poruka i njihove interpretacije na ovom nivou komunikacije testirani su slučajevi nepredviđenih otkaza sistema pri čemu se pre svega misli na uključivanje i isključivanje UACB platforme u toku rada. Aplikacija je uspešno zadovoljila sve kriterijume i uspešno se oporavila od pojavljivanja nekonzistentnih situacija

VCT deo rešenja je testiran interaktivno simulacijom akcija korisnika u okviru menija. Simulirane su sve predviđene situacije i ponašanje sistema u njima. Posebno deo testiranja se odnosio na uspešnost inicijalizacije sistema prilikom uključivanja radnih platformi pri čemu su uređaji (UACB i VCT platforme) u različitom redosledu uključivani u sistem da se proverí uspešnost njihove inicijalizacije.

Za testiranje UACB platforme korištena je «Visual I2C» aplikacija pomoću koje je se pristupalo memoriji na UACB platformi i proveravana valjanost podataka sa unapred definisanih adresa.

VI. Zaključak

Ovaj rad je primer kako se funkcije TV uređaja mogu proširiti sa jednom nestandardnom funkcijom kao što je VoIP. Skype TV aplikacija predstavlja dobru polaznu osnovu za dalji razvoj i nadogradnju.

Postoji mogućnost proširenja postojeće aplikacije dodavanjem novih funkcija kao što su razmena tekstualnih poruka (eng. *chat*) i prenos video signala. Pored proširenja postojeće aplikacije moguće je relativno lako izmeniti sistem uvođenjem drugačije sprege između PC-a i TV uređaja (serijska komunikacija, WLAN, Bluetooth, DECT...)

Ovaj model rešenja je lako primenjiv i na druge TV platforme i slične sisteme.

LITERATURA

- [1] VCTIApplicationboard_QFP144_V0200_schematic-color.pdfVMU DVR.pdf.
- [2] vct49_48xyi_1ai_2004_11_23.pdfMICTOSDesignSpec.pdf
- [3] Skype_api_doc_en.pdfISO-IEC 13818-2.pdf. Available:<http://developer.skype.com/Docs/ApiDoc/FrontPage>.
- [4] UACB Standard Application Board v0101.pdf.
- [5] VCTi_ControllerArchitecture.pdf.

ABSTRACT

This paper presents one implementation VoIP application interface for VCT Controller using UACB Micronas IC and SkypeAPI program interface for Windows OS.

ONE IMPLEMENTATION OF VoIP APPLICATION INTERFACE FOR VCTP CONTROLLER USING SKYPE API

Tihomir Andelić, Nikola Teslić, Branislav Čurčić, Zoran Marčeta, Vladimir Đurković, Mile Davidović