

# Jedno rešenje proširenja TV arhitekture za podršku televiziji visoke definicije

Milan Z. Bjelica, *Fakultet tehničkih nauka, Novi Sad*  
Milan Savić, Tatjana Aleksić, Tomislav Maruna, *MicronasNIT, Novi Sad*

**Sadržaj** — Cilj rada je da predstavi koncept, izradu i primenljivost jednog pristupa proširenju fizičke arhitekture i programske podrške digitalne TV platforme za omogućavanje prikazivanja video i audio tokova koji se koriste u televiziji visoke definicije (*HDTV*). Dat je opis problema proširenja sa stanovišta ponovne primenljivosti postojećih platformi i brzine razvoja. Ukratko, predstavljen je koncept rešenja i neki detalji izrade.

**Ključne reči** — *HDTV, H.264, MPEG-4, koprocesor, komunikacija, televizija.*

## I. UVOD

ARHITEKTURA digitalnih elektronskih sistema koji se nalaze u digitalnim TV prijemnicima predstavlja skup većeg broja visoko integrisanih blokova fizičke arhitekture za različite primene u digitalnoj televiziji (*DTV*). Neki od prisutnih blokova su video i audio dekoderi, podsistemi za prihvatanje i slanje audio i video tokova, grafički podsistemi za primenu algoritama obrade video signala ili prikaz grafičkih elemenata, različite ulazno-izlazne sprege kao što su *USB, I2C, UART, GPIO* priključci, *SmartCard* i sl. Upravljanje ovim komponentama, njihova kontrola i komunikacija obavlja se posebnim upravljačkim procesorom.

Trend visoke integracije u industriji ugrađenih (*embedded*) sistema postavlja kao cilj smeštanje svih blokova fizičke arhitekture sistema unutar jednog integrisanog kola (*System On Chip - SOC*). Zahtevi za proširenjima takvih sistema najpre dovode do razvoja dodatnih, izdvojenih blokova koji ostvaruju komunikaciju sa osnovnim integrisanim kolom, da bi se naknadno svi blokovi integrisali unutar istog kola. Ovakvi sistemi su pogodni za verifikaciju zahvaljujući mogućnosti izvedbe na *FPGA (Field-programmable gate array)*.

Ovaj rad je delimično finansiran od Ministarstva za nauku Republike Srbije, projekat 11005, od 2008. god.

Milan Z. Bjelica, Fakultet tehničkih nauka, Univerzitet u Novom Sadu, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija; (telefon: 381-21-4801112; faks: 381-21-450721; e-mail: milan.bjelica@rt-rk.com).

Milan Savić, MicronasNIT, Institut za informacione tehnologije, Fruškogorska 11, 21000 Novi Sad, Srbija; (e-mail: milan.savic@micronas.com).

Tatjana Aleksić, MicronasNIT, Institut za informacione tehnologije, Fruškogorska 11, 21000 Novi Sad, Srbija; (e-mail: tatjana.aleksic@micronas.com).

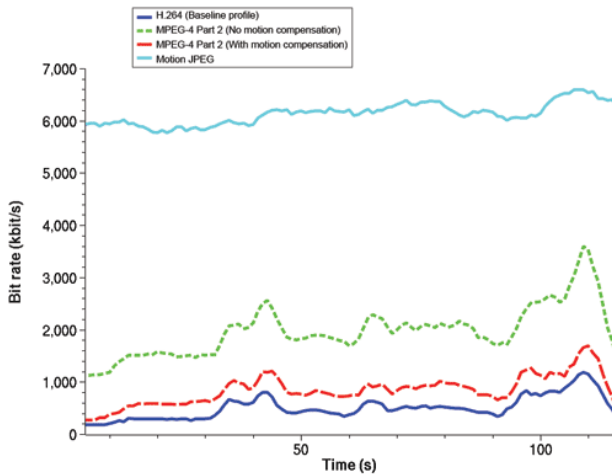
Tomislav Maruna, MicronasNIT, Institut za informacione tehnologije, Fruškogorska 11, 21000 Novi Sad, Srbija; (e-mail: tomislav.maruna@micronas.com).

Dinamičan napredak tehnologije i potreba za primenom novih standarda u projektovanju uređaja, nameće zahteve za brzim projektovanjem proširenja, uz zadržavanje kompatibilnosti sa ranijim rešenjima i ponovnu iskoristivost već razvijene programske podrške. Pojava novih formata slike i zvuka u *HDTV* (standard *DVB-S2*) omogućuju korisnicima da prate TV program izuzetnog kvaliteta (rezolucija slike 1920x1080, 6-kanalni audio). Da bi se omogućio prikaz ovih formata na postojećoj TV arhitekturi, neophodno je istu proširiti novim blokovima fizičke arhitekture, kao što je video dekodera za *H.264* format kompresije slike i audio dekodera za *AC3* format zvuka. Skup dodatnih blokova isporučuje se u vidu jednog integrisanog kola sa posebnim procesorom za nadzor i upravljanje. Problem proširenja svodi se na sledeće probleme: povezivanje postojećeg upravljačkog procesora osnovnog integrisanog kola sa koprocesorom dodatnog integrisanog kola i obezbeđivanje pouzdane upravljačko - kontrolne komunikacije, (pre)usmeravanje audio i video tokova traženih formata ka sprežnim jedinicama tog kola, omogućavanje transparentnog iniciranja memorijske slike koprocesora (*boot*), kao i transparentna integracija dodatka u pogledu programske podrške i minimalan skup izmena programske podrške osnovnog integrisanog kola.

## II. TEORIJSKE OSNOVE

Pod televizijom visoke definicije podrazumeva se sistem digitalne video difuzije *DVB (Digital Video Broadcasting)* sa većom rezolucijom od tradicionalnih TV sistema. Zbog velike količine podataka koji se moraju preneti, televizija visoke definicije koristi digitalni prenos podataka, uz potrebu za korišćenjem poboljšanih algoritama kompresije slike i zvuka. *H.264* format kompresije (deo *MPEG-4* skupa algoritama) koristi do 16 referentnih okvira (*frame*) za predviđanje, što omogućuje izuzetno dobru kompresiju video materijala koji se ponavlja manjom učestanošću.

*H.264* takođe omogućuje kompenzaciju pokreta uz promenljivu veličinu bloka, a koristi i višestruke vektore pomeraja za čuvanje informacije o pomeraju bloka, što veoma smanjuje potreban intenzitet bita (*bit-rate*). Takođe, koder može da utiče na faktore predviđanja u zavisnosti od brzine scene ili tipa promene koja se dešava. Za kompresiju bez gubitaka koristi entropijsko kodovanje, oslanjajući se na verovatnoću pojave sintaksnih elemenata. Dijagram na kome se vidi intenzitet bita utrošen za *H.264* u odnosu na ranije kompresione algoritme prikazan je na Slici 1.



Slika 1 – Intenzitet bita H.264 kompresije u poređenju sa drugim algoritmima

AC3 format kompresije zvuka omogućava šestokanalni zvučni zapis po standardima *DolbyDigital*.

### III. OPIS FIZIČKE ARHITEKTURE I PROGRAMSKE PODRŠKE CILJNIH PLATFORMI

U okviru projekta korišćene su TV platforme sa 32-bitnim upravljačkim procesorom. Osnovna platforma sadrži integrisano kolo sa blokovima fizičke arhitekture za primene u digitalnoj televiziji. Opremljena je *MIPS* procesorom za upravljanje i kontrolu različitih blokova. Platforma omogućava povezivanje sa bilo *WXGA* ili *HD* panelima, sa podrškom za sliku u slici (*Picture in Picture - PiP*), prikaz dve slike istovremeno (*Picture and Picture - PaP*), prikaz slike zajedno sa grafičkim elementima ili tekstom (*Picture and Graphics - PaG*, odnosno *Picture and Text - PaT*). Programska podrška za *MIPS* zasnovana je na *Linux* operativnom sistemu sa definisanom *LinuxTV* programskom spregom kojom se ostvaruje funkcionalnost za potrebe digitalne televizije po *DVB (Digital Video Broadcasting)* standardima. Unutar programske podrške izrađeni su i odgovarajući rukovaoci za sve blokove fizičke arhitekture osnovnog integrisanog kola. Opšta struktura programske podrške predstavlja video i audio dekodere kao uređaje u sistemu datoteka, npr:

```
\\dev\drv\adapter0\video0
```

Sve operacije nad uređajima (postavljanje parametara, start, stop i sl) obavljaju se korišćenjem uobičajenih *POSIX IO* funkcija za otvaranje, pisanje, čitanje, zatvaranje i kontrolu (*open, read, write, close, ioctl*). Za osnovnu platformu, pored rukovaoca u jezgru operativnog sistema, razvijena je i kompletna programska podrška u korisničkom prostoru (*DTV* međusloj i korisnička aplikacija za upravljanje TV prijemnikom – izmena kanala, meniji i sl).

Integrisano kolo sa podrškom za prihvatanje, dekodovanje i isporuku video i audio sadržaja u formatima *H.264* i *AC3* za podršku televiziji visoke definicije sadrži blokove fizičke arhitekture za dekodovanje ovih formata, prihvatanje i slanje audio i video tokova i sl. Ono u sebi takođe sadrži *MIPS* kao upravljački koprocesor, koji se dodatno koristi i za dekodovanje audio sadržaja. Video dekodovanje obavlja se posebnim *MIPS* procesorom. Osnovna primena

koprocorskog sistema je isključivo za dekodovanje audio i video sadržaja, bilo da se isti isporučuje direktno, putem ulaza transportnog toka (*TS input*), ili se usmerava preko *External Bus Interface (EBI)* sprežnog podsistema.

Programska podrška upravljačkog procesora na osnovnoj platformi, zbog specifičnih vremenskih zahteva koje nameće audio i video obrada, koristi operativni sistem realnog vremena.

### IV. KONCEPT PROŠIRENJA

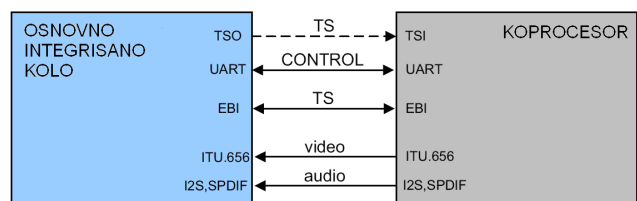
#### A. Komunikacija

Komunikacija na relaciji osnovna platforma - koprocesor u smislu upravljačkih poruka obavlja se korišćenjem *UART* serijske sprege. *UART* se koristi u osnovnoj postavci, sa jednom linijom za prijem (*Rx*) i jednom za slanje podataka (*Tx*). Kontrolna komunikacija zahteva visoku pouzdanost u slanju i prijemu podataka, odnosno, nedopustiv je gubitak podataka pri prenosu. Takođe, neophodno je omogućiti protokol sa povezivanjem (*connection-oriented*) kako bi se obezbedio kontekst za pouzdan prenos. Usvojeni model za komunikacioni protokol bazira se na *peer-to-peer* komunikaciji sa povezivanjem rukovanjem (*handshake*). Obe strane su ravnopravne u komunikaciji, mogućnost slanja i primanja poruka je asinhrona, i može se inicirati sa bilo koje strane.

#### B. Usmeravanje audio i video tokova

U okviru projekta obrađen je osnovni *use-case* scenario, gde su video i audio podaci uvek prisutni na *TS* ulazu koprocesorskog sistema. Sa strane osnovne platforme će, u slučaju da ne podržava format slike i zvuka koji je zahtevan, biti poslat komunikacioni zahtev koprocesoru za početak dekodovanja, uz naknadnu razmenu neophodnih parametara o formatu slike. Nakon prijema parametara, osnovna platforma stara se o pravilnom prikazu izlaznog toka, baš kao da je reč o lokalno dekodovanom toku.

Način povezivanja sa prikazom video i audio sprege prikazan je na Slici 2.



Sli. 2. Sprega na relaciji osnovno integrisano kolo - koprocesor

#### C. Iniciranje koprocesora

S obzirom da je koprocesorski sistem dodatno integrisano kolo, neophodno je izvršiti *boot* programske podrške za *MIPS* koprocesor, sa strane osnovne platforme. Ovo će se obaviti punjenjem datoteke sa memorijskom slikom iz *Linux* sistema datoteka sa osnovne platforme, preko *EBI* sprege, u radnu memoriju koprocesora. Osnovna kontrolna komunikacija za ove operacije obaviće se korišćenjem *UART* sprege.

#### D. Programska podrška na osnovnoj platformi

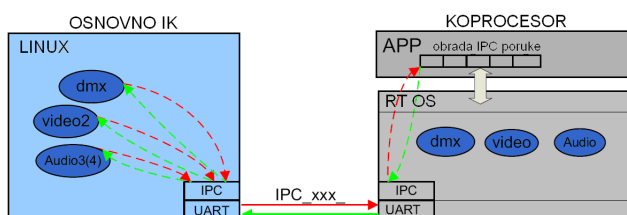
Jedan od osnovnih zahteva pri pisanju programske podrške za *Linux* na osnovnoj platformi je transparentnost. U tom smislu, aplikacije u korisničkom prostoru treba da vide koprocesorski sistem kao skup dodatnih uređaja u sistemu datoteka. Tako se koprocesorski video dekodier predstavlja kao *video2* uređaj (uz postojeće, lokalne, *video0* i *video1*), dok su dve instance audio dekodera označene sa *audio3* i *audio4*. Aplikacije koriste ove uređaje baš kao da je reč o lokalnim uređajima. Neophodna proširenja, dakle, odnose se najpre na izradu komunikacionih protokola na transportnom nivou i nivou sesije, što omogućuje (1) pouzdan prenos podataka i (2) formatiranje poruka u skladu sa *DVB* potrebama, te na pisanje rukovaoca za koprocesorske uređaje u okviru rukovaoca jezgra operativnog sistema osnovne platforme.

Svaki *DVB* rukovalac opisan je posebnom strukturom koja, između ostalog, sadrži pokazivače na sve neophodne funkcije, npr. za početak i kraj reprodukcije, postavljanje *PID* (*Packet identification*) identifikatora pre dekodovanja i sl. Ove funkcije su povezane sa apstrakcijom odgovarajućeg uređaja u *Linux* sistemu datoteka, pa se pozivaju pri pozivu *POSIX* rutina iz korisničkog prostora. Koprocesorski rukovaoci razlikuju se od lokalnih po tome što, umesto da pozivaju odgovarajuće funkcije nižeg nivoa za kontrolu blokova osnovne platforme, šalju komunikacioni zahtev unapred određenog formata koprocesorskoj strani. Niti zadužene za praćenje asinhronih audio i video događaja, koje pojedini blokovi mogu objaviti, proširene su, tako da prihvate događaje koje objavljuje komunikacioni protokol, a koje se odnose na asinhronu događaje na koprocesorskoj strani.

#### E. Programska podrška koprocesora

Za koprocesor, pored izrade komunikacionog protokola, potrebna je aplikacija za praćenje komunikacionih događaja. Zadatak ove aplikacije je da izdvaja poruke pristigle sa osnovne platforme, i da, u zavisnosti od njihovog formata, pozove odgovarajuće funkcije aplikacionog sloja kojima se pokreće i zaustavlja reprodukcija slike i zvuka i sl. Takođe, treba proširiti obrađivače događaja koji pristižu od strane ostalih koprocesorskih blokova, tako da omogućuje upućivanje asinhronog komunikacionog obaveštenja osnovnoj platformi sa neophodnim podacima (npr. podaci o video formatu nakon uspešnog prepoznavanja od strane video dekodera).

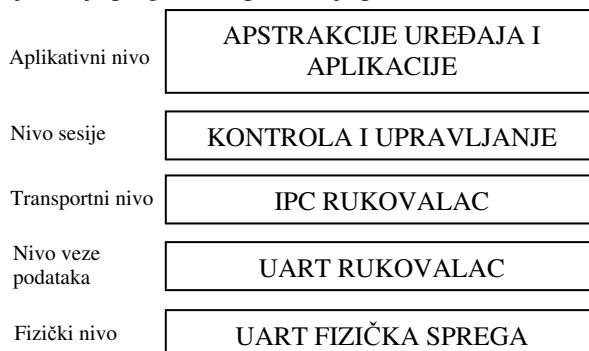
Koncept programske podrške prikazan je na slici 3.



Sl. 3. Povezivanje i apstrakcija u programskoj podršci

#### V. NEKI DETALJI IZRADE

Komunikaciona programska podrška izrađena je na ISO nivou veze, transporta, sesije i aplikacije. Nivo veze je prvi nivo programske podrške koji se oslanja na fizičku arhitekturu i upravlja samom *UART* komunikacijom. *IPC* protokol se nalazi na transportnom nivou, dok se na nivou sesije nalazi upravljački protokol za izdavanje stvarnih naredbi. Na nivou aplikacije, rukovaoci i programi koji ih koriste služe se primitivama nižih komunikacionih nivoa. Hijerarhija programske podrške je prikazana na Slici 4.



Slika 4 – Hijerarhija programske podrške

#### A. IPC Protokol

U svrhu implementacije komunikacionih zahteva, izrađen je *InterProcessor Communication* (*IPC*) protokol. *IPC* je jednostavan, datagramski, binarni *peer-to-peer* protokol na transportnom nivou. On obezbeđuje pouzdano otkrivanje grešaka u prenosu, ali ne obavlja automatsku retransmisiju izgubljenih datagrama, a takođe ne omogućuje fragmentaciju / defragmentaciju datagrama. Neka osnovna obeležja protokola su:

- Svaki datagram poseduje „magični“ kod početka datagrama (0xAFFE) da bi se omogućila ponovna sinhronizacija nakon gubitka i povratka fizičke veze (npr. pri isključenju pa priključenju kabla);
- Svaki datagram zaštićen je CRC32 kontrolnom sumom kako bi se obezbedio integritet podataka;
- Model ponašanja protokola je model jednostranog potvrđivanja (*one-way-acknowledge*) sa rednim brojem datagrama;
- Uspostavljanje veze se obavlja rukovanjem, s obzirom na *peer-to-peer* prirodu protokola.

Veza se uspostavlja tako, što po uključivanju svaka strana šalje *connect* datagram drugoj strani, po jedan na svaki istek određenog vremenskog intervala (npr. 500 ms). Po prijemu potvrde za *connect* datagram, obe strane postavljaju *local\_connected* zastavicu. Po prijemu *connect* datagrama svaka strana šalje potvrdu (*ACK*) drugoj strani, te postavlja zastavicu *remote\_connected*. Po postavljanju obeju zastavica na svakoj od strana, zaključuje se da je veza uspostavljena i slanje datagrama sa podacima postaje omogućeno. Ukoliko sada pristigne novi *connect* zahtev, strana pretpostavlja da se druga strana restartovala i postupak povezivanja počinje iz početka.

Datagrami sa podacima se razmenjuju tako što svaka strana održava sopstveni brojač za dodeljivanje rednog broja svakom datagramu koji šalje. Slanje se smatra uspešnim kada pristigne *ACK* datagram od druge strane, u

kojem je postavljen redni broj poslatog datagrama. U slučaju greške u redosledu ili u prenosu (neispravan *CRC*, npr) i sl. strane se međusobno obaveštavaju *NACK* datagramima.

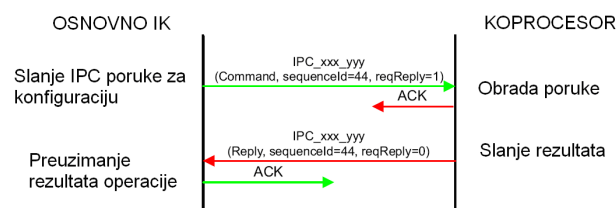
Jednostavna retransmisija u slučaju greške u prenosu ostvaruje se na nivou sesije. Postupak retransmisije je jednostavan: ukoliko slanje bude neuspešno, isto se ponavlja još 3 puta dok se ne odustane od slanja uz poruku o grešci.

Izrada protokola za *OS* koprocesora obavljena je u okviru *POSIX (Portable Operating System Interface)* uređaja, tako da se korišćenje protokola obavlja korišćenjem uređaja */dev/ipcX*. U okviru *Linux OS* obavljena je izrada *line-discipline* rukovaoca u okviru *TTY* sloja. Korišćenje *IPC* na *LinuxTV* obavlja se otvaranjem odgovarajućeg serijskog uređaja (npr. *ttyS1*), te postavljanjem *line-discipline* nad tim uređajem. Protokol se inicira ubacivanjem modula u jezgro operativnog sistema sa pozivom odgovarajuće funkcije, nakon završetka inicijalizacije jezgra.

### B. Protokol nivoa sesije

Protokol nivoa sesije oslanja se na primitive *IPC* protokola, omogućujući razmenu konkretnih upravljačkih poruka na relaciji osnovna platforma - koprocesor, kao i asinhronih statusnih poruka na relaciji koprocesor – osnovna platforma. Protokol omogućava slanje uz zahtev rezultata izvršenja operacije. Prikaz jednog scenarija dat je na Slici 5.

Format poruke sastoji se od koda uređaja kojem je poruka namenjena, koda same poruke (*play*, *stop*, *set format*, *set decoder feed* i sl), rednog broja poruke za omogućavanje povezivanja poruke zahteva sa porukom rezultata, zastavica koje označavaju da li je zahtevan rezultat operacije, odnosno da li sama poruka jeste poruka sa rezultatom operacije, polja sa veličinom cele poruke i, konačno, polja za smeštanje podataka koji se prenose.



Sl. 5. Jedan scenario komunikacije na nivou sesije

Problem se javlja pri pretvaranju podataka koji se prenose iz konvencije prisutne na osnovnoj platformi, u konvenciju koprocesora. Zbog toga, polje sa podacima se popunjava serijalizovanjem podataka pre slanja, dok se deserializacija vrši na odredištu. Svaka strana obavlja ove rutine u skladu sa tipovima svojih struktura i *API*-jem koji se koristi.

## VI. REZULTATI

Nakon izrade programske podrške, konačna TV arhitektura organizovana je tako što je osnovna platforma fizički povezana sa koprocesorskim integrisanim kolom. Integracija sa postojećim korisničkim aplikacijama

obavljena je za jako kratko vreme. Korisnički *use-case* scenario sastojao se iz sledećeg:

1. Inicira se jezgro *Linux* operativnog sistema;
2. Obavi se *boot* koprocesora;
3. Inicijalizuje se *IPC* i obavi se povezivanje rukovanjem;
4. Pokrene se korisnička aplikacija koja je svesna novih uređaja;
5. Prebacuje se trenutni kanal – sa *HD* u *H.264* na *MPEG2 SD* signal i obrnuto;
6. Meri se vreme prebacivanja i prati redosled komunikacije.

Posle dugotrajnog ispitivanja ponašanja sistema pri prebacivanju kanala, konstatovano je odsustvo grešaka. Vreme prebacivanja sa kanala na kanal bilo obavljeno za ~2 sekunde, što je prihvatljivo malo sa stanovišta korisnika TV uređaja.

## ZAHVALNICA

Zahvaljujemo se prof. dr Slavku Svirčeviću za uloženi trud pri pregledanju rada i pomoć pri prevodu tehničke terminologije na srpski jezik.

## LITERATURA

- [1] Milan Bjelica, "Jedno rešenje komunikacionog podsistema za proširenje funkcionalnosti TV uređaja," *MicronasNit*, interna dokumentacija 2008.
- [2] Milan Savić, Nikola Smiljković, Siniša Stanjlović "AVC-Accelerator-SW-Concept," *MicronasNIT*, interna dokumentacija 2007.
- [3] *VCTH Design Specification*, interna dokumentacija, verzija 1.2, 2007.
- [4] *MDXH Design Specification*, interna dokumentacija, 2007.
- [5] Michel Dupagne, Peter Benjamin Seel, "High-Definition television: A Global Perspective," Iowa state press, 1997.
- [6] Iain Richardson, Iain E.G. Richardson, "H.264 and MPEG-4 Video Compression: Video Coding for next generation multimedia," *Wiley*, 2003.
- [7] Ronald De Bruin, Jan Smits, "Digital video broadcasting – Technology, Standards and Regulations," *Artech House Publishers*, 1999.
- [8] Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman, "Linux device drivers – Third edition," *O'Reilly*, 2005.
- [9] *Linux DVB API Version 4*, [www.linuxdvb.org](http://www.linuxdvb.org), April 2005.

## ABSTRACT

This paper presents a conceptual solution as well as some implementation details, when extending standard digital TV architecture to support high-definition television. Extension is related to a need for additional hardware capable to decode *H.264* video and *AC3* audio streams. New chip equipped with all necessary routines is connected to host by serial interface, to maintain communication of command and control. Software extension is transparent to *Linux* user-space application that reside on host, by introducing new chip's devices as file system additional device abstractions.

## ONE SOLUTION FOR EXTENDING TV ARCHITECTURE TO SUPPORT HIGH DEFINITION TELEVISION

Milan Bjelica, Milan Savić,  
Tatjana Aleksić, Tomislav Maruna