

## Uvod

Vizualni tipovi podataka, kao što su slike, video i grafika su sve prisutni u današnjem digitalnom svijetu.

Sve je više podataka, sve je veća potražnja za pristupom, prijenosom i pohranom takvih podataka.

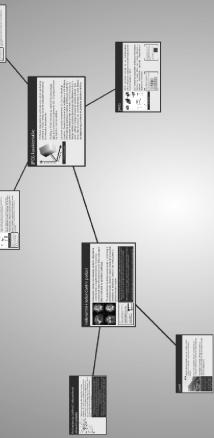
Tehnologija kompresije slike je glavni uvjet za ostvarivanje ovih zahtjeva, koji se očituje u mnogim različitim aplikacijama.

Na primjer, digitalna fotografija ne bi mogla biti komprimirana bez kompresije slike jer nujno je potrebno korištenje na mobilnim telefonima danas imaju presegno 10 megapiksela CCD (što generira 30 MB sirovih RGB podataka po snimku). Ako se za kompresiju koristi JPEG standard, tada će biti do 10-15 puta manje podataka (od 30 MB na originalu na 1-2 MB na komprimiranu sliku).

## 8. Kompresija slike

Disc. Martin Žagar

Multimedijiske arhitekture i sustavi



## 8. Kompresija slike

Disc. Martin Žagar

Multimedijiske arhitekture i sustavi

Na primjer, digitalna fotografija ne bi bila moguća bez kompresije slike jer i najobičnije kamere na mobilnim telefonima danas imaju prosječno 10 megapikselski CCD (što generira 30 MB sirovih RGB podataka) po slici. Većina kamera za kompresiju slike koristi JPEG standard, gdje se prosječna slika od 10 megapiksela sažima na 1 MB - 2MB.

### Redundancija (zalihost) i relevantnost slikovnih podataka

Tehnike kompresije slike mogu biti i sa i bez gubitaka, ali najčešće su kompresije slike hibridne, kombinirajući učinkovito korištenje algoritama sa i bez gubitaka za sažimanje slikovnih podataka. Takve tehnike se temelje na analizi slikovnih podataka u skladu sa dva važna aspekta:

- Uklanjanje irrelevantnih podataka – prvenstveno vizualno irrelevantnih (na primjer kad je rezolucija slike veća od rezolucije uređaja za prikazivanje), ali i aplikacijsko specifičnih irrelevantnosti (na primjer u medicinskim aplikacijama kada je određeno područje slike potpuno nebitno za daljnju obradu).
- Uklanjanje zalihosti – slikovni podaci (kao i svaki drugi podaci) imaju statističku zalihost, jer vrijednosti piksela nisu slučajne, nego visoko povezane, bilo u lokalnim područjima i bilo na razini cijele slike.



Redundancija (zalihost) i relevantnost slikovnih podataka

Tehnike kompresije slike mogu biti i sa i bez gubitaka, ali najčešće su kompresije slike hibridne, kombinirajući učinkovito korištenje algoritama sa i bez gubitaka za sažimanje slikovnih podataka. Takve tehnike se temelje na analizi slikovnih podataka u skladu sa dva važna aspekta:

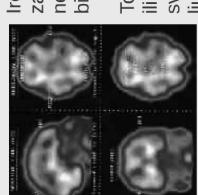
• Uklanjanje irrelevantnih podataka – prvenstveno vizualno relevantnih (na primjer kad je rezolucija slike veća od rezolucije uređaja za prikazivanje), ali i aplikacijsko specifičnih relevantnosti (na primjer u medicinskim aplikacijama kada je određeno područje slike potpuno nebitno za daljnju obradu).

• Uklanjanje zalihosti – slikovni podaci (kao i svaki drugi podaci) imaju statističku zalihost, jer vrijednosti piksela nisu slučajne, nego visoko povezane, bilo u lokalnim područjima i bilo na razini cijele slike.

## Irelevantni i redundanti podaci

Irelevantni podaci se uklanjuju pomoću algoritama za kompresiju sa gubicima budući da postoji nedostatak relevantnosti (važnosti) bilo u percepciji bilo u važnosti za potrebne aplikacije.

To se postiže primjenom kvantizacije u prostornoj i/ili frekvenčkoj domeni. Takve distorzije moraju biti svedene na minimum, kako bi bile neopazežene za ljudski vizualni sustav.



Zajednička karakteristika većine slika je da su susjedni pikseli u korelaciji koja nastaje zbog toga što slika nije slučajna zbirka piksela, nego koherentna struktura koja se sastoji od objekata. Osim toga, ako se vrijednost piksela promjeni, promjena je postupna.

Ukupno se svih piksela u svim regijama zajedno ne bi mogli prikazati u korelaciji, ali kada se razmatra manje područje, manje je odstupanje između piksela.

Ova lokalna sličnost odnosno redundancija se analizira u tehnikama kompresije slike.

To se postiže primjenom kvantizacije u prostornoj i/ ili frekvenčkoj domeni. Takve distorzije moraju biti svedene na minimum, kako bi bile neopazežene za ljudski vizualni sustav.

Zajednička karakteristika većine slika je da su susjedni pikseli u korelaciji koja nastaje zbog toga što slika nije slučajna zbirka piksela, nego koherentna struktura koja se sastoji od objekata. Osim toga, ako se vrijednost piksela promjeni, promjena je postupna.

Ukupno se svih piksela u svim regijama zajedno ne bi mogli prikazati u korelaciji, ali kada se razmatra manje područje, manje je odstupanje između piksela.

Ova lokalna sličnost odnosno redundancija se analizira u tehnikama kompresije slike.

Zajednička karakteristika većine slika je da su susjedni pikseli u korelaciji koja nastaje zbog toga što slika nije slučajna zbirka piksela, nego koherentna struktura koja se sastoji od objekata. Osim toga, ako se vrijednost piksela promjeni, promjena je postupna.

Ukupno se svih piksela u svim regijama zajedno ne bi mogli prikazati u korelaciji, ali kada se razmatra manje područje, manje je odstupanje između piksela.

Ova lokalna sličnost odnosno redundancija se analizira u tehnikama kompresije slike.

Redundancija može biti klasificirana na sljedeći način:

- **prostorna redundancija** – vrijednosti piksela u manjim područjima su vrlo slične
- **spektralna redundancija** – kad se podaci preslikavaju u frekvenčku domenu, nekoliko frekvenčija dominira nad drugima.

	24 ips	01	02	03	04
05	05	06	07	08	09
10	10	11	12	13	14
15	15	16	17	18	19
20	20	21	22	23	24

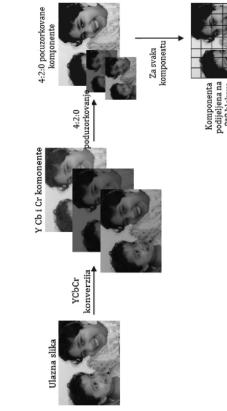
Kod videa postoji još jedna vrsta redundancija:

**vremenska redundancija**

Kod videa postoji još jedna vrsta redundancija:

**vremenska redundancija**

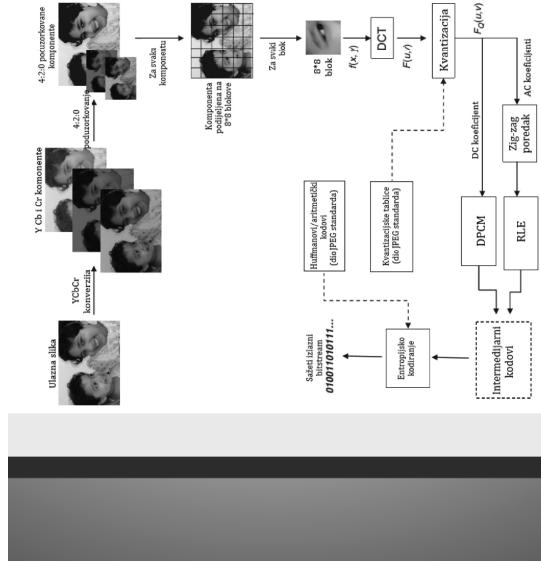
## JPEG tehnici Cosir



JPEG kompresija počinje transformacijom u prostor boja YCbCr komponentama.

Prostor boja YCbCr je vrlo sličan YUV prostoru boja koji smo prije objasnili.

## DCT dome prilik direkt



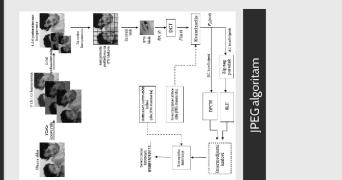
JPEG standard temelji se na transformacijskoj tehnički Diskretnoj kosinusnoj transformaciji (Discrete Cosine Transform - DCT).

DCT je izabran zbog dobre distribucije frekvenčke domene za realne slike, kao i zbog efikasnosti u prilikom aritmetičkih operacija direktno na hardveru.

JPG kompresija zasnovana je na transformaciji prostora boja YCbCr u YUV prostoru boja kroz operacije kodiranja i dekodiranja.

U nastavku će biti detaljnije opisane pojedinosti o tome kako se transformacija DCT koristi u procesu kompresije slike.

## JPEG



Svaka od  $Y$ ,  $Cb$  i  $Cr$  komponenta se računa neovisno o razbijanjem u  $8 \times 8$  blokove. U osnovnom načinu rada, blokovi su skenirani u linije s lijeva na desno i podzgora prema dolje i nad svakim se provodi DCT transformacija, kvantizacija koeficijenata i entropijsko kodiranje.

Svaka od  $Y$ ,  $Cb$  i  $Cr$  komponenta se računa neovisno o razbijanjem u  $8 \times 8$  blokove. U osnovnom načinu rada, blokovi su skenirani u linije s lijeva na desno i podzgora prema dolje i nad svakim se provodi DCT transformacija, kvantizacija koeficijenata i entropijsko kodiranje.

Istaknute karakteristike svakog koraka prikazane su u slijedećem popisu:



Istaknute karakteristike svakog koraka prikazane su u sljedećem popisu:

staknute karakteristi-  
u sljedećem popisu:

## JPEG karakteristike



(1) Svaka slika nožno se uzeti kao učez, ali se uviđek pro prevara u YCbCr format da bi se razvojio luminantna i krominantne komponente.

(2) Slika u YCbCr formatu se zatim 4:2:0 poduzorkuje, gde su krominatne komponente Cb i Cr poduzorkovane na četvrtinu izvorne veličine.

(3) Zatim se svaka komponenta (Y, Cb, Cr) obraduje posebno. Svaka komponenta je podijeljena u 8\*8 blokove. (8\*8 veličina je optimalna veličina prizora za prostorne i spektralne korelacije koje se koriste u DCT kvantizaciji). Manje veličine prozora povećavaju broj blokova u slici, a veće veličine snimanju korelacija između piksela. Ako širina slike (ili visina) nije višekratnik broja 8, rubni blokovi se popunjavaju s nulama da postignu potreblju veličinu.

## JPEG karakteristike

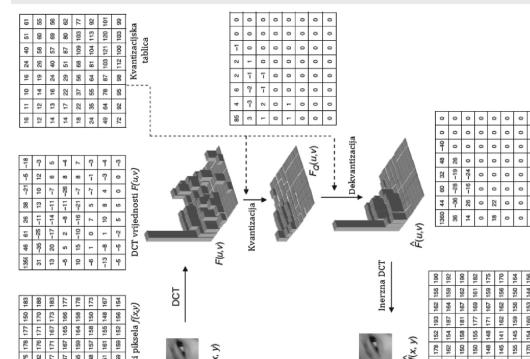
(1) Svaka slika nožne se užeti kao ulaz, ali se uviđek pro prevarava u YCbCr format da bi se razvojio luminantna i krominantne komponente.

(2) Slika u YCbCr formatu se zatim 4:2:0 poduzorkuje, gde su krominantne komponente Cb i Cr poduzorkovane u četvrtinu, izvorne veličine.

(3) Zatim se svakom komponentom ( $Y$ ,  $Cb$ ,  $Cr$ ) obrađuje posebno. Svaka komponenta je podijeljena u 8\*8 blokove. (8\*8 veličina je optimalna veličina prizora za prostore i spektralne korelacije koje se koriste u DCT kvarantaciji). Manje veličine prizora povećavaju broj blokova u slici, a širina slike (ili visina) nije višekratnik izroba 8. rubni blokovi se popunjavaju s nulama da postignu potreblju veličinu.

(4) Sv:  
transf:  
koefici:

Slika f  
prikaza  
DCT tr  
brojev  
naiibliž



Slika: DCT izračun 8\*8 bloka

Istaknut se DCT koeficijent  $F(u, v)$  kvantiziraju pomoću kvantizacijske tablice definirane JPEG standardom. Ova tablica je prikazana u gornjem desnom katu tablice. Svaki broj na položaju  $(u, v)$  daje kvantizacijski interval za odgovarajuću vrednost.

Dobrobit i kvantizacijskoj tablici se temelje na eksperimentalnim procjenama koju su pokazale da su niske frekvencije dominantne u slikama, a i ljudski uminski sustav je više osjetljiv na gubitak informacija u području niskih frekvencija. Sukladno tome, brojevi u nisko-frekvenčnom području (gornji lijevi dijelovi tablice) su manji i povećavaju se kako se krećemo prema visokofrekvenčnim koeficijentima u ostala tri kuta. Koristeći ovu tablicu možemo

vantizacijskoj tablici. Izračunate

3D crt  
odnosc

PEG-karakteristike

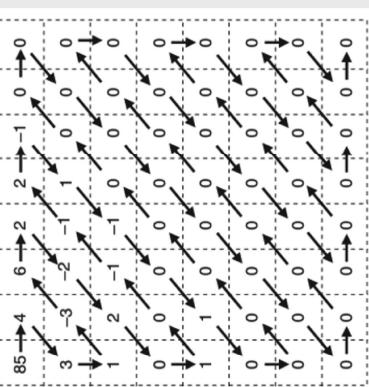
Nakon kvantitacije, gotovo sve visoke frekvencije  $F_Q(u, v)$  su nula, dok je nekoliko nisko-frekvencijskih vrijednosti ostalo. Za potrebe vrednovanja, također je prikazan proces dekodiranja koji  $F_Q(u, v)$  i deviantizira vrijednosti za izračun  $F'(u, v)$ .

Gubitak podataka u koeficijentima frekvencija je prilično očit, kada se usporedi  $F(u, v)$  i  $F'(u, v)$ .

(6) Kvantiranju koeficijenata  $F_Q(u, v)$  se zatim kodiraju koristećim određenog uzorka DC na poziciji  $F_Q(0, 0)$  odgovara najvišoj energiji i se obrađuju drugačije i usporenije s drugim AC koeficijentima u višim

DC koeficijenti blokova su kodirani u bomoću diferencijalne pulsokodne modulacije (PCM). AC koeficijenti se nov skeniraju u zig-zag redoslijedu, da bi se nu-uvredljivo poravnalo smanjio kružno vrijeme i smanjio dubinu visoke frekvencije u grupiranju prema kraju. To stvara manju entropiju skeniranih AC koeficijenata, koji se kodaju RLE algoritmom.

Nakon kvantizacije dok je nekoliko vrednovanja, tada se dekvantizira vr



i se obrađuje u frekvencija.

DC koeficijenti

Gubitak podatč  
usporede F (u,

(6) Kvantitiranje određenog učinkova i se obrađuje u frekvenciju.

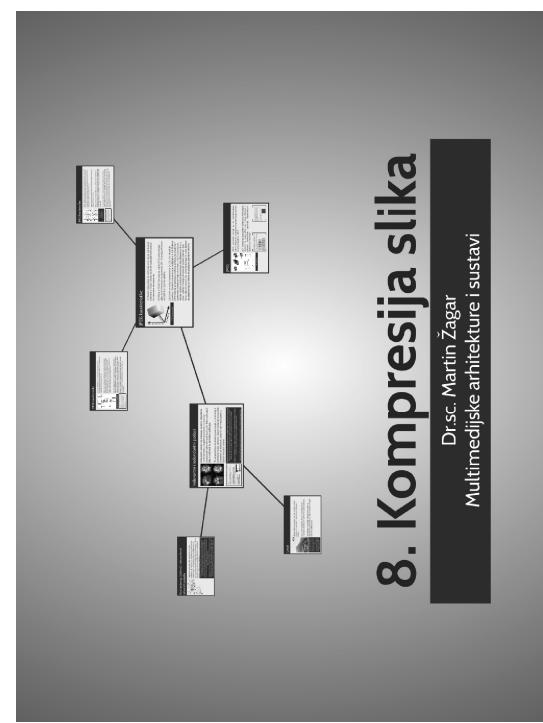
(1) Slijedeći, koeficijenti se učestalom povećavaju DC-ima koeficijentu.  $f_{DC}$  je enopriješnji koeficijent, ovi su koeficijenti pretvaraju u intermedijarne koeficijente. Za prikaz DC koeficijenta, to je razlika između DPCM vrijednosti tekućeg i prethodnog bloka. Razlika je prikazana kao par koji pokazuje veličinu u bitovima koji se koriste za kodiranje DPCM razlike i amplitudu DPCM razlike.

U našem primjeru, kvantizirana vrijednost DC koeficijenta je 85. Uz pretpostavku da vrijednost DC koeficijenta prethodnog bloka bila 82, DPC razlika je 3, koji treba

Za AC koeficijente, intermedijni kod se koristi samo za vrijednosti različite od nule. Svaki AC koeficijent različit od nule je predstavljen parom simbola – kombinacijom duljine niza i veličine, te amplitude AC koeficijenta. Prvi simbol le kombinirana vrijednost duljine niza uzastopnih nula koji nastupaju žig poretkom i broja bitova koji se koriste za kodiranje amplitude AC koeficijenta. Drugi simbol kodira amplitudu prvog AC koeficijenta različitog od nule.

U našem slučaju, prvi AC koeficijent različit od nule ima vrijednost 4, a drugi je jednak nuli. To znači da je u prethodnoj mreži nema nula koju mu prethode, tako da je duljina niza 0. Amplituda iznosa 4 treba za kodiranje 3 bita. Dakle, intermedijarni kod je  $<0,3>$   $<4>$ . Na slici su prikazani svi intermedijarni kodovi za sve AC koeficijente različiti od nule.

... 0 0 0 0 0 0 0 0



## 8. Kompresija slike

Dr.sc. Martin Žagar  
Multimodálna arhitektúra i súčasnosti

1010  
OB  
Biljniarni niz:

## Multimedijiske arhitekture i sustavi (dio 2)

Prof.dr.sc. Mario Kovač  
Prof.dr.sc. Hrvoje Mlinarić  
Doc.dr.sc.Josip Knežović

## Informacija

- Osnovna ideja: pronaći i ukloniti redundanciju u podacima
- Svojstvena (vlastita informacija)
  - Kvantitativna mjera za količinu informacije

$$i(A) = \log_b (1/P(A)) = -\log_b P(A)$$

gdje je:  $P(A)$  vjerovatnost pojavljivanja događaja A

- primjer:
  - za  $P(A)=1$   $i(A)=0$
  - za  $P(A)=0$   $i(A)=\infty$

Multimedijiske arhitekture i sustavi

## Informacija

- $i(A)$  nam formalno definira ono što je logično:
  - ako se nešto događa često (sa velikom vjerovatnošću) tada nam takav događaj ne donosi puno informacija (npr. suma brojeva na dvije kockice je 7)
  - rijetki događaji donose puno informacija (npr. suma brojeva na dvije kockice je 2)
  - ako izaberemo  $\log_2$  tada je količina informacija izražena u bitovima

Multimedijiske arhitekture i sustavi

2

3

## Entropija

- $S = \{a_1, a_2, a_3, \dots, a_N\}$  – skup svih mogućih simbola sa vjerovatnostima  $P(a_1), P(a_2), \dots, P(a_N)$
- $H(S) = \sum P(a_n) i(a_n) = -\sum P(a_n) \log_b P(a_n)$
- H predstavlja prosječni vlastiti sadržaj informacije za izvor S i naziva se Entropija.
- Gornji izraz predstavlja pojednostavljenje stvarnog izraza za entropiju i naziva se entropija nultog reda predavanja

Multimedijiske arhitekture i sustavi

4

## Entropija

- Ako izaberemo  $\log_2$  tada vrijednost entropije definira najmanji prosječan broj bitova potreban za kodiranje pojedinog simbola iz ulaznog niza
- Stvarnu entropiju slučajnog izvora za općeniti slučaj nikada ne možemo doznati

Multimedijiske arhitekture i sustavi

## MAS – Osnove kompresije

- Kompresija podataka:
  - Entropijski koder
  - Kodovi
  - Podaci
- Entropijski koder (statističko kodiranje): uklanjanje statističku redundanciju iz toka podataka

Multimedijiske arhitekture i sustavi

6



## Modeli

### Osnovni modeli

- Da li možemo na neki način još više 'smanjiti' vrijednost entropije? (entropija je za izabrani izvor uvek ista ali mi pokušavamo smanjiti našu procjenu entropije i time smanjiti kolicinu podataka)
- Ako na neki način možemo predvidjeti ponašanje izvora (tada govorimo o **modelu** tog izvora) tada možemo bolje predvidjeti entropiju (i u idealnom slučaju smanjiti procjenu)

Multimedijске arhitekture i sustavi

13

### Fizikalni model

- ako poznajemo fizikalna svojstva izvora
- obično je ovo prekompleksan model te se u takvim slučajevima pokušava koristiti neki alternativni model

### Vjerovatnosni model

- često korišten model, osnova za neke vrlo efikasne modelle
- pretpostavka je da su svi simboli generirani iz izvora potpuno neovisni
- Dvije osnovne varijante
  - 'slijepi' model: dodatno pretpostavljamo da je za sve simbole vjerovatnost pojavljivanja ista
  - statistički model: na neki način izračunati učestalost pojavljivanja simbola te na temelju toga definirati vjerovatnost
- Problem: mogućnost POVEĆANJA ENTROPIJE!!

Multimedijске arhitekture i sustavi

15

### Osnovni modeli

- Vjerovatnosni model
  - često korišten model, osnova za neke vrlo efikasne modelle
  - pretpostavka je da su svi simboli generirani iz izvora potpuno neovisni
  - Dvije osnovne varijante
    - 'slijepi' model: dodatno pretpostavljamo da je za sve simbole vjerovatnost pojavljivanja ista
    - statistički model: na neki način izračunati učestalost pojavljivanja simbola te na temelju toga definirati vjerovatnost
  - Problem: mogućnost POVEĆANJA ENTROPIJE!!

Multimedijске arhitekture i sustavi

14

### Osnovni modeli

- Markovijev model (A.A. Markov 1856-1922)
  - ovim modelom opisuje se izvor 's pamćenjem' tj izvor kod kojeg vjerovatnost pojavljivanja određenog simbola ovisi o svim simbolima koji su se pojavili prethodno u nizu
  - tako za Markovijev model prvi red vrijedi
$$P(A_i | A_{i-1}, A_{i-2}, A_{i-3}, \dots) = P(A_i | A_{i-1})$$
- ovaj model je izuzetno efikasan za određene tipove podataka (npr. model "predviđanja" spada u ovaj model)
- ponovo postoji opasnost od povećanja entropije!!

Multimedijске arhitekture i sustavi

16

### Osnovni modeli

- vrlo često samo jedan model ne može iskoristiti sve činjenice koje znamo o izvoriu te se stoga koriste višestrukti modeli kojima se značajno poboljšava opis izvora
  - Tijekom ovih predavanja vidjeti ćemo konkretnе primjere za većinu prije navedenih modela

### MAS – neki modeli

- Primjeri modela:
  - Slikovni podaci – prostorna korelacija u 2D prostoru, HVS
  - Video podatci – prostorna i vremenska korelacija podataka, HVS
  - Financijski podatci – korelacija u 1D, predviđanje

Multimedijске arhitekture i sustavi

17

Multimedijске arhitekture i sustavi

18

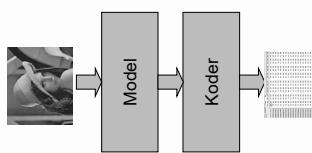
## MAS

Algoritam za kompresiju slike

### Model za slikovne podatke

- Podaci bi se mogli obraditi (kompresirati) i bez upotrebe posebno odabranog modela
- Npr. ZIP, RAR, ...
- Iz iskustva znamo da time nećemo ostvariti značajne omjere kompresije
- Upotrebljava se model koji omogućuje znatno veće omjere kompresije

### Arhitektura sustava za kompresiju slike/videa

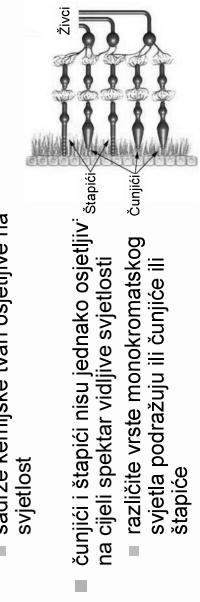


20

Multimedijeke arhitekture i sistavi

### Ljudski vizualni sustav

HVS model



### Model zasnovan na našem vizualnom sustavu

- fotoreceptori mrežnice
  - štapići (rods) i čunjići (cones)
  - sadže kemijske tvari osjetljive na svjetlost
- čunjići i štapići nisu jednako osjetljivi na cijeli spektar vidljive svjetlosti
  - različite vrste monokromatskog svjetla podražuju ili čunjiče ili štapiće
- čunjići doprinose osjetu i razlikovanju boja, a postaju aktivni pri višim razinama luminancije
- kod razina luminancije između 1cd/m<sup>2</sup> i 100cd/m<sup>2</sup> aktivni su i štapići i čunjići (fotopski vid)
- pri razinama luminancije većim od 100 cd/m<sup>2</sup> štapići postaju zasićeni i aktivni su samo čunjići

23

Multimedijeke arhitekture i sistavi

### HVS – spektralna osjetljivost

- štapići
  - osjetljivi na svjetlo i pri niskim razinama luminancije ispod 1cd/m<sup>2</sup> ("noćno" gledanje ili skotopski vid)
  - mogu razlikovati samo promjene u luminanciji, a nisu osjetljivi na boju
- čunjići
  - doprinose osjetu i razlikovanju boja, a postaju aktivni pri višim razinama luminancije
  - kod razina luminancije između 1cd/m<sup>2</sup> i 100cd/m<sup>2</sup> aktivni su i štapići i čunjići (fotopski vid)
  - pri razinama luminancije većim od 100 cd/m<sup>2</sup> štapići postaju zasićeni i aktivni su samo čunjići

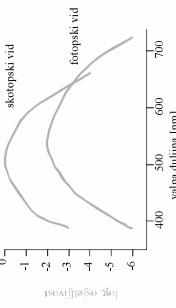
24

Multimedijeke arhitekture i sistavi

21

Multimedijeke arhitekture i sistavi

## HVS – spektralna osjetljivost



Ljudsko oko ima cca 10-20 puta više štapića nego čunjica

ZAKLJUČAK #1:

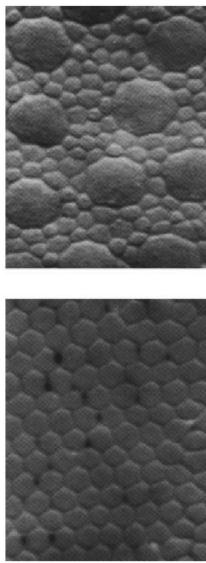
Znatno smo osjetljiviji na promjene u luminanciji nego u boji

Multimedijeke arhitekture i sustavi

25

## HVS – prostorna osjetljivost

- Gustoća receptora



(a) Fovea

(b) Periphery

Multimedijeke arhitekture i sustavi

26

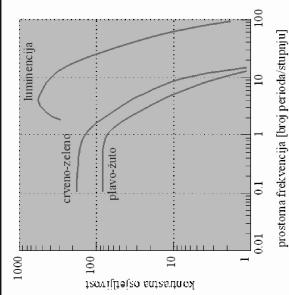
## HVS – prostorna osjetljivost

- Zbog konačne fizičke gustoće receptora, oko može razaznati detalje ako su zrake svjetlosti pod kutem upada većim od cca 1 minute
- Prostorna frekvencija



27

## HVS – prostorna osjetljivost



ZAKLJUČAK #2:

Oko je neosjetljivo na visoke prostorne frekvencije , a posebno to vrijedi za komponente boje.

Multimedijeke arhitekture i sustavi

28

## HVS Model za slike

- Kako iskoristiti prethodna saznanja o ljudskom vizualnom sustavu da bi omogućili veće omjere kompresije

Cilj bi nam bio:

- Zadržati što više informacija o luminantnoj komponenti
- Izbaciti prostorne frekvencije koje oko ne vidi

Multimedijeke arhitekture i sustavi

29

## Korištenje karakteristika HVS

- Kako? Moramo transformirati podatke
  - 1. Prebaciti u prostor boja što sličnije oku
  - 2. Prebaciti u frekvencijsku domenu

- 1. Promjena prostora boja
- 2. Transformacija
- Fizički model !!!

Multimedijeke arhitekture i sustavi

30

## Promjena prostora boja

- Ulazni podaci: uglavnom RGB
- Prikidan prostor boja: YUV
- RGB-YUV konverzija vrlo se lako može obaviti jednostavnom matričnom operacijom:

$$\begin{aligned}Y &= (0.257^*R) + (0.504^*G) + (0.098^*B) + 16 \\U &= -(0.148^*R) - (0.291^*G) + (0.439^*B) + 128 \\V &= (0.439^*R) - (0.368^*G) - (0.071^*B) + 128\end{aligned}$$

- Kompleksnost (za kasnije analize): za izračun svakog piksela treba 9 množenja i 9 zbrajanja (+dohvat, spremanje)

RGB  $\leftrightarrow$  YUV

32

Multimedijeke arhitekture i sustavi

## Promjena prostora boja

- Postoji i nešto drugačija formula za konverziju koja se koristi u JPEG-u a definirana je u JFIF dokumentu:

$$\begin{aligned}Y &= 0.299 R + 0.587 G + 0.114 B \\Cb &= -0.1687 R - 0.3313 G + 0.5 B + 128 \\Cr &= 0.5 R - 0.4187 G - 0.0813 B + 128\end{aligned}$$

33

Multimedijeke arhitekture i sustavi

## Promjena prostora boja

- Postoji i nešto drugačija formula za konverziju koja se koristi u JPEG-u a definirana je u JFIF dokumentu:

33

Multimedijeke arhitekture i sustavi

## Što smo postigli konverzijom?

- Dobili smo dva skupa komponenata
  - Y: luminanca, oko je znatno osjetljivije na ovu komponentu
  - U,V: boja, oko manje osjetljivo
- Ideja je da prethodna dva skupa obrađujemo RAZLIČITO
- No ovime još uvijek nismo postigli absolutno nikakvu promjenu u količini podataka

Multimedijeke arhitekture i sustavi

34

## Poduzorkovanje

- "subsampling"
- Jedna od najjednostavnijih metoda kako nakon konverzije u YUV prostor boja možemo smanjiti količinu podataka je poduzorkovanje:
  - Oko nje toliko osjetljivo na prostornu frekvenciju komponenata boje te se one ne prenose za svaki pixel
    - 4:4:4, 4:2:2, ...

Multimedijeke arhitekture i sustavi

35

## Transformacija

DCT

## Transformacija

### Transformacija: DCT

- Prebacivanjem podataka u frekvenčisku domenu želimo dobiti informacije o frekvenčnim karakteristikama svake komponente
- Koju transformaciju izabratи?
- Želja nam је да се nakon transformacije većina informacija zadriži u što manjem broju što nižih frekvenčkih elemenata
- Prema teoriji: Karhunen-Loeve (KLT) je ideana (ali je potpuno nepraktična za primjenu)

Multimedijiske arhitekture i sustavi

37

### DCT

- Iz teorije se može vidjeti da je diskretna kosinusna transformacija (DCT) vrlo bliska idealnoj
- Prednost DCT:
  - Može se jednostavnije izračunati (brzi algoritmi)

Multimedijiske arhitekture i sustavi

38

$$F(u,v) = \frac{2}{\sqrt{M \cdot N}} \cdot C(u) \cdot C(v) \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) \cdot \cos \left[ \frac{(2 \cdot i + 1) \cdot u \cdot \pi}{2 \cdot M} \right] \cdot \cos \left[ \frac{(2 \cdot j + 1) \cdot v \cdot \pi}{2 \cdot N} \right]$$

$F(u,y)$  - transformacijski koeficijent

$f(i,j)$  - amplitudne elemente slike u bloku

$u, v$  - koordinate u području transformacije (prostorne frekvencije)

$i, j$  - koordinate u području elemenata slike

$C(u)=C(v)=(1/2)^{1/2}$ , za  $u,v=0$

$C(u)=C(v)=1$ , za  $u=1,2,\dots,M-1, v=1,2,\dots,N-1$

Multimedijiske arhitekture i sustavi

39

### DCT za slike

- DCT kod obrade slike uglavnom se obavlja nad blokovima podataka veličine 8x8:

$$F(u,v) = \frac{1}{4} \cdot C(u)C(v) \cdot \sum_{i=0}^7 \sum_{j=0}^7 f(i,j) \cdot \cos \left[ \frac{(2 \cdot i + 1) \cdot u \cdot \pi}{16} \right] \cdot \cos \left[ \frac{(2 \cdot j + 1) \cdot v \cdot \pi}{16} \right]$$

Poduzorkovanje је vrlo "primitivna" i nekvalitetna metoda

DC koeficijent:

$$F(0,0) = \frac{1}{8} \cdot \sum_{i=0}^7 \sum_{j=0}^7 f(i,j)$$

(DC=8 x srednja vrijednost elemenata bloka)

Multimedijiske arhitekture i sustavi

40

### IDCT

- Na sličan način definirana је i inverzna DCT:

$$f(i,j) = \frac{1}{4} \cdot \sum_{v=0}^7 \sum_{u=0}^7 C(u)C(v) F(v,u) \cdot \cos \left[ \frac{(2 \cdot i + 1) \cdot u \cdot \pi}{16} \right] \cdot \cos \left[ \frac{(2 \cdot j + 1) \cdot v \cdot \pi}{16} \right]$$

DCT dovodi do proširenja opsega podataka (dinamički opseg za 8x8 2D DCT je  $2^3$  puta veći u odnosu na ulaz)

Ako na ulazu imamo 8 bitovne podatke nakon 2D DCT imati ћemo 11 bitovne koeficijente!

Još uvjek su SVL podaci sačuvani i moguće je obaviti perfektnu rekonstrukciju (uz uvažavanje nepreciznosti matematičkih izračuna)

### Opseg podataka

- DCT dovodi do proširenja opsega podataka (dinamički opseg za 8x8 2D DCT je  $2^3$  puta veći u odnosu na ulaz)
- Ako na ulazu imamo 8 bitovne podatke nakon 2D DCT imati ћemo 11 bitovne koeficijente!
- Još uvjek su SVL podaci sačuvani i moguće je obaviti perfektnu rekonstrukciju (uz uvažavanje nepreciznosti matematičkih izračuna)

Multimedijiske arhitekture i sustavi

41

Multimedijiske arhitekture i sustavi

42

## Pomak kod DCT

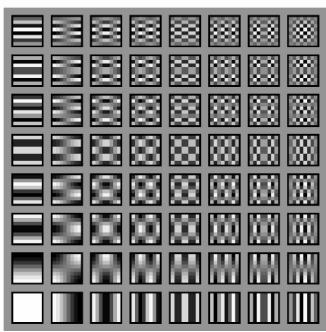
- Slikovni podaci na ulazu su pozitivni cijeli brojevi (npr. 0-255)
- S obzirom da je DCT definirana i za pozitivno i negativno područje na ovaj način bi se polovica ulaznog prostora izgubila ( a time i znatno smanjila efikasnost)
- Zato se prije DCT sve vrijednosti na ulazu translatiraju za polovicu opsega tj. -128
- Prema tome umjesto da ulazni elementi budu u opsegu [0-255] biti će u opsegu [-128 – 127]

Multimedijeke arhitekture i sustavi

43

## 2D-DCT bazne funkcije

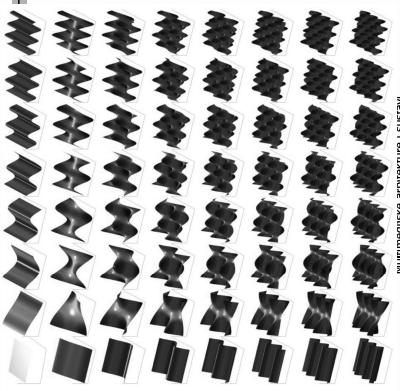
- bijelo pozitivne vrijednosti, crno negativne vrijednosti



Multimedijeke arhitekture i sustavi

45

## Prikaz u 3D



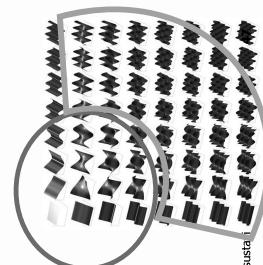
Multimedijeke arhitekture i sustavi

46

## Što smo dobili sa DCT?

- Energija signala koncentrirana u nižim frekvencijama

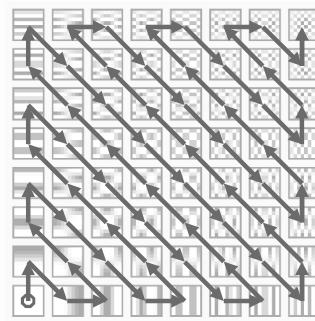
- Značajniji koeficijenti
- Manje značajni koef.



Multimedijeke arhitekture i sustavi

48

## Cik-cak (Zig-zag) reorganizacija



Multimedijeke arhitekture i sustavi

49

## Kako je to na stvarnoj slici....



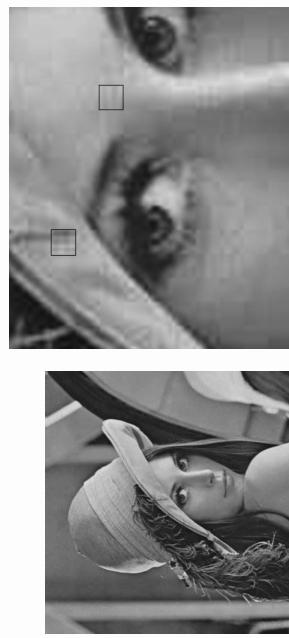
Multimedijeke arhitekture i sustavi

49

## Kako je to na stvarnoj slici...

### Primjer

- Primjer bloka



Multimediješke arhitekture i sustavi

50

## Kako iskoristiti DCT?

- Nakon DCT, uz poznavanje prosječnih osjetljivosti oka možemo smanjivati količinu podataka koja opisuje frekvencije na koje naše oko nije osjetljivo
  - To ne možemo učiniti potpunim odbacivanjem visokofrekventnih komponenata već smanjenjem njihove preciznosti (problemi naglih prijelaza !)

Multimediješke arhitekture i sustavi

52

## Kvantizacija nakon DCT

- kvantizacijska tablica sa 64 vrijednosti  $q(u,v)$  određenih na temelju HVS (koraci kvantizacije)
- Svaki DCT koeficijent  $F(u,v)$  dijeli se sa pripadnim (skaliranim) faktorom  $q(u,v)$ , a rezultat se zaokružuje na najbližu cijelobrojnu vrijednost
- Skaliranjem sa faktorom  $S$  se pojednostavljeno određuje stupanj kompresije i "kvalifeta" slike
- vrijednosti nastale kvantizacijom  $S(u,v)$  su:

$$S(u,v) = \text{round}\left(\frac{F(u,v)}{Q(u,v)}\right) = \text{round}\left(\frac{F(u,v)}{q(u,v)S}\right)$$

Multimediješke arhitekture i sustavi

54

## Što je kvantizacija

- Postupak kojim se smanjuje dinamički opseg ulaznih vrijednosti (a time i potreban broj bita)
- Ulazni podatak dijeli se sa zadanim brojem (kvantizacijski korak)
- Kod inverznog postupka podatak se množi sa kvantizacijskim korakom

### Primjer (Q=4):

- Ulazni niz 5,11,4,17,1 (potrebno 5 bitova za prikaz)
- Izlazni niz 1,3,1,4,0 (potrebno 3 bita za prikaz)
- Restaurirani niz 4,12,4,16,0 (GUBITAK PODATAKA)

**KVANTIZACIJOM DOLAZI DO NEPOVRATNOG GUBITKA INFORMACIJA !!!**

## Kvantizacija



Multimediješke arhitekture i sustavi

51

Multimediješke arhitekture i sustavi

55

## Kvantizacija nakon DCT

- Utjecaj pojedinih prostornih frekvencija može se proizvoljno kontrolirati postupkom kvantizacije
- S obzirom da smo razdvojili Y i U, V komponente sada možemo Y DCT frekvenčijske koeficijente kvantizirati sa različitim kvantizacijskim vrijednostima od U, V komponenti (kvant. koef. za U, V biti će veći zbog HVS)
- Također kvantizacijski koeficijenti za visoke frekvencije mogu biti veći

Multimedijeke arhitekture i sustavi

56

## Tipične kvantizacijske tablice

Table K.1 – Luminance quantization table															
16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	53	18	21	25	64	59	99	99	99
14	13	16	24	49	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	59	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	59	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	59	99	99	99	99	99	99	99
72	92	95	98	111	100	103	99	59	99	99	99	99	99	99	99

Table K.2 – Chrominance quantization table															
16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	53	18	21	25	64	59	99	99	99
14	13	16	24	49	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	59	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	59	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	59	99	99	99	99	99	99	99
72	92	95	98	111	100	103	99	59	99	99	99	99	99	99	99

Table K.2 – Chrominance quantization table															
16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	53	18	21	25	64	59	99	99	99
14	13	16	24	49	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	59	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	59	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	59	99	99	99	99	99	99	99
72	92	95	98	111	100	103	99	59	99	99	99	99	99	99	99

Table K.2 – Chrominance quantization table															
16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	53	18	21	25	64	59	99	99	99
14	13	16	24	49	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	59	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	59	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	59	99	99	99	99	99	99	99
72	92	95	98	111	100	103	99	59	99	99	99	99	99	99	99

Table K.2 – Chrominance quantization table															
16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	53	18	21	25	64	59	99	99	99
14	13	16	24	49	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	59	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	59	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	59	99	99	99	99	99	99	99
72	92	95	98	111	100	103	99	59	99	99	99	99	99	99	99

Table K.2 – Chrominance quantization table															
16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	53	18	21	25	64	59	99	99	99
14	13	16	24	49	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	59	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	59	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	59	99	99	99	99	99	99	99
72	92	95	98	111	100	103	99	59	99	99	99	99	99	99	99

Table K.2 – Chrominance quantization table															
16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	53	18	21	25	64	59	99	99	99
14	13	16	24	49	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	59	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	59	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	59	99	99	99	99	99	99	99
72	92	95	98	111	100	103	99	59	99	99	99	99	99	99	99

Table K.2 – Chrominance quantization table															
16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	53	18	21	25	64	59	99	99	99
14	13	16	24	49	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	59	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	59	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	59	99	99	99	99	99	99	99
72	92	95	98	111	100	103	99	59	99	99	99	99	99	99	99


## Što nakon HVS modela

- Statističkom analizom podataka nakon obrade u koderskom modelu može se vidjeti da je njihova raspodjela izvršna za daljnju kompresiju algoritmima zasnovanim na statističkim modelima
- Drugi dio JPEG kodera radi upravo to

Multimedijeke arhitekture i sustavi

62

## DZ1

- Prije nego pređemo na statistički koder definirajmo zadatak za DZ
- ... Nakon ovog predavanja mislim da DZ ne bi trebala biti problem...

Multimedijeke arhitekture i sustavi

63

- Natuknice: ne prepisujte, isprobajte, usporedite,...
- [MAS\\_DZ1.ppt](#)

Multimedijeke arhitekture i sustavi

63

## Statistički/entropijski koder

- Nakon ovog predavanja mislim da DZ ne bi trebala biti problem...

63

## Od čega krećemo

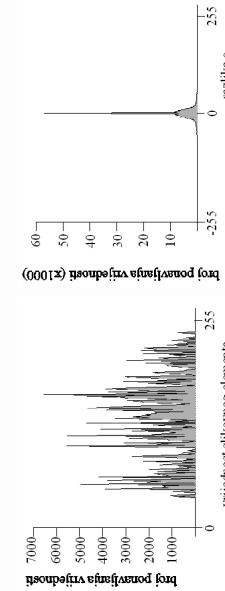
- Entropijski koder prima kvantizirane koeficijete presložene po cik-cak algoritmu u nizove od po 64 koeficijenta za svaku komponentu bloka
- Koje karakteristike možemo uočiti? (koristite DZ i proučite rezultantne vrijednosti za nekoliko slika)

Multimedijeke arhitekture i sustavi

65

## DC koeficijenti

- Podloga: analiza statistike DC komponenata



Multimedijeke arhitekture i sustavi

66

## DC koeficijenti

- Teorijska podloga: DC koeficijent predstavlja srednju vrijednost svih 64 elementa bloka
- dva susjedna bloka obično imaju vrlo slične srednje vrijednosti

### Rezultat:

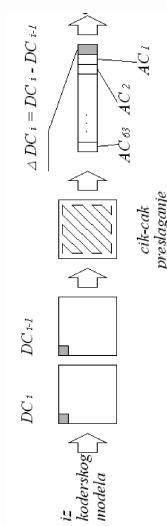
- Distribucija DC vrijednosti je prilično jednolika i nepogodna za kodiranje
- Distribucija RAZLIKE dva susjedna DC koeficijeta je vrlo gusta oko nule i idealna za kodiranje

Multimedijeke arhitekture i sustavi

67

## DC razlika

- Iz prethodnoga odlučujemo da ne želimo kodirati vrijednost DC elementa već razliku između trenutnog i prethodnog
- Znatna ušteda u bitovima (malo po malo...)



Multimedijeke arhitekture i sustavi

68

## Kodiranje duljine niza nula (ZRL)

- Slijedeći korak u našoj analizi proizlazi iz postupka kvantizacije nakon koje je mnogo elemenata viših frekvencija postalo nula.
- Za ovakve nizove izuzetno je pogodna metoda kodiranja duljine niza (ZRL coding)
- Metoda radi na način da umjesto da se kodira svaki koeficijent koji je nula u nizu, da se kodira broj uzastopnih nula koji se nalazi u nizu

Multimedijeke arhitekture i sustavi

69

## DC/AC simboli

- Dodatako se u statističkom modelu određuje pripadnost pojedinog ulaznog elementa određenoj kategoriji prema apsolutnoj vrijednosti njegove amplitudne
- Određivanje kategorije povezano je sa postupkom modificiranog Huffmanovog kodiranja o čemu će biti više riječi kasnije. Na izlazu iz statističkog modela svaka ulazna DC razlika i svaki ulazni AC koeficijent različiti od nule biti će zamjenjeni slijedećim simbolima:
  - DC simbol: [Kategorija] [amplituda]
  - AC simbol: [[duljina niza nula],[kategorija]] [amplituda]
  - DC simbol nema komponentu koja određuje duljinu niza nula što je i logično jer JPEG zasebno obradjuje svaki slikovni blok a DC razlika uveće je prvi element bloka.
  - Za AC komponente duljina niza nula opisuje koliko koeficijentata ima vrijednost nula prije nekog koeficijentata koji je različit od nule.

Multimedijeke arhitekture i sustavi

70

## ZRL/EOB

- Veličina polja koje opisuje duljinu niza je ograničena na 4 bita kojima se mogu opisati nizovi od 0 do 15 nula. U stravom nizu ulaznih elemenata može se pojaviti niz koji sadži i više od 15 nula te se u tom slučaju koristi specijalni simbol nazvan ZRL (Zero Run Length).
- ZRL simbol označava niz od 16 nula a nakon tog simbola nule se počinju brojati iz početka.
- Drugi, i zadnji, specijalni simbol označava da su od trenutnog elementa do kraja bloka svi elementi jednaki nuli. Ovaj simbol ima označku ECB (End Of Block).
- Razlog je prirođan. EOB simbol označava sve nule od trenutnog mjeseta do kraja bloka pa su prema tome ZRL simboli koji eventualno prethode EOB simboli nepotrebni.

Multimedijeke arhitekture i sustavi

71

## Kategorija

Kategorija	DC razlika	AC koeficijent
0	0	-1,1
1	-1,1	-3,-2,3
2	-3,-2,3	-3,-2,3
3	-7,-4,7	-7,-4,7
4	-15,-8,15	-15,-8,15
5	-31,-16,31	-31,-16,31
6	-63,-32,63	-63,-32,63
7	-127,-64,127	-127,-64,127
8	-255,-128,128,255	-255,-128,128,255
9	-511,-256,256,511	-511,-256,256,511
10	-1023,-512,512,1023	-1023,-512,512,1023
11	-2047,-1024,1024,2047	-2047,-1024,1024,2047

- Neke vrijednosti se rijetko pojavljuju pa nije imalo smisla svakom broju pridjeljivati kod
- Kategorije su određene prema vrijednosti pojavljivanja ulaznih elemenata
- Kako se kategorijom ne može točno odrediti vrijednost elementa poslije kategorije mora se poslati još podatak [amplituda] koji unutar kategorije definira koji je to element.
- Amplituda ima različit broj bitova za svaku kategoriju (Koji je jednak rednom broju kategorije)

Multimedijeke arhitekture i sustavi

72

## Kategorija

- Dodatako se u statističkom modelu određuje pripadnost pojedinog ulaznog elementa određenoj kategoriji prema apsolutnoj vrijednosti njegove amplitudne
- Određivanje kategorije povezano je sa postupkom modificiranog Huffmanovog kodiranja o čemu će biti više riječi kasnije. Na izlazu iz statističkog modela svaka ulazna DC razlika i svaki ulazni AC koeficijent različiti od nule biti će zamjenjeni slijedećim simbolima:
  - DC simbol: [Kategorija] [amplituda]
  - AC simbol: [[duljina niza nula],[kategorija]] [amplituda]
  - DC simbol nema komponentu koja određuje duljinu niza nula što je i logično jer JPEG zasebno obradjuje svaki slikovni blok a DC razlika uveće je prvi element bloka.
  - Za AC komponente duljina niza nula opisuje koliko koeficijentata ima vrijednost nula prije nekog koeficijentata koji je različit od nule.

Multimedijeke arhitekture i sustavi

73

## Tablica simbola za AC komponente

Dujina rječi	Kategorija	Dujina koda	Kodna riječ
0	EOB	0/1	0/10
1	x	1/1	0/10
...	x	:	...
14	x	14/1	14/10
15	25L	15/1	15/10

Multimedijeke arhitekture i sustavi

74

## Amplituda

- Za kodiranje amplitude koristi se sljedeće pravilo:
- Pretpostavimo da je koeficijent C zapisan u formatu dvojnog komplementa, a K je kategorija kojoj taj koeficijent pripada.
- Ako je C pozitivan broj tada će se Huffmanovom kodu, kao proširenje, dodati K nižih bitova od C.
- Ako je C negativan tada će se Huffmanovom kodu dodati K nižih bitova od vrijednosti koeficijenta C umanjenog za jedan, tj. (C-1).

Multimedijeke arhitekture i sustavi

75

## Huffman-ova tablica za DC simbole

Kategorija	Dujina koda	Kodna riječ
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Multimedijeke arhitekture i sustavi

76

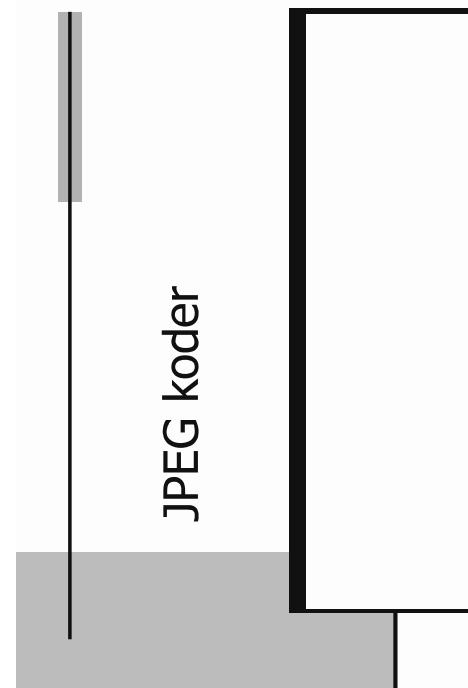
## Huffman-ova tablica za AC simbole

Dujina rječi/kategorija	Dujina koda	Kodna riječ
0/0 (EOB)	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	111000
0/7	8	1111000
0/8	10	11111010
0/9	16	11111110000010
0/10	16	111111110000011
1/1	4	1100
1/2	5	11011
...	...	...

Multimedijeke arhitekture i sustavi

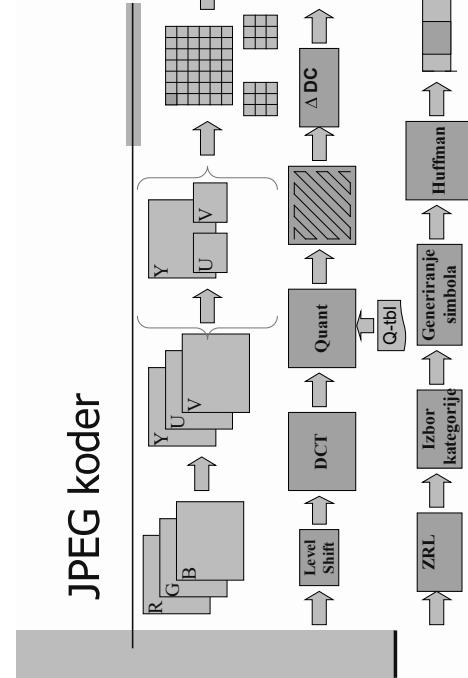
77

## JPEG koder



Multimedijeke arhitekture i sustavi

## JPEG koder



Multimedijeke arhitekture i sustavi

79

## JPEG grayscale

Potpuni primjer kodiranja jednog bloka

$$\begin{bmatrix} 107 & 113 & 122 & 122 & 122 & 120 & 137 & 139 \\ 113 & 122 & 122 & 122 & 122 & 120 & 133 & 139 \\ 122 & 122 & 122 & 122 & 122 & 120 & 133 & 146 \\ 107 & 113 & 122 & 122 & 122 & 120 & 133 & 146 \\ 113 & 122 & 122 & 122 & 122 & 120 & 133 & 146 \\ 122 & 122 & 122 & 122 & 122 & 120 & 133 & 146 \\ 107 & 113 & 122 & 122 & 122 & 120 & 133 & 146 \\ 113 & 122 & 122 & 122 & 122 & 120 & 133 & 147 \end{bmatrix} \xrightarrow{\text{Pomak}} \begin{bmatrix} -21 & -15 & -6 & -6 & 11 & 9 & 11 \\ -21 & -15 & -6 & -6 & 11 & 5 & 11 \\ -21 & -15 & -6 & -6 & 11 & 5 & 11 \\ -21 & -15 & -6 & -6 & 11 & 5 & 11 \\ -21 & -15 & -6 & -6 & 11 & 5 & 11 \\ -21 & -15 & -6 & -6 & 11 & 5 & 11 \\ -21 & -15 & -6 & -6 & 11 & 5 & 11 \\ -21 & -15 & -6 & -6 & 11 & 5 & 11 \end{bmatrix} \xrightarrow{\text{DCT}} \begin{bmatrix} -31 & -84 & 2 & -16 & -5 & -10 & 13 & -3 \\ 12 & -52 & 1 & 1 & -9 & 2 & 1 & -7 \\ -11 & 7 & -6 & 3 & -1 & 1 & 1 & 0 \\ 5 & -6 & -2 & 7 & -2 & 6 & -1 & -2 \\ -1 & -1 & -2 & 1 & -4 & 1 & 2 & -1 \\ -1 & 3 & -9 & 1 & -1 & 0 & 4 & -3 \\ -2 & -2 & 9 & -4 & 1 & -6 & -4 & -3 \\ 12 & 0 & -8 & 1 & 1 & 2 & -3 & 0 \end{bmatrix} \xrightarrow{\text{Cik-cak}} \begin{bmatrix} -2 & -8 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

## Pomak (level shift)

## 2D DCT, kvantizacija

## Cik-cak + entropijsko kodiranje

-2 -8 1 -10 0 -10 1 0 0 0 0 0 ... 0  
01101010101100100100100010111000101110110101010

## JPEG+JFIF

Kako je zapisana slika u računalu

## Kako se slika zapisuje i prenosi

- Niz binarnih podataka pohranjen ne može se interpretirati...
  - koja je dimenzija, boje/cb, kako je kvantizirano,...

- Upravo zato JPEG norma definira način zapisa podataka vezanih za kompresiju slike

Markeri

- Sve počinje definiranjem MARKERa  
Markeri su specijalni dvo-bajtni podaci koji  
počinju sa podatkom 0xFF kojeg slijedi  
podatak različit od 0 ili 0xFF a kojim se  
identificiraju različiti strukturalni dijelovi  
kompresiranog niza podataka

Minimalna struktura JPEG datoteke

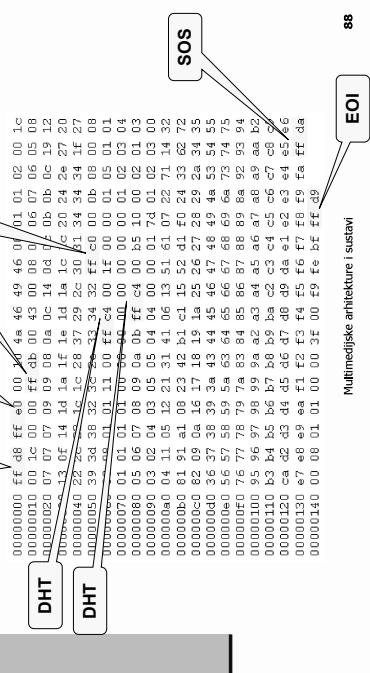
SOI	FFD8	Start of Image
DQT	FFDB	Quantization table(s)
DHT	FFC4	Huffman table(s)
SOF	FFC0	Frame header
SOS	FFDA	Scan header
Kompresirani podaci		
EOI	FFD9	End of Image

Multimedijiške arhitekture i sustavi

Multimedijiske arhitekture i sustavi

87

Primjer: (test1\_crno\_gray.jpg)



Multimedijiske arhitekture i sustavi  
FOI

Multimedijiske arhitekture i sustavi

87

33

## Uvod

U kompresiji slike, analiziramo jedan statičan okvir ispitujući prostornu i spektralnu redundanciju. Ta je redundancija postojila je u sliku koherentna struktura, a ne slučajna zbirka piksela.

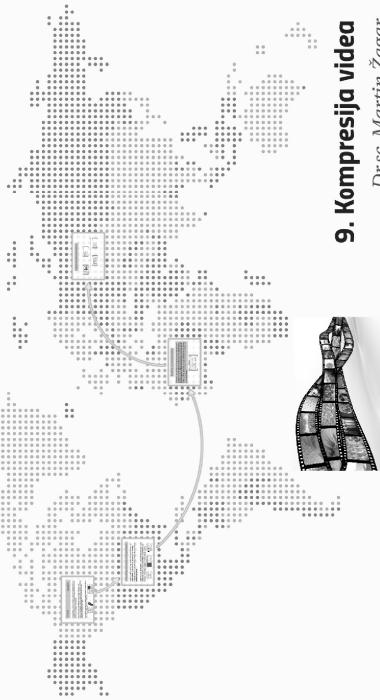
Budući da je video slijeđi slikeovnih okvira, od kojih svaki ima svoje prostone redundancije, možemo prepostaviti da je prilikom kompresije videa potrebno primijeniti neki algoritam za kompresiju slike (npr. JPEG) na svaki slikeovni okvir u video.

Ovaj algoritam je poznat kao Motion JPEG (M-JPEG). M-JPEG sažima svaki kadr u video sekvenci kao JPEG sliku.



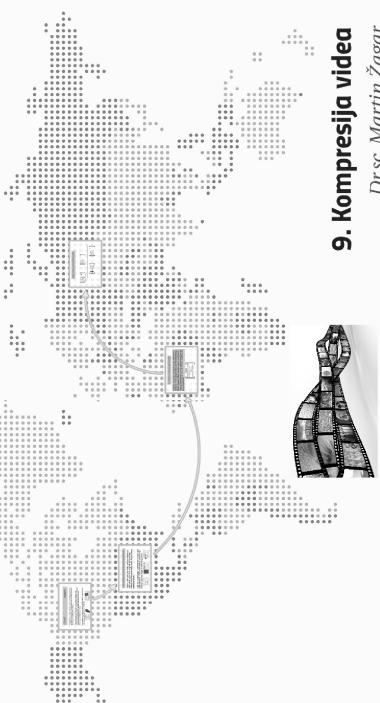
## 9. Kompresija video

Dr.sc. Martin Žagar  
Multimedijalne arhitekture i sustavi



## 9. Kompresija video

Dr.sc. Martin Žagar  
Multimedijalne arhitekture i sustavi

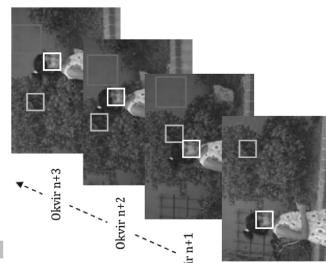


# U u video sekvenci

Značajno veći omjer kompresije može se postići uklanjanjem druge vrste redundancije u video - vremenske redundancije.

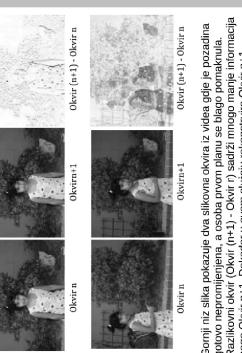
Baš kao što su vrijednosti piksela lokalno slične unutar jedne slike, one su takođe povezane i između slikeovnih okvira.

Na primjer, u video se većina piksela koji čine pozadinu ne razlikuje mnogo od okvira do okvira. Područja koja se mijenjaju su ili objekti koji se pomiču ili se pomiče kamera, ali ti pokreti su kontinuirani i stoga predviđljivi.

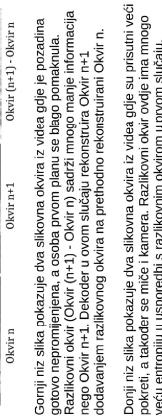


Postojanje vremenskih zavisnosti rezultira time da razlikovni okvir između dva slikeovna okvira sadrži manje podataka od samog slikeovnog okvira.

Zato je učinkovitije kodirati video kodiranjem razlikovnih okvira. Primjer razlikovnog okvira prikazan je na slici.



Gornji niz slika pokazuje dva slikeovna okvira iz video grupe je pozadina gotovo neprimjenjena, a osobna privremena planu se bilo pomaknula. Razlikovni okvir (Okvir (n+1) - Okvir n) sadrži manje informacija nego Okvir n+1. Dekoder u ovom slučaju rekonstruira Okvir n+1 dodavanjem razlikovnog okvira na prethodno rekonstruirani Okvir n. Donji niz slika pokazuje dva slikeovna okvira iz video grupe su prisutni veći pokreti, a takođe se miče i kamera. Razlikovni okvir ovdje ima mnogo veću entropiju u usporedbi s razlikovnim okvircima u prvom slučaju.



Gornji niz slika pokazuje dva slikeovna okvira iz video grupe je pozadina gotovo neprimjenjena, a osobna privremena planu se bilo pomaknula. Razlikovni okvir (Okvir (n+1) - Okvir n) sadrži manje informacija nego Okvir n+1. Dekoder u ovom slučaju rekonstruira Okvir n+1 dodavanjem razlikovnog okvira na prethodno rekonstruirani Okvir n. Donji niz slika pokazuje dva slikeovna okvira iz video grupe su prisutni veći pokreti, a takođe se miče i kamera. Razlikovni okvir ovdje ima mnogo veću entropiju u usporedbi s razlikovnim okvircima u prvom slučaju.

## Predviđanje na temelju kretanja blokova

Trenutni okvir slike, koji treba biti sažet naziva se **ciljni okvir**. Ciljni okvir se predviđa na temelju kretanja bloka iz prethodnog okvira, koji se naziva **referentni okvir**.

Ciljni okvir je podijeljen u blokove koji se zovu još i **makroblokovi** i svaki makroblok se predviđa na temelju najsličnije regije iz referentnog okvira.



Kad god postoji pokret zbog kretanja nekog objekta, vremenske redundancije mogu se bolje ukloniti predviđanjem kretanja objekta ili određenih blokova između okvira, a ne okvira u cijelini.

Predviđanje pokreta na temelju makroblokova služi iz dva razloga.

Prvo, vremenska zavisnost se lakše analizira na temelju blokova koji se kreću od okvira do okvira, nego cijelih okvira.



Druge, učinkovitije je podjeliti okvir na blokove i generirati vektor pomaka za svaki blok umjesto vektora pomaka za svaki piksel.

# UVJETI I VJEĆI

## Predviđanje podrazumijevanje vektora pokreta

za svak blok, kad se pokret može točno predviđati s vektorom poketa, odgovarajućeg makrobloka. U takvim slučajevima, razliku između stranog i predviđenog bloka je vrlo mala.

To ne mora uvjet biti slučaj zbog različitih razloga: npr. predviđanje podrazumijevanje vektora pokreta za makroblok mora biti točno za bolje predviđanje i kompresiju.

Međutim i podrazumijevanje vektora pomaka za makroblok mora biti posljedice imaju veću entropiju. Primjerice takvih makroblokova se mogu vidjeti na donjem desnom dijelu Okvira n<sup>1</sup>1. Usporedba na dnu slike pokazuje da je svezukupna razlika s predviđanjem pokreta mnogo manja nego bez predviđanja pokreta.

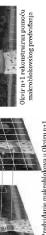
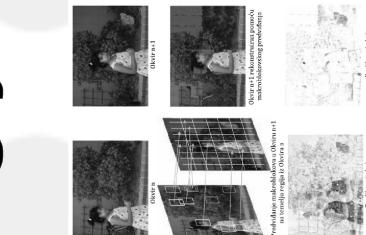
Proces predviđanja ciljnog okvira na temelju referentnog okvira pomoću makroblokova se naziva **kompenzacija pokreta**.

- Pronalaženje najboljeg vektora pomaka za svaki makroblok
- Stvaranje predviđenog okvira ili okvira nastao od kompenzacijom pokreta
- Računanje razlikovnog okvira.

- Kompresiju razlikovnog okvira pomoću JPEG algoritma (s gubicima) i vektora pomaka pomoću entropijskog kodiranja (bez gubitaka)

Pronalaženje najboljeg vektora pomaka za makroblok mora biti točno za bolje predviđanje i kompresiju.

Nakon što su vektori pomaka dostupni za svaki makroblok, ciljni okvir može se izračunati pomoću vektora pomaka i referentnog okvira.



## Izračun vektora pomaka

U prosjeku, 60% do 80% od ukupnog vremena kodiranja troši se samo na tražnje vektora pomaka. Područje pretraživanja je specifично za svaki algoritam pretraživanja. Cilj traženja vektora pomaka je pronaći naviše podudarno područje u referentnom okviru za ciljani makroblok. Da bi to mogli kvantificirati moramo uvesti mjeru poremećaja koja će biti minimalna za područje koje se najviše podudara.



## Mjere poremećaja

Ako definiramo gornji lijevi kut kao ishodište (0,0) makrobloka širine  $m$  i visine  $n$ ,  $C(n+1)$  ( $x, y$ ), gde i  $m$  je vrijednost  $[0, m]$  i  $y$  je vrijednost  $[0, n]$ , definira vrijednost za svaki piksel u makrobloku.

Pikseli u referentnom okviru koji se razmatraju u odnosu na makroblok iz ciljnog okvira (primijetite da referenti okvir nije podijeljen na blokove) mogu se definirati kao  $C_0(x, y)$ , gde  $x$  i  $y$  imaju raspon.

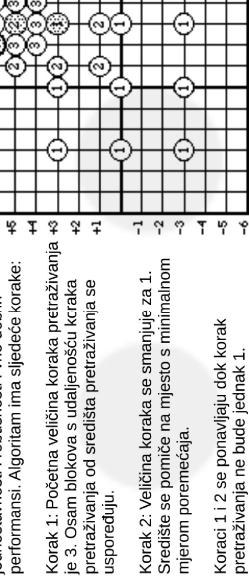
Za svu moguću takvu područja s pomačkom ( $i, j$ ) u odnosu na razmatrani makroblok moramo izračunati srednju apsolutnu razliku (MAD – Mean Absolute Difference) na sljedeći način:

$$MAD(i, j) = \frac{\sum_{p=1}^n \sum_{q=1}^m |C_0(p, q) - C_0(p + i, q + j)|}{mn}$$

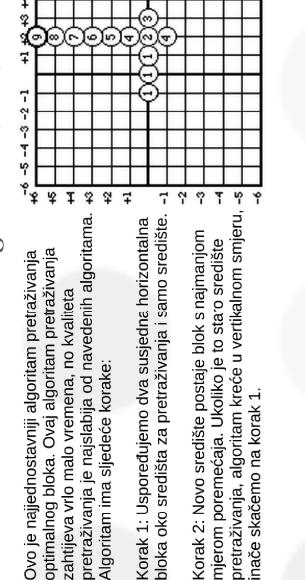
Cilj je naći vektor pomaka ( $i, j$ ) takav da je pripadni MAD ( $i, j$ ) minimalan. U prethodnoj formuli konzili smo srednju apsolutnu razliku kao mjeru pronašlaška najboljeg vektora pomaka.

Vetina komercijalnih kodera koristi mjeru poremećaja sumu apsolutnih razilika (SAD – Sum of Absolute Differences) zbog nje cijene. Recimo da je SAD sličan MAD ali bez dijeljenja sa  $(mn)$ .

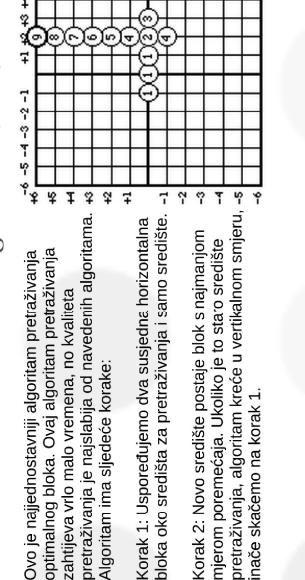
## Algoritmi pretraživanja



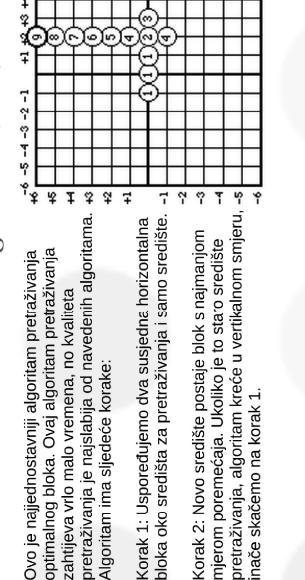
## Two Dimensional Logarithmic Search (TDL)



## Orthogonal Search Algorithm (OSA)



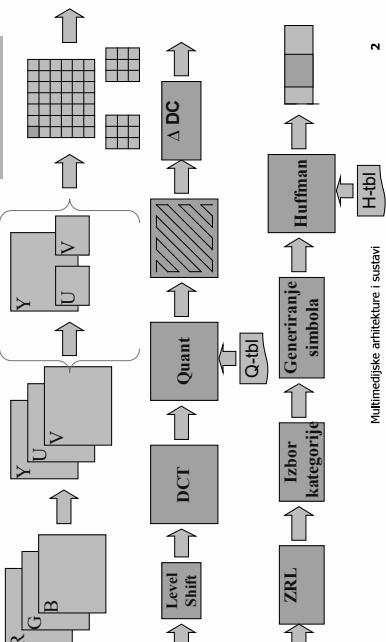
## One at a Time Algorithm (OTA)



# Multimedejske arhitekture i sustavi

prof.dr.sc. Mario Kovač  
izv.prof.dr.sc. Hrvoje Mlinarić

## JPEG koder



Multimedejske arhitekture i sustavi

1

## Cilj...

- U prethodna dva predavanja uzeли smo primjer jednog tipičnog mm algoritma
- U nastavku cilj ће nam biti vidjeti kako se neki od algoritama implementiraju u stvarnom svijetu : IZVEDBENI POGLED NA TEHNOLOGIJE
- S obzirom da је cilj diplomskog studija omogućiti rješavanje kompleksnih problema...

3

## Količine podataka

- Jednostavan izračun daje nam okvirne količine nekih tipičnih formata koje ste upoznali

Rezolucija slike	Bita/pikselu	Veličina
640x480	8	307kB
1280x1024	24	3,9MB

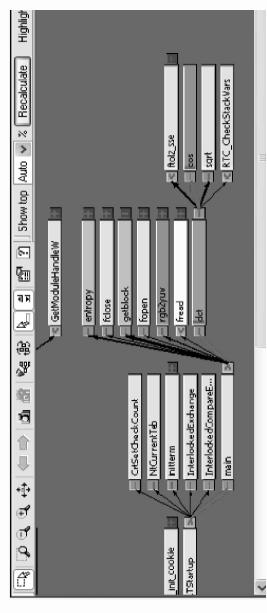
Video rezolucija	Slika/s	Veličina/s
640x480 (8bpp)	10	3 MB/s
1280x1024 (24bpp)	30	120 MB/s

Multimedejske arhitekture i sustavi

4

## Najzahtjevниje operacije

- Obrađivati takve podatke se može na različite načine
- Teoretski kad razmatramo kompresiju npr slike ili videa je nužna...no ako je želimo izvesti onda se susrećemo sa tipičnim inženjerskim problemima
- Pokušajmo predviđeti kompleksnost nekih funkcijskih blokova a onda se uvjerimo kakva je uistinu kompleksnost



3

## I sada počinje ozbiljna analiza..

Multimedejske arhitekture i sustavi

5

Multimedejske arhitekture i sustavi

6

## Kako to implementirati u SW ili HW

- Najzahtjevnejše operacije pri obradi slike/videa
  - Procjena/predviđanje pokreta !!!!!...!!!
  - DCT
  - RGB-YUV transformacija
  - Entropijsko kodiranje

Multimedijiske arhitekture i sustavi

7

## Analiza izvedbe JPEG algoritma

- Iz tablica koje smo ranije proučili vidljivo je da je podataka kod slike i video puno... no mi mislimo da su naši današnji processori brzi...
- NAPOMENA: u okviru ovog predmeta nećemo ulaziti u detaljne analize efikasnosti prevoditelja, OS-a i razvojne okoline. No zapamtite da efikasnost rješenja može ZNAČAJNO ovisiti o ovim i nekim dodatnim uvjetima

Multimedijiske arhitekture i sustavi

8

## Konverzija prostora boja

- Kao što ste naučili RGB način zapisa podataka za sliku nije dobar za kompresiju podataka već se za to koristi YUV (YCrCb) prostor
- Prije pokretanja kompresije slike u RGB zapisu konvertira se u YUV
  - Inverzna transformacija obavlja se nakon dekompresije a prije prikaza na zaslonu

Multimedijiske arhitekture i sustavi

9

## Konverzija prostora boja

- RGB-YUV konverzija vrlo se lako može obaviti jednostavnom matričnom operacijom:
$$Y = 0.299 R + 0.587 G + 0.114 B$$
$$Cb = -0.1687 R - 0.3313 G + 0.5 B + 128$$
$$Cr = 0.5 R - 0.4187 G + 128$$
- Vidimo da za izračun svakog piksela treba 9 množenja i 8 zbrajanja (+dohvat, spremanje)
- Ako napravimo jednostavan program za test (Primjer1.exe) možemo dobiti vrio jednostavnu usporedbu efikasnosti:
- RGB2YUV v.1. AVG ≈ 160

Multimedijiske arhitekture i sustavi

10

## Konverzija prostora boja

- Verzija 2: osnovna approximacija
$$Y = 0.299 R + 0.587 G$$
$$Cb = -0.3313 G + 0.5 B + 128$$
$$Cr = 0.5 R - 0.4187 G + 128$$
  - RGB2YUV v.2. AVG ≈ 138
- Verzija 3: poboljšana approximacija
$$\text{data}[i+j] = (\text{byte})(R/4 + G/2);$$
$$\text{data}[i+r] = (\text{byte})(-G/4 + B/2 + 128);$$
$$\text{data}[i+r] = (\text{byte})(R/2 - (G/2) + 128);$$
  - RGB2YUV v.3. AVG ≈ 35!!!
- Ako se napiše u asembleru: za red veličine brže

Multimedijiske arhitekture i sustavi

11

## Konverzija prostora boja

- Verzija 2: osnovna approximacija
$$Y = 0.299 R + 0.587 G$$
$$Cb = -0.3313 G + 0.5 B + 128$$
$$Cr = 0.5 R - 0.4187 G + 128$$
  - RGB2YUV v.2. AVG ≈ 138
- Verzija 3: poboljšana approximacija
$$\text{data}[i+j] = (\text{byte})(R/4 + G/2);$$
$$\text{data}[i+r] = (\text{byte})(-G/4 + B/2 + 128);$$
$$\text{data}[i+r] = (\text{byte})(R/2 - (G/2) + 128);$$
  - RGB2YUV v.3. AVG ≈ 35!!!
- Ako se napiše u asembleru: za red veličine brže

Multimedijiske arhitekture i sustavi

12

## Analiza

- Za jednu vrlo zahtjevnu operaciju uspjeli smo značajno smanjiti procesorske zahtjeve
- No to smo postigli uz **degradaciju kvalitete izračuna podataka**
- Da li je kvaliteta zadovoljavajuća ili ne vrlo je teško empirijski definirati te je zato potrebno napraviti mnogo testova i subjektivnih provjera

Multimedijeke arhitekture i sustavi

13

## Zadatak za vježbu

- neobavezno
- Korištenjem bilo kojeg alata (Visual Studio, Matlab, Mathematica,...) usporedite kvalitetu tri ranije opisane metode aproksimacije konverzije prostora boja
- Postupak:
  - Napišite funkciju koja učitava sliku u boji u RGB
  - Napišite tri različite funkcije transformacije boja (od kojih je jedna po punim formulama)
  - Izračunate MSE za svaku komponentu ( $Y, U, V$ ) za jednu testnu sliku za dva aproksimativna rješenja

Multimedijeke arhitekture i sustavi

14

## Načini optimizacije algoritama

- Vidjeli smo prvi primjer jedne jednostavne metode za optimizaciju nekog algoritma
- Iako se može činiti da je ovo dobar pristup njega na žalost ne možemo primijeniti u većini situacija
- Ponekad nije dozvoljeno unošenje ovako značajnih grešaka u izračune no ipak se računalni zahtjevi moraju značajno smanjiti

Multimedijeke arhitekture i sustavi

15

## Načini optimizacije algoritama

- Neke osnovne grupe optimizacija:
- Smanjenje preciznosti izračuna
- Razvoj ekvivalentnih matematičkih algoritama sa manjom kompleksnost
- Promjena programske razvojne okoline
- Promjena programske izvedbene okoline
- Promjena arhitekture sustava za izvođenje

Multimedijeke arhitekture i sustavi

16

## Načini optimizacije algoritama

- Da bi mogli razmotriti moguće načine optimiranja MORAMO DOBRO POZNAVATI:
  - Algoritme koje optimiramo
  - Programska rješenja koja koristimo
  - Arhitekturu sustava na kojem se algoritmi izvode

Multimedijeke arhitekture i sustavi

17

## Načini optimizacije algoritama

- U projektiranju visokoeffikasnih proizvoda morati ćemo se vrlo često poslužiti SVM dostupnim metodama i njihovim kombinacijama
- U nastavku ćemo proučiti na koji način se može pristupiti optimizaciji i izvedbi nekih ključnih dijelova multimedijskih algoritama

Multimedijeke arhitekture i sustavi

18



## 2D DCT izvedba

- Za izvedbu DCT, ali i svake druge operacije programski ili na integriranom sklopu, od velike je važnosti procijeniti ukupnu efikasnost s potrebnom količinom resursa/logike (što na kraju rezultira u vremenu obrade ili površini silicija, a time i cijeni).
- Vojne primjene

Multimedijeke arhitekture i sustavi

25

## 1D DCT

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{(2x+1)\pi u}{2N}\right)$$

- Ako pogledamo kompleksnost vidimo da je približna kompleksnost ove formule za svaki element:
  - $\alpha() \Sigma f^* \cos()$
  - Teoretski 9 množenja, 7 zbrajanja, 8 izračuna cos() funkcije (koja u parametrima ima 1 dijeljenje, 3 množenja, 1 zbrajanje)

Multimedijeke arhitekture i sustavi

26

## 2D DCT preko 1D DCT

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{(2x+1)\pi u}{2N}\right)$$

- Priličan problem je cos() funkcija
- S obzirom na diskretizirane vrijednosti parametara cos() faktori se NE računaju već se koriste unaprijed izračunate vrijednosti pohranjene u tablicu (tzv.lookup tablica) te se kao takvi oni obično i kombiniraju sa cos() parametrom.
- Samo ovim postupkom ZNAČAJNO smo smanjili kompleksnost računanja
- No i sa tako izvedenim cos() kompleksnost je velika

Multimedijeke arhitekture i sustavi

27

## 2D DCT

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{(2x+1)\pi u}{2N}\right)$$

- 2D DCT za blok 8x8:
  - 4096 množenja, 4032 zbrajanja
  - Samo za jednu sliku 1280x1024
    - 83886080 množenja, 82575360 zbrajanja

Multimedijeke arhitekture i sustavi

28

## 2D DCT preko 1D DCT

- No i ovo što smo do sada predložili nije dovoljno da u stvarnom svjetu izvedete neki algoritam za kompresiju uz razumno potrošnju resursa (procesora, vremena, energije,...)
- Potreban je novi pristup računanju....
- BRZI ALGORITMI

Multimedijeke arhitekture i sustavi

29

## 2D DCT

Multimedijeke arhitekture i sustavi

30

## Simetrije, tablice,..., i još ...

- Kad razmatramo kako implementirati neki algoritam moramo pokušati proučiti problem sa mnogih strana
- U slučaju DCT analizom formula za 1D DCT mogu se uočiti neke simetrije članova koji se računaju te uz malo trigonometrije reducirati broj izračuna cos() funkcije
- Isto tako mogu se uočiti neki parovi elemenata koji se zbrajaju i oduzimaju

Multimedijске arhitekture i sustavi

31

## Ligtenberg, Vetterli

- Na taj način mogu se identificirati elementi:
  - $C_k$
  - $S_k$
  - $s_{jk}$  i  $d_{jk}$

$$\begin{aligned} C_k &= \cos(k\pi/16) \\ S_k &= \sin(k\pi/16) \\ C_1 &= S_1 = 0.9808 \\ C_2 &= S_2 = 0.9239 \\ C_3 &= S_3 = 0.8315 \\ C_4 &= S_4 = 0.7071 \\ C_5 &= S_5 = 0.5556 \\ C_6 &= S_6 = 0.3827 \\ C_7 &= S_7 = 0.1951 \end{aligned}$$
$$\begin{aligned} s_{jk} &= s(j) + s(k) \\ d_{jk} &= s(0) - s(j) \\ s_{12} &= s(0) + s(7) \\ s_{13} &= s(1) + s(6) \\ s_{14} &= s(2) + s(5) \\ s_{15} &= s(3) + s(4) \\ s_{16} &= s(4) - s(3) \\ s_{17} &= s(5) - s(2) \\ s_{18} &= s(6) - s(1) \\ s_{19} &= s(7) - s(0) \\ d_{12} &= s_1 + s_8 \\ d_{13} &= s_1 - s_5 \\ d_{14} &= s_1 - s_4 \\ d_{15} &= s_1 + s_3 \\ d_{16} &= s_1 - s_2 \\ d_{17} &= s_1 + s_0 \\ d_{18} &= s_1 - s_7 \\ d_{19} &= s_1 + s_6 \end{aligned}$$

Multimedijске arhitekture i sustavi

32

- Na temelju prethodnih analiza Ligtenberg i Vetterli su izveli formule za 1D DCT:
  - $2S(0)=C4(s07+s12)+(s34+s56)$
  - $2S(1)=C1d07+C3d16+C5d25+C7d34$
  - ...

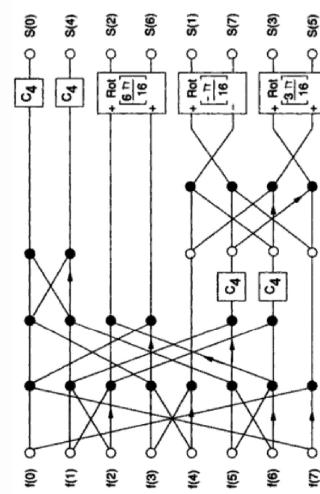
- Također se može vidjeti da ako dva podatka x i y želimo rotirati za kut  $k\pi/16$  tada nove vrijednosti iznose:
  - $X=Ck^*x+Sk^*y$
  - $Y= -Sk^*x+Ck^*y$

Multimedijске arhitekture i sustavi

33

## Ligtenberg, Vetterli

- Kao rezultat ovog razmatranja Ligtenberg i Vetterli su izveli algoritam koji zahtjeva samo 13 operacija množenja i 29 operacija zbrajanja za računanje 1-D DCT.
- Već ovaj algoritam pokazuje drastično smanjenje broja matematičkih operacija u usporedbi s konzervativnim načinom računanja.



Multimedijске arhitekture i sustavi

34

## Brzi DFT....

- Bitno drugačiji pristup računanju DCT predložili su neki autori koji su se intenzivno bavili i proučavanjem brzih algoritama za računanje diskretne Fourierove transformacije (DFT). Tako je u svom radu Haralick pokazao da se DCT s N točaka može izračunati koristeći dvije brze Fourierove transformacije (FFT) s N točaka koristeći se simetrijom ulaza.
- Kroz brojne članke (npr. Tseng i Miller) pokazalo se kako se DCT može efikasnije izračunati na način da se, uz simetrično raspoređene ulaze, koriste samo realni dijelovi prvih N koeficijenata iz DFT s 2N točaka.

Multimedijске arhitekture i sustavi

35

Multimedijске arhitekture i sustavi

36

## Skalirana DCT

- Ovi radovi pored toga uvođe dodatnu prednost:
  - Naimje kod konzervativnog pristupa ili kod primjera Lightenbergovog i Vetterlievog algoritma, kvantizacija koeficijenata koja slijedi transformaciju uvedi dodatne matematičke operacije u postupak.
  - Algoritam Tsenga i Millera naziva se tzv. Skaliranim algoritmom jer za dobivanje DCT koeficijenata rezultati DFT se moraju pomnožiti s određenim konstantama (skalirati).
  - Ako nakon postupka transformacije odmah slijedi kvantizacija tada se kvantizacijski koraci mogu prije pomnožiti s konstantama skaliranja. Na taj način na izlazu će se bez dodatnih operacija dobiti kvantizirani DCT koeficijenti. Koristeći se ovim račinom razmisljanjem, DCT s 8 točaka koja se koristi u JPEG normi može biti zamjenjena DFT-om s 16 točaka i operacijom skaliranja.

Multimedijeke arhitekture i sustavi

37

## Brzi algoritmi – što se koristi

- Tipično se za izračun 1D DCT preko 1D FFT koristi teorija:
  - **AAN algoritam za skalirani 1D DCT (8 točaka):**
    - 5 množenja, 29 zbrajanja, 16 dvojnih komplementa
  - **Kovač, Ranganathan algoritam za skalirani 1D DCT (8 točaka):**
    - 5 množenja, 29 zbrajanja, 12 dvojnih komplementa

Multimedijeke arhitekture i sustavi

38

## KR algoritam

- Korak 1:  
 $b_0 = a_0 + a_7;$   
 $b_4 = a_2 + a_5;$   
Korak 2:  
 $c_0 = b_0 + b_5;$   
 $c_4 = b_0 - b_5;$   
Korak 3:  
 $d_0 = c_0 + c_3;$   
 $d_4 = c_2 - c_5;$   
Korak 4:  
 $e_0 = d_0;$   
 $e_4 = m4 * d_6$
- ..... I tako daje

Multimedijeke arhitekture i sustavi

39

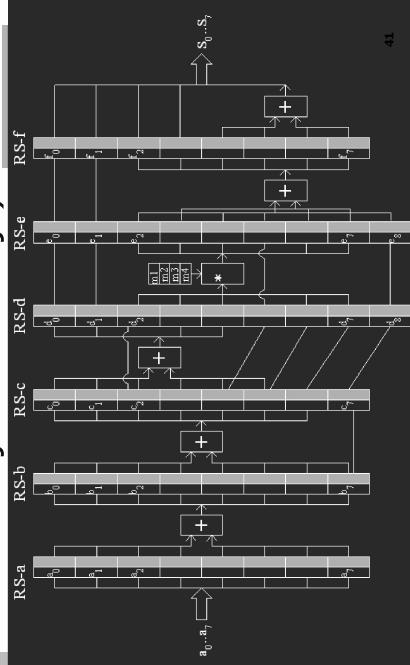
## Uštede ....

- Korištenjem ovih brzih algoritama za jednu sliku 1280x1024 potrebno je približno:
  - 1638400 množenja
  - 9502720 zbrajanja
  - 3932160 dvojnih komplementa
- Najvažnije je da je broj operacija množenja (SPORO u procesoru) smanjen cca. 15 puta

Multimedijeke arhitekture i sustavi

40

## I što nam to još omogućuje... (o tome će riječi biti kasnije)



41

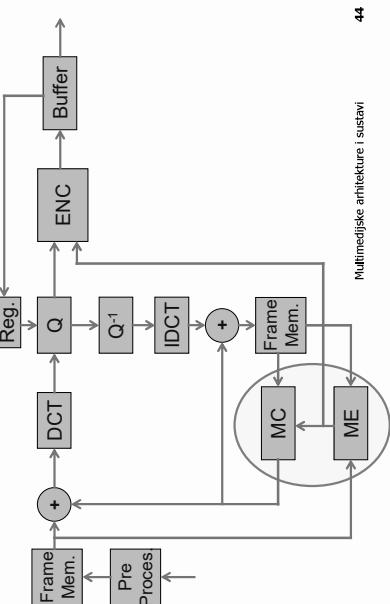
## Zadatak za vježbu (neobavezno)

- Napišite program koji će računati 2D DCT za neku ulaznu sliku koristeći osnovni (teorijski) algoritam i koristeći odvojivost te 1D DCT

Multimedijeke arhitekture i sustavi

42

MPEG koder



43

Multimedijeke arhitektura i sustavi

## Intra i inter-blokovska kompresija

- **Unutar-blokovska** kompresija
  - Uklanjanje informacija visokih frekvencija koje ljudsko oko ne može prepoznati
  - Isto kao kod kompresije slike (JPEG) što je prije objašnjeno
- **Među-blokovska** kompresija
  - Smanjuvanje vremenske redundancije
  - Obje se metode zasnivaju na karakteristikama ljudskog vizualnog sustava (HVS)

Multimedielle Architektur i sustavi

Vremenska redundancija

vremenska redundancija

CARJUMP

MPEG



Multimedijiške arhitektire i sustavi

Multimediske arhitekture i sustavi

Međublokovska kompresija

- vro malo mijenjaju ili su čak nepromijenjeni  
Pozadina slike često se ne mijenja

CARJUMP

vremenska redundancija

47

Multimedijiške arhitekture i sustavi

46

## Vremenska redundancija

### Procjena i kompenzacija pokreta

#### ■ Procjena pokreta (motion estimation, ME):

- Izdvajanje dijelova slike (segmentacija pokreta)
- Opisivanje pokreta (procjena)
- Izvodi se u koderu

BRZI/AUTO

#### ■ Kompenzacija pokreta (motion compensation, MC):

- Korištenje rezultata procjene pokreta
- Izvodi se u dekoderu

Multimedijeke arhitekture i sustavi

Multimedijeke arhitekture i sustavi

50

### Procjena i kompenzacija pokreta

#### ■ Procjena pokreta (motion estimation, ME):

- Kako bi omogućili jednostavniju implementaciju nameću se potrebna pojednostavljenja:

- Segmentacija na pravokutne dijelove unaprijed znanih dimenzija (blok)
- Svaki je pokret translacijski (vektor pomaka)

Multimedijeke arhitekture i sustavi

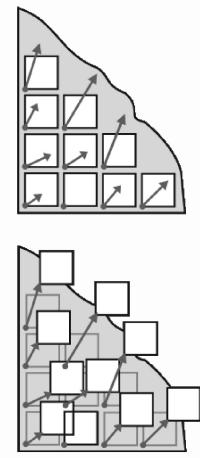
Multimedijeke arhitekture i sustavi

51

## Procjena i kompenzacija pokreta

- Algoritmi procjene pokreta za svaki blok računaju pripadni vektor pomaka

- Vektor pokazuje izvođenje bloka u prethodnoj slici prije pomaka na poziciju u trenutnoj slici

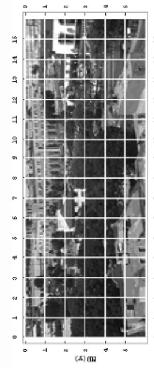


Multimedijeke arhitekture i sustavi

52

## Procjena i kompenzacija pokreta

- Radi efikasnije obrade za procjenu pokreta dovoljna je komponenta Y



Multimedijeke arhitekture i sustavi

53

## Slika kao pojedinačni objekt pri kompresiji

- Radi efikasnije obrade za procjenu pokreta dovoljna je komponenta Y

Multimedijeke arhitekture i sustavi

54

## Mjera poremećaja

- Mjera sličnosti dvaju blokova jest **mjera poremećaja** slike od jednog trenutka do drugog
- Procjena pokreta svodi se na **optimizaciju mjere poremećaja** kao kriterijske funkcije
- Algoritmi u definiranom **području pretraživanja** nastoje pronaći minimum kriterijske funkcije
  - Promjena slike na tom je mjestu najmanja
  - proglašavamo da se blok pomakao duž vektora

Multimedijiske arhitekture i sustavi

55

## Mjera poremećaja – primjer

- Neke moguće mjere poremećaja dvaju blokova:
  - **Srednja kvadratna pogreška (MSE)** – računski zahtjevna 
$$MSE(x, y; d_x, d_y) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [Y(x+i, y+j, t_i) - Y(x+d_x + i, y+d_y + j, t_0)]^2$$
  - **Srednji apsolutni poremećaj (MAD)** – široko prihvaćena 
$$MAD(x, y; d_x, d_y) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |Y(x+i, y+j, t_i) - Y(x+d_x + i, y+d_y + j, t_0)|$$
  - **Broj podudaračkih slikovnih elemenata (MPC)** 
$$MPC(x, y; d_x, d_y) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} T[Y(x+i, y+j, t_i), Y(x+d_x + i, y+d_y + j, t_0)]$$

Multimedijiske arhitekture i sustavi

56

## Algoritmi

- Ovisno o načinu pretraživanja razlikujemo:
  - **Algoritam potpunog pretraživanja**
    - pretražuje sve točke unutar područja pretraživanja
    - računski vrlo zahtjevan
    - Daje najbolji mogući rezultat
  - **Nepotpuni (brzi, napredni) algoritmi**

Multimedijiske arhitekture i sustavi

57

## Algoritam potpunog pretraživanja

- Veoma zahtjevan algoritam za računalne resurse:
  - Izračun jedne vrijednosti poremećaja (MAD):
    - Dohvat podataka (vrlo zahtjevno)
      - Za blok 16x16: 256 oduzimanja + 1 pomak (dijeljenje sa 256)
    - Ako područje pomaka iznosi +-8 u obje dimenzije:
      - $(2^{8+1})^*(2^{8+1})^*(256+\text{dohvat}) = 73984 + \text{dohvat}$
      - Pronalaženje minimuma
        - Za jednu sliku 1280x1024  $5120 * 73984 = 378.798.080 + \text{dohvat}$
        - Za 30 slike u sekundi:  $1.1 * 10^{10} \text{ zbrajanja/s}$

Multimedijiske arhitekture i sustavi

58

## Algoritam potpunog pretraživanja

- Vidljivo je da algoritam potpunog pretraživanja daje optimalni rezultat ali je nažalost u praksi teško primjenjiv
  - Koriste ga uglavnom samo HW koderi
  - Upravo zato potreba za brzim algoritmima
  - Prednosti: manje potrebnih operacija
  - Nedostaci: veće razlike koje se trebaju kodirati a time je i izlazna količina podataka veća
  - Kvalitetna nekih algoritama zadovoljavajuća te su blizu potpunom pretraživanju
  - Problemi sa tipovima pokreta (objašnjeno kasnije)

Multimedijiske arhitekture i sustavi

59

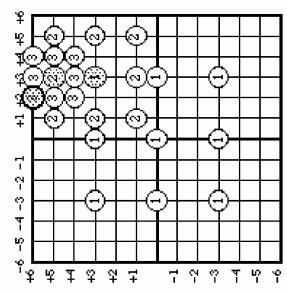
## Brzi algoritmi

- **Brzi, napredni algoritmi pretraživanja** usmjeravaju pretraživanje ovisno o vrijednosti poremećaja slike
- Neki primjeri:
  - Logaritamsko pretraživanje (LOG)
    - Pretraživanje u tri koraka (3SS)
    - Ortogonalno pretraživanje (ORT)
    - Algoritam gradijentnog spusta temeljen na blokovima (BBGDS)

Multimedijiske arhitekture i sustavi

60

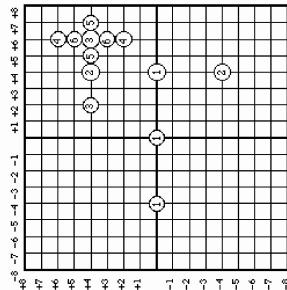
## Algoritmi – primjer (3SS)



Multimedijске arHITEKURE I sustavi

61

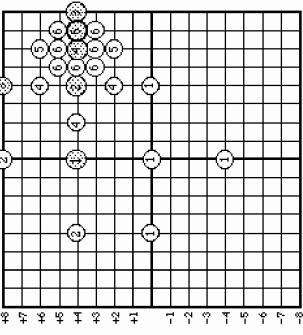
## Algoritmi – primjer (ORT)



Multimedijске arHITEKURE I sustavi

62

## Algoritmi – primjer (LOG)



Multimedijске arHITEKURE I sustavi

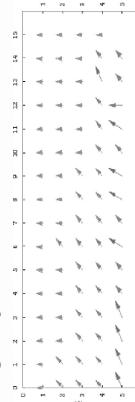
63

## Algoritmi – primjer (LOG)

LOG

### Raspodjela vektora pomaka

- Prikaz upućuje na ravnomjerno raspoređenepokrete između dvije slike sekvene
- Vektori su većinom kratki – centrirana raspodjela vektora - tipična za sekvene iz stvarnog svijeta

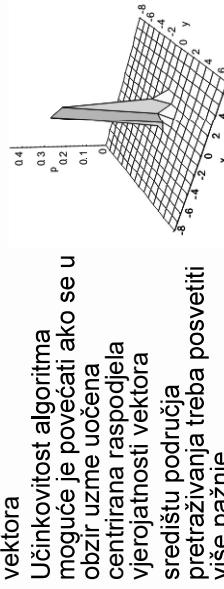


Multimedijске arHITEKURE I sustavi

64

### Raspodjela vektora (2)

- Prebrojavanje vektora za sve moguće parove komponenti ( $x,y$ ) tijekom cijele sekvence daje relativne frekvencije vektora
- Učinkovitost algoritma moguće je povećati ako se u obzir uzme uočena centrirana raspodjela vjerojatnosti vektora
- središtu područja pretraživanja treba posvetiti više pažnje



Multimedijске arHITEKURE I sustavi

65

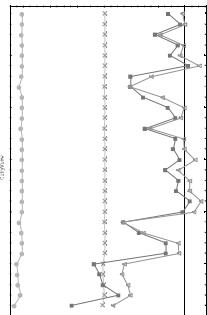
## Primjeri usporedbe algoritama

- Parametri koji određuju učinkovitost algoritma:
  - Ukupna mjera poremećaja nakon procijene pokreta treba biti što manja (bolja kompresija)
  - Ukupni broj ispitnih točaka također treba biti što manji (veća brzina)
  - Potreban je kompromis **kompresija ↔ brzina**
  - Primjer sekvenci koje sadrže različite vrste pokreta:
    - "City View" - ujednačeno raspoređen i malen pokret
    - "Car Jump" - ločran i velik pokret
    - "Troops" - ujednačeno raspoređen i velik pokret

Multimedijeke arhitekture i sustavi  
67

## Implementacija – primjer

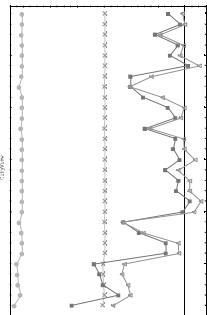
- Norma za kompresiju **ne specificira** postupke procijene pokreta
- Kvalitetna aplikacije znatno ovisi o načinu procijene pokreta

"City View" sekvenca – broj ispitnih točaka:  
  
Legend:

- 3SS: ▲
- ORT: ✕
- BBGDS: ■
- LOG: ◆

Multimedijeke arhitekture i sustavi  
68

## Rezultati (1)

- "City View" sekvenca – broj ispitnih točaka:  
  
Legend:
  - 3SS: ▲
  - ORT: ✕
  - BBGDS: ■
  - LOG: ◆

Multimedijeke arhitekture i sustavi  
69

## Rezultati (2)

- Centrirana rasподјела: BBGDS najbolji
- ORT i 3SS konstantan i relativno velik broj ispitnih točaka

Multimedijeke arhitekture i sustavi  
70

## Rezultati (3)

- Učinkovitost pojedinog algoritma ovisi o vrsti sadržanog pokreta
- Učinkovit postupak će na temelju vrste pokreta heuristički odabratи najprikladniji algoritam: **adaptivni algoritmi**

Multimedijeke arhitekture i sustavi  
71

## Izvedba

- Ovime smo ukratko analizirali neke najzajtevnejše algoritme pri obradi i kompresiji multimedijiskih podataka
- Vidjeli smo i načelnu kompleksnost izračuna bez ulazeњa u previše detalja

Multimedijeke arhitekture i sustavi  
72

Optimizacije

- Neke osnovne grupe optimizacija:
    - Smanjenje preciznosti izračuna
    - Razvoj ekivalentnih matematičkih algoritama sa manjom kompleksnostju
    - Promjena programske razvojne okoline
    - Promjena programske izvedbene okoline
    - Promjena arhitektture sustava za izvođenje

Multimedijičke arhitektire i sistavi

Osnovni načini SW podrške za MM

- Morate biti svjesni da se program pisan u Java ili npr. C# sporije izvodi od programa koji su prevedeni u kod za ciljni procesor
  - Dodatno: viši programski jezici generiraju sporiji kod od onoga pisano u kombinaciji sa asemblerom

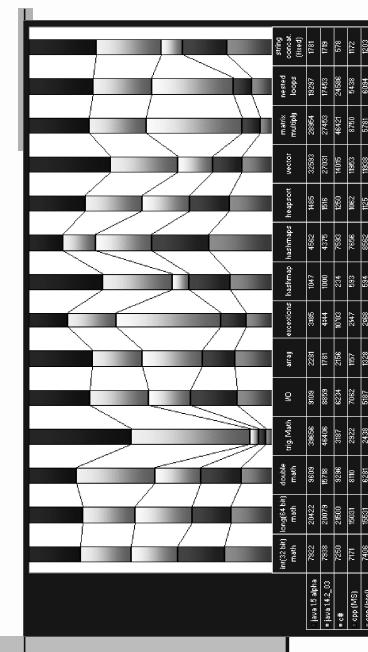
Multimedijiške arhitektire i sistavi

## Performance

- Morate biti svjesni da se program pisan u Java ili npr. C# sporije izvodi od programa koji su prevedeni u kod za ciljni procesor
  - Dodatno: viši programski jezici generiraju sporiji kod od onoga pisano u kombinaciji sa asemblerom

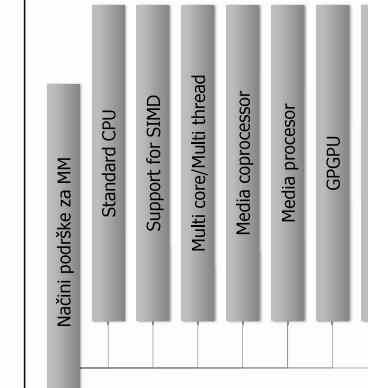
74

Usperedba Java, C#, C++



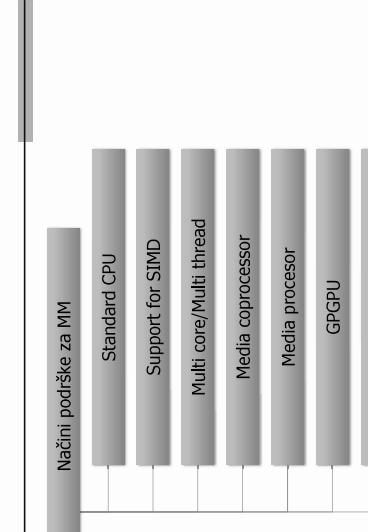
[www.tommi-systems.de/main-Dateien/reviews/announces/benchmarks.htm](http://www.tommi-systems.de/main-Dateien/reviews/announces/benchmarks.htm)

Osnovni načini HW podrške za MM



104

ovni načini HW podrške za MM



8

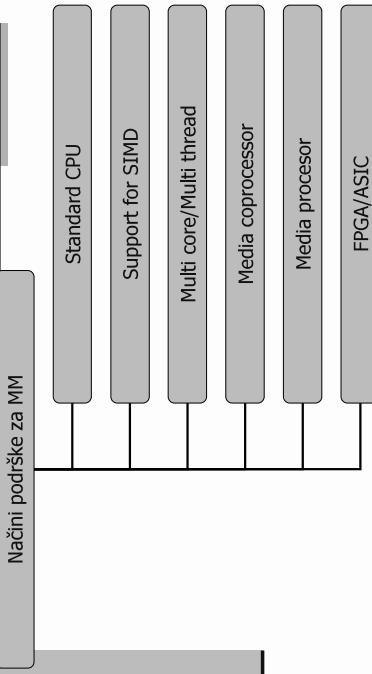
Multimedijijske arhitektire i sistavi

## Multimedijiske arhitekture i sustavi (dio 4)

prof.dr.sc. Mario Kovač  
izv.prof.dr.sc. Hrvoje Mlinarić



### Osnovni načini CPU podrške za MM



### Standardni CPU

- Ako se prisjetimo Arhitekture računala onda znamo da obični procesori mogu izvesti jednu ALU operaciju po periodu
- Operacija je širine koju ima ALU
- Kako bi ubrzali izvođenje moramo mnogo pažnje posvetiti dohvatu podataka te optimizacijama petlji

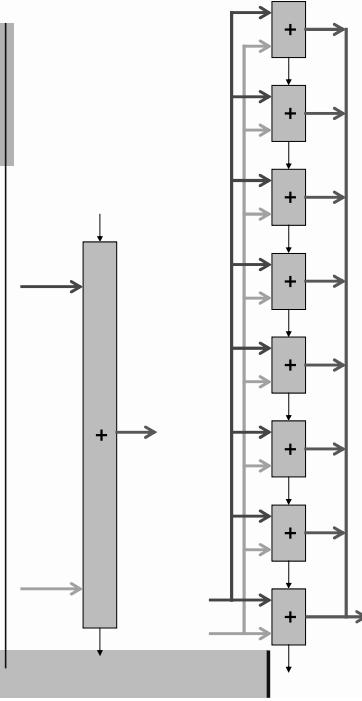
### Standardni CPU

- Za DSP operacije (npr DCT) izuzetno je korisno ako procesor ima sklopovski izvedeno množenje i pripadnu naredbu
- Dodatna značajina prednost ako postoji naredba množenja sa zbrajanjem (Multiply Accumulate)
- Kod primjera računanja DCT, DFT i slično imamo većinu "leptir" operacija kod kojih je MLA osnovna karika

### Podrška za SIMD

- SIMD (Single Instruction Multiple Data)
  - Arhitektura puta podataka u procesoru koja omogućuje obradu više podataka (u načelu manje preciznosti) sa jednom naredbom
- Osnovna ideja:
  - Ako imamo npr 64 bitovnu ALU onda je potpuno neefikasno s njom obrađivati 8 bitovne podatke
  - Reorganizirati ALU na način da se može "podijeliti"

### ALU – obična i SIMD



# MM proširenja

- Ovo (osnovno) načelo služi kako bi se obrada multimedijiskih algoritama značajno ubrzala

## Primjeri:

- Prvi: VIS, PA-RISC
- Stari: MMX, 3DNNow!
- Ne tako stari: SSE4
- Novi: AVX, AVX2, NEON (ARM)
- Najnoviji: AVX-512

- AVX-512 instruction are encoded with the new EVEX prefix. It allows 4 operands, 7 new 64-bit opmask registers, scalar memory mode with automatic broadcast, explicit rounding control, and compressed displacement memory addressing mode. The width of the register file is increased to 512 bits and total register count increased to 32 (registers ZMM0-ZMM31) in x86-64 mode.

	AVX-512 Subset	F	C0	ER	PF	VL	BW	DQ	IFMA	VBMI
Xeon Phi/200 Co-processor (2016)		Yes	Yes	No						
Skylake EP/CX Xeon "Purley" Processors (expected in H2 2017)	Yes	Yes	No	Yes	No	Yes	No	Yes	No	

Multimedijiske arhitekture i sustavi

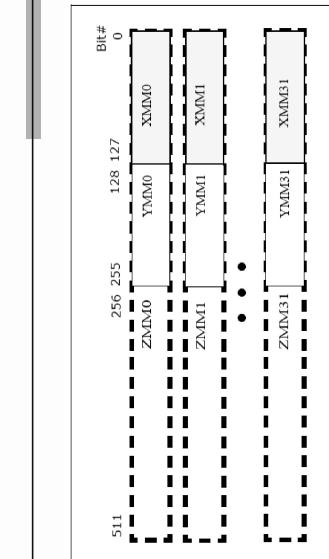
8

- AVX-512 consists of multiple extensions, not all meant to be supported by all processors implementing them. The instruction set consists of the following:
  - AVX-512 Conflict Detection Instructions (CDI) – efficient conflict detection to allow more loops to be vectorized supported by Knights Landing[1]
  - AVX-512 Exponential and Reciprocal Instructions (ERI) – exponential and reciprocal operations designed to field implement transcendental operations, supported by Knights Landing[1]
  - AVX-512 Prefetch Instructions (PFI) – new prefetch capabilities supported by Knights Landing[1]
  - AVX-512 Vector Length Extensions (VL) – extends most AVX-512 operations to also operate on XMM (128-bit) and YMM (256-bit) registers (including XMMS1/XMM31 and YMMS1/YMM31 in x86\_64 mode)[2]
  - AVX-512 Byte and Word Instructions (BW) – extends AVX-512 to cover 8-bit and 16-bit integer operations[2]
  - AVX-512 Doubleword and Quadword Instructions (DQ) – enhanced 32-bit and 64-bit integer operations[2]
  - AVX-512 Integer Fused Multiply Add (IFMA) – fused multiply add for 52-bit integers [2/21/746]
  - AVX-512 Vector Byte Manipulation Instructions (VBM1) adds vector byte permutation instructions which are not present in AVX-512BW[2]
  - Only the core extension AVX-512F (AVX-512 Foundation) is required by all implementations; desktop processors will additionally support CDI, VL, and BW/DQ, while computing coprocessors will support CDI, ERI and PFI.

9

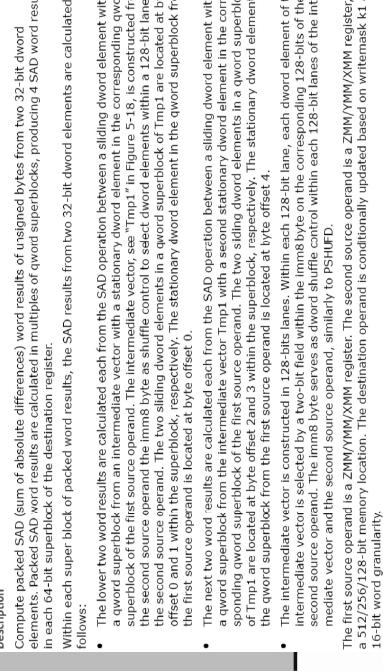
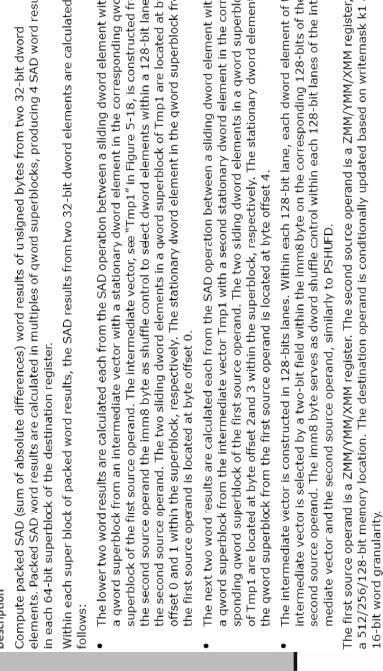
Multimedijiske arhitekture i sustavi

# AVX-512



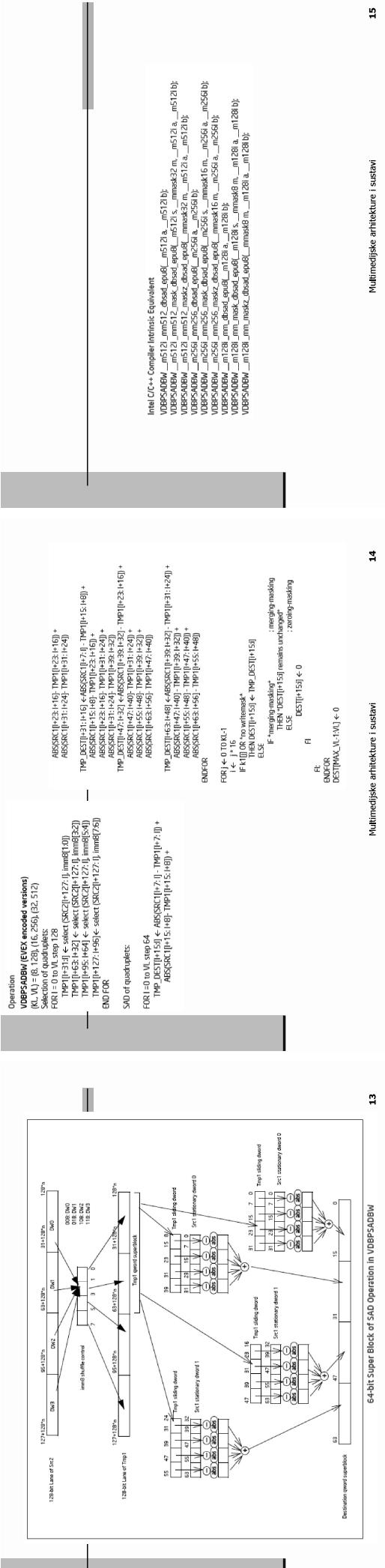
512-Bit Wide Vectors and SIMD Register Set

Multimedijiske arhitekture i sustavi



Multimedijiske arhitekture i sustavi

10



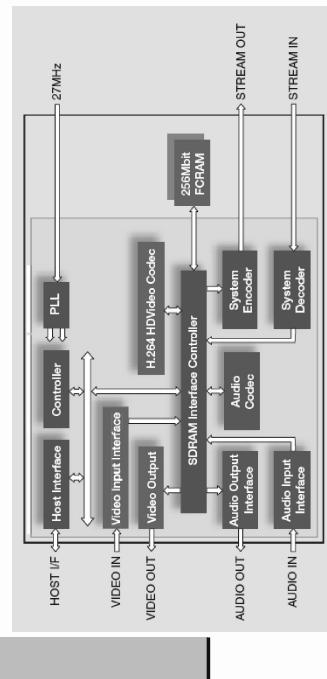
Multicore/multithread

- U prethodnim predavanjima:
    - jedan CPU
  - Danas: vrlo zahtjevni zadaci (lijnij) mogu se na još jedan način koristiti više dretvi (thread) ili (core)
  - Bez ulaska u teoriju, jasno je djelomičnom parallelizacijom postići veća brzina

Što je medijski koprocesor

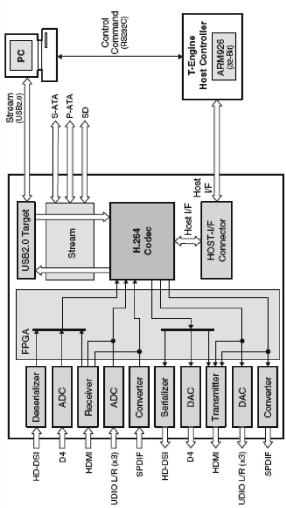
- Standardni procesori nisu pogodni za mnoge primjene
  - Iz dosadašnjeg izlaganja znamo kakva arhitektura je pogodna za visokozahtjevne algoritme i puno podataka
  - Moguće rješenje:
    - Zahtjevni dijelovi algoritma obrađuju se u posebnom procesoru koji se stavlja u sustav i služi samo toj funkciji
    - -> Medijski koprocessor

## Primjeri: Fujitsu



Fujitsu

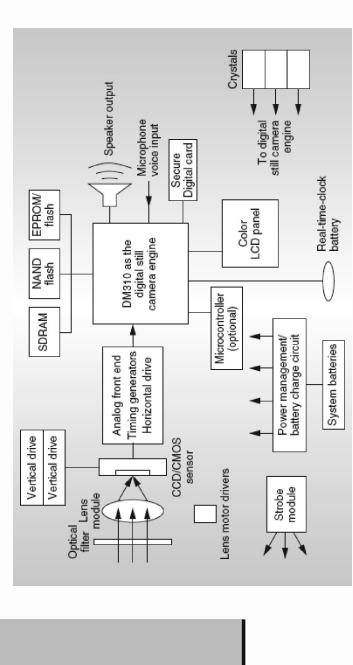
Primjer sustava



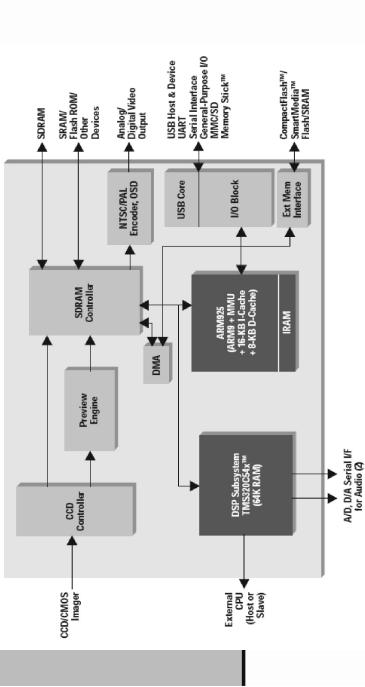
Primjer: TI



卷之三

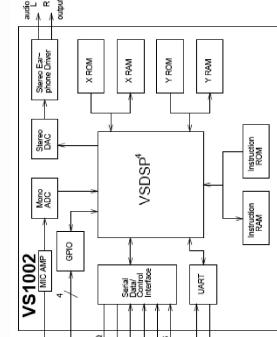


II

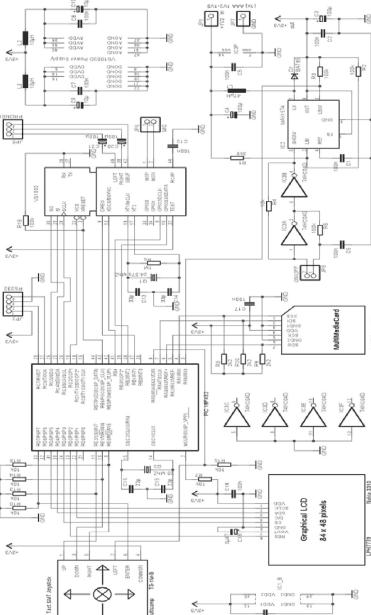


VLSI Solution

## ■ VS 1002: MP3 decoder IC



Primjer jednostavnog MP3 playera



Oníčí nroč

- PIC18LF45x** jednostavan CPU, radi na 20MHz, upravlja radom sustava, upravlja s LCD, tipkama, FLASH karticom,...  
**VS1002D** MP3 koprocessor, radi samo dekodiranje

VS1002



## Medijski koprocesor

- U osnovi : DSP sa mnoštvom periferija
- Jeffinji od procesora opće namjene
- Performanse prilagođene aplikaciji (manje od GPP)
- Manja potrošnja
- Programabilan !! (mogućnost poboljšanja i dodavanja aplikacija)
- Predviđen za porodicu algoritama
- 32-bitovni CPU (npr. ARM) obično dobar za ostale poslove (mreža, user if, ...)

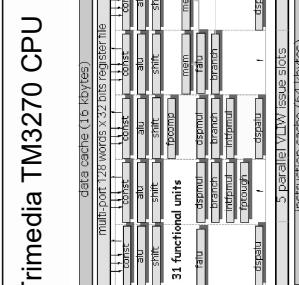
## Medijski procesor

- U dosadašnjim primjerima procesor je bio prevenstveno projektiran za "opće" zadatke
- Za obradu multimedije koristile su se "poboljšanja"
- Novi pristup: procesor se projektira PRVENSTVENO za obradu multimedijskih podataka, a ostale stvari može obrađivati tek sporedno
  - -> Medijski procesor

## Medijski procesor

- Procesor se projektira sa ciljem visokih performansi za određen skup algoritama
- No to je još uvijek PROCESOR: može se programirati
  - Prednosti: promjena algoritama, dodavanje funkcionalnosti,...

## Primjer: NXP



## TM

- 31 funkcionalna jedinica
- 5 paralelnih izvedbenih polja
- Very Large Instruction Word (VLIW) arhitektura
- Neke naredbe omogućuju SIMD
- VLIW ima prednosti nad superscalar arhitekturom jer parallelizam ne određuje procesor (scheduling unit on CPU) već programer (compiler tijekom dizajna)
- Procesor jeftiniji i bri

## Medijski procesor

- Izuzetno visoke performanse (veće od GPP, medijskih koprocesora)
- Složen postupak programiranja
- Paralelno izvođenje operacija unutar jedne naredbe
- Još uvijek programabilan
- Podrška za opće zadatke mora biti osigurana od dodatnog procesora

## FPGA/ASIC

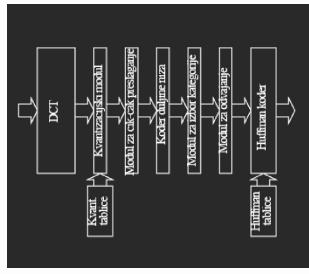
- Radi što većih performansi a niže cijene krajnja mogućnost je projektirati sklop koji sklopovski izvodi izabran algoritam

### Dva pristupa

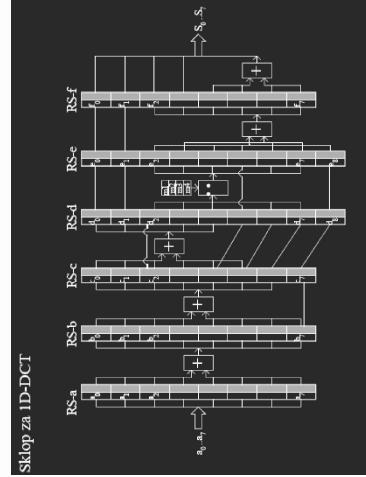
- FPGA (Field Programmable Gate Array)
- ASIC (Application Specific IC)

## JAGUAR

### HW JPEG encoder



## JAGUAR



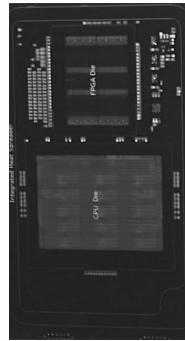
## VLSI/ASIC

- Najpogodnije rješenje za zadani algoritam: performanse, potrošnja, cijena,...
- Jednostavni za integraciju i korištenje

- Nemogućnost poboljšanja, nadogradnje
- Samo jedan dobavljač: opasnost

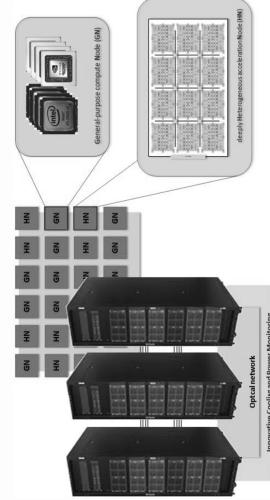
## Heterogeni procesori

- Najnoviji pristup poboljšanju arhitekture
- CPU + GPU
- CPU + FPGA
- CPU + GPU + FPGA
- Intel Broadwell Xeon with a built-in FPGA (2016)



## MANGO Arhitektura

- Deeply heterogeneous QoS aware architecture

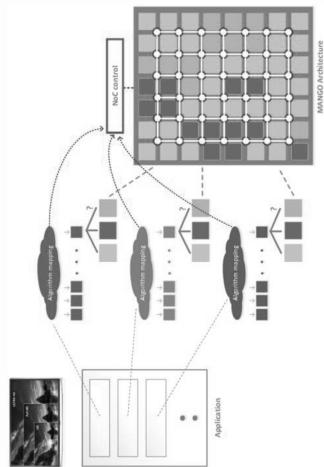


35

36

Multimedijanske arhitekture i sustavi

## Pogled sa strane aplikacija



Multimedijске arhitekture i sustavi

37

## Zaštita podataka

- Simetrični
  - Kriptiranje i dekriptiranje podataka obavlja se sa istim ključem
- Asimetrični
  - Različiti ključevi za kriptiranje i dekriptiranje

Multimedijске arhitekture i sustavi

39

## Kripto algoritmi

- Simetrični
  - Kriptiranje i dekriptiranje podataka obavlja se sa istim ključem
- Asimetrični
  - Različiti ključevi za kriptiranje i dekriptiranje

Multimedijске arhitekture i sustavi

39

## Računalna kompleksnost

- Kompleksnost za 3DES je 8091 operacija za 64 bitovni blok (cca 1000 op. po bajtu)

Step #	operation	#Times	Bytes/7-bits*	Note
1.0	64 bit transposition	2	64x2	16ops
2.1.7	32 bit COPY	16	16	f function
3.1.7	32 bit transpose	16	48x8x16	
4.1.7	48 bit transpose	16	48x8x16	
5.1.7	48 bit XOR	16	3x128	
6.1.7	64 bit TWO	16		
2.1.7	Dimension	8x16		
3.1.7	Algorithmic			
4.1.7	Algorithmic			
5.1.7	Algorithmic			
6.1.7	Algorithmic			
7.1.7	32 bit transpose	16	32x16	K3 function
8.1.7	32 bit transpose	16	56	
9.1.7	32 bit transpose	16	2x16	Present
10.1.7	32 bit transpose	16	48x8x16	D3S Total
11.1.7	48 bit transpose	1	1	3085
12.1.7	32 bit SWAP	1		
13.1.7				
14.1.7				
15.1.7				
16.1.7				
17.1.7				
18.1.7				

Multimedijске arhitekture i sustavi

40

## Diskusija....

- Kako najefikasnije zaštiti digitalni sadržaj ?
- Koji su Vaši prijedlozi.....
- Analiza sigurnosnih rizika...

Multimedijске arhitekture i sustavi

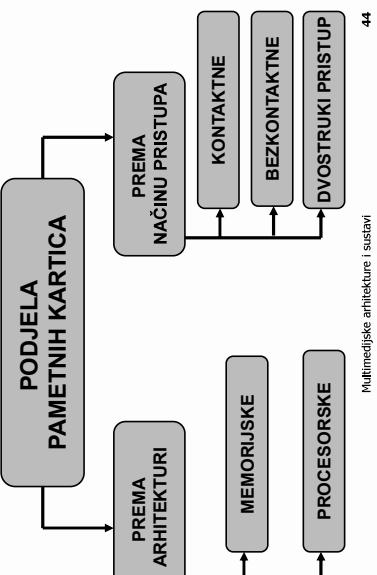
41

## Pametne kartice

## Svrha

- Mobilnost
- Pohrana ili obrada podataka
- Zaštita podataka

## Podjela



## Fizička izvedba

- Pametne kartice
  - Danas su još kartice ali uskoro više ne...
  - Osnovni parametri: ISO 7816-1,2,3,4...
  - Elektronički sklop umetnut u sloj plastike
  - Uspostava komunikacije sa vanjskim svijetom preko metalnih kontaktata ili induktivnom vezom



45

## Dimenzije

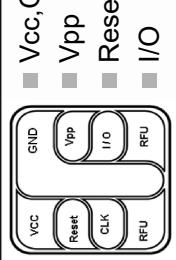
- Kartice – tri formata
  - ID-1 (85,6mm x 54mm)
  - ID-00 (66mm x 33mm)
  - ID-000 (25mm x 15mm)



Multimedijeke arhitekture i sustavi

46

## Kontaktne kartice

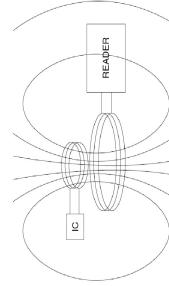


- Vcc, GND napajanje
- Vpp V programiranje
- Reset
- I/O podaci
- CLK takt

Multimedijeke arhitekture i sustavi

## Beskontaktne kartice

- komunikacija na 13,56 MHz
- čitač emitira elektromagnethno polje
- val nosilac na 847,5 kHz



Multimedijeke arhitekture i sustavi

- 106 Kbps

47

## Napajanje

- Postoje 3 metode napajanja pametnih kartica:
- Iz vanjskog izvora napajanja preko kontakata
- Iz vanjskog izvora napajanja induktivno
- Iz baterije koja je ugrađena u karticu ( rijetko )

Multimedijске arhitekture i sustavi

49

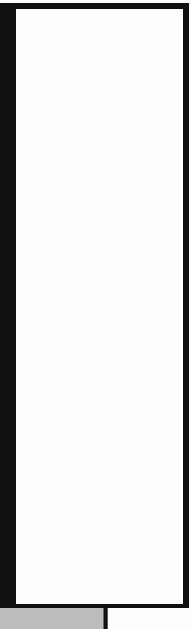
## Memorijske kartice

- Vrlo jednostavne kartice
- Niska cijena
- Cjelokupna logika izvedena je korištenjem kombinacijskih sklopova

Multimedijске arhitekture i sustavi

51

## Memorijske kartice

- 
- Izvanjskog izvora napajanja
  - Izvanjskog izvora napajanja
  - Iz baterije koja je ugrađena u karticu ( rijetko )

Multimedijске arhitekture i sustavi

50

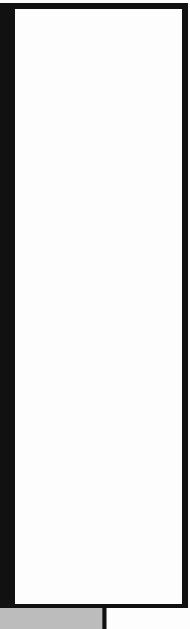
## Memorijske kartice

- Funkcionalnosti:
  - Kartice s pohranjenom vrijednošću
    - Logički zaštićena memorija (ireverzibilna)
    - Kartice s običnom memorijom
      - Pohrana podataka
      - Bez zaštitnih mehanizama
  - Kartice s zaštićenom memorijom
    - Jednostavni mehanizmi kontrole pristupa podacima

Multimedijске arhitekture i sustavi

52

## Procesorske kartice

- 
- Mali računalni sustavi zasnovani na jednostavnijim, uglavno 8-bitovnim procesorima
  - Pored procesora: ROM (za OS), EEPROM (za aplikacije i trajne varijable), RAM(za varijable)
  - Napredne inačice imaju i kripto koprocessor
  - Procesor izvodi Card Operating System (COS)
  - Razni proizvođači nude različite COS

Multimedijске arhitekture i sustavi

54

## Processorske kartice

- Procesor omogućuje pohranu i obradu podataka
- Procesor komunicira sa vanjskim svjetom serijskom vezom definiranom ISO normom
- Komunikacija ne ovisi o OS-u ili procesoru

Multimedijeke arhitekture i sustavi

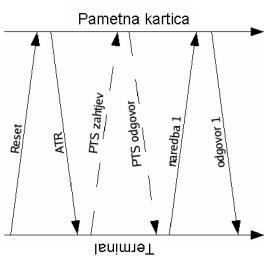
55

## Upostavljanje komunikacije

- ATR
  - Answer to Reset
  - Definiraju se parametri komunikacije
- PTS
  - Protocol Type Selection
  - Definiraju se protokoli naredbe
- Dvosmjerna komunikacija preko čitača Half duplex – ne istovremeno u oba smjera
- Komunikacija podatkovnim paketima
- APDU protokol (Application Protocol Data Unit)
  - Razmijena naredbi i odgovora
  - Odnos "gospodar – sluga" (engl. "master – slave")

Multimedijeke arhitekture i sustavi

56



## Komunikacija s računalom

- Dvosmjerna komunikacija preko čitača Half duplex – ne istovremeno u oba smjera
- Komunikacija podatkovnim paketima
- APDU protokol (Application Protocol Data Unit)
  - Razmijena naredbi i odgovora
  - Odnos "gospodar – sluga" (engl. "master – slave")

Multimedijeke arhitekture i sustavi

57

## Komunikacijski protokoli

- APDU odgovor
  - tijelo
  - statusna riječ
  - Podatkovno polje
  - SW1
  - SW2
- Podatkovno polje
  - CLA
  - INS
  - P1
  - P2
  - Lc
  - Podatkovno polje
  - Le
- Način obrade podataka
  - Procesor prima naredbu i eventualne podatke
  - Naredba se izvodi, međurezultati se spremaju u internu memoriju
  - Eventualni povratni rezultati šalju se natrag u odgovoru na poruku

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
zaglavje	tijelo
CLA	INS

APDU naredba	
<th	

## Kripto-koprocessor

- Kripto algoritmi su izuzetno zahtjevnii te ako postoji potreba za brižom obradom pored generalnog procesora dodaje se kriptografski procesor
- Kriptografski procesor ima ponekad i cilj zaštitu generatora slučajnih brojeva (!) te nekih master-ključeva

Multimedijiske arhitekture i sustavi

61

## Primjeri

### MIFARE

- Kripto algoritmi su izuzetno zahtjevnii te ako postoji potreba za brižom obradom pored generalnog procesora dodaje se kriptografski procesor
- Kriptografski procesor ima ponekad i cilj zaštitu generatora slučajnih brojeva (!) te nekih master-ključeva

Multimedijiske arhitekture i sustavi

62

## O MIFARE-u

- ISO 14443: **tip A** i tip B
- NXP Semiconductors (Philips)
- RF komunikacijska tehnologija
- Memorije kartice
- Doseg očitana do cca 10 cm

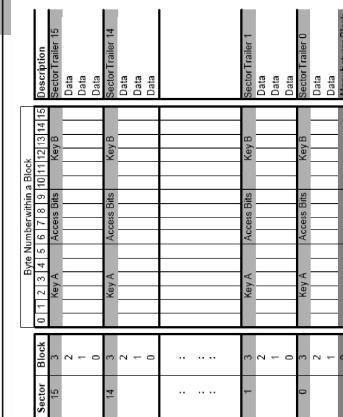


- Najraširenija tehnologija beskontaktnih kartica u svijetu
- 500 mil. kartica i oko 5 mil. čitača u upotrebi

Multimedijiske arhitekture i sustavi

64

## Organizacija podataka



Multimedijiske arhitekture i sustavi

65

## Organizacija kartice

- 16 sektora sa 4 bloka
- 0. sektor → serijski broj
- 0. sektor → MAD
- Svaki sektor zasebni par ključeva (A i B)
- Ostalih 15 sektora je rezervirano za aplikacije

Multimedijiske arhitekture i sustavi

66

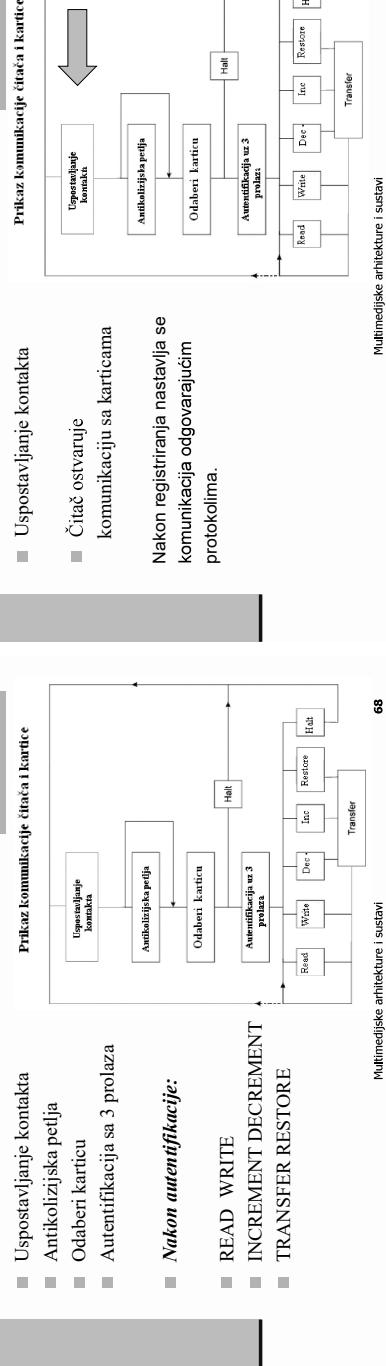
## Organizacija kartice

- MAD (direktorij) definira zajedničku strukturu podataka za sve aplikacije koje se naizade na kartici. MAD time omogućuje terminalu koji očitava karticu da u što kraćem vremenu i što učinkovitije pronađe odgovarajuću aplikaciju.
- Svaka MIFARE kartica sadrži integrirani čip sa jedinstvenom identifikacijom (UID) koja je dodijeljena još za vrijeme procesa proizvodnje.
- Serijski broj nije kriptiran i nalazi se u sektoru 0 (blok 0) i ne može se nista preko njega upisati u to područje.

Multimedijeke arhitekture i sustavi

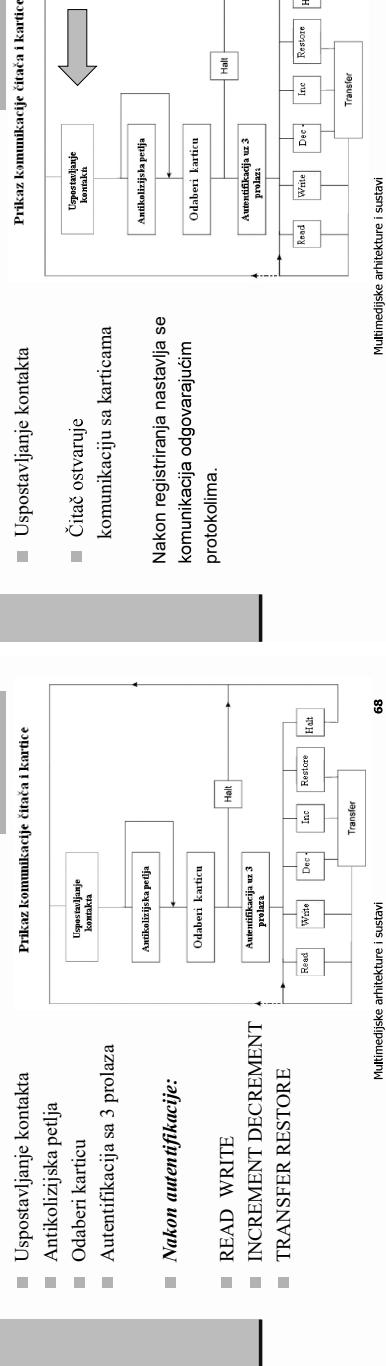
67

## Način komunikacije



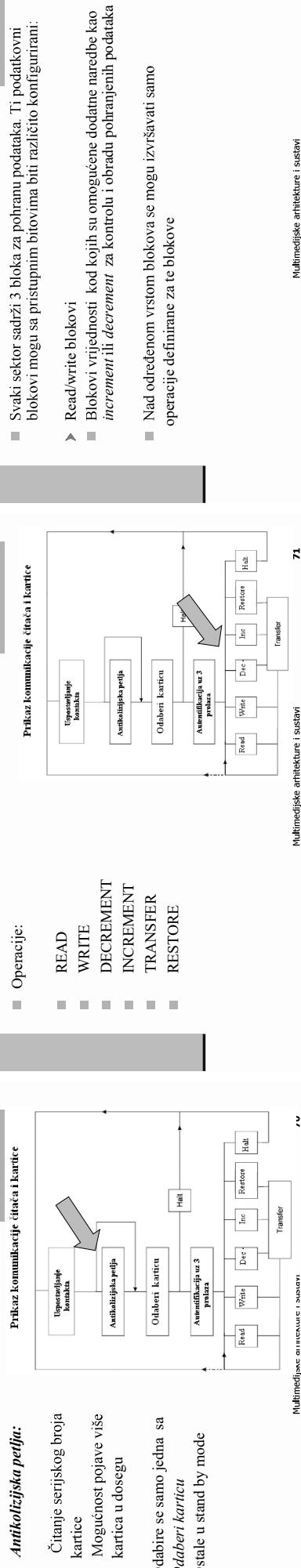
68

## Način komunikacije



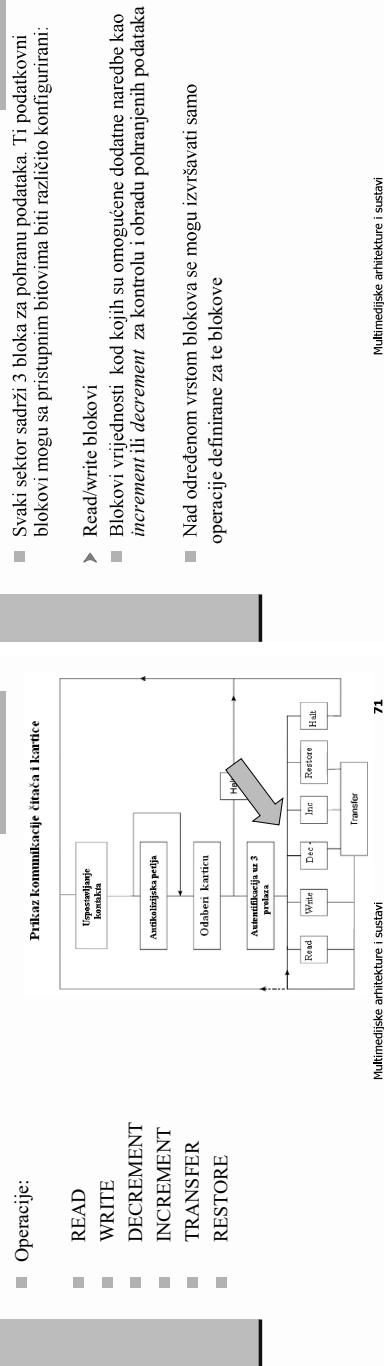
69

## Način komunikacije



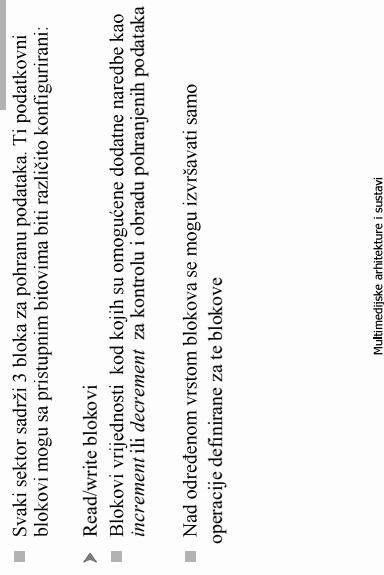
70

## Način komunikacije



71

## Podatkovni blokovi



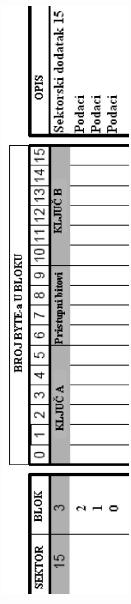
72

Multimedijeke arhitekture i sustavi

## Sektorske karakteristike

Svaki sektor ima poseban dio koji sadrži:

- Tajne ključeve A i B, koji vraćaju logičke nule pri čitanju
- Pristupne uvjetе za 4 bloka tog sektora
- Ukoliko se ne koristi jedan od ključeva, jer nema potrebe za njim, njegovi bitovi se mogu pretvoriti u podatkovne bitove.



## Funkcijske karakteristike

Uvjediti za pristup podacima:

- *Sector trailer* – poseban dio sektora u kojem se nalaze pristupni uvjeti u obliku 3 bita
- Kontroliraju razine memorijskog pristupa koristeći ključeve A i B
- Sa svakim memorijskim pristupom, unutar njega provjerava format pristupnih uvjeta.

Pristupni blokovi	Naredbe	BLOK	OPIS
C1, C2, C3,	read, write	→	3 sek. dodatak
C1, C2, C3,	read, write, increment, decrement, transfer, restore	→	2 podaci
C1, C2, C3,	read, write, increment, decrement, transfer, restore	→	1 podaci
C1, C2, C3,	read, write, increment, decrement, transfer, restore	→	0 podaci

## Pristupna konfiguracija za podatke

Table 4. Access conditions for data blocks

C1	C2	C3	Access condition for		Application
			read	write	
0	0	0	key A E1	key A E1	decrement, transfer, restore
1	0	0	key A E1	key E1	key A E1  transport configuration
1	0	1	key A E1	key E1	never never never never never never read/write block
1	1	0	key A E1	key E1	key A E1  value block
1	0	1	key A E1	key E1	never never never never never never read/write block
1	1	1	key A E1	key E1	never never never never never never read/write block
1	0	1	key B E1	key E1	never never never never never never read/write block
1	1	1	key B E1	key E1	never never never never never never read/write block

75

## Primjer:

- Korištenje MIFARE kartice za dvije jednostavne aplikacije:

- Čitanje podataka
- Elektronički novčanik

## Java Card

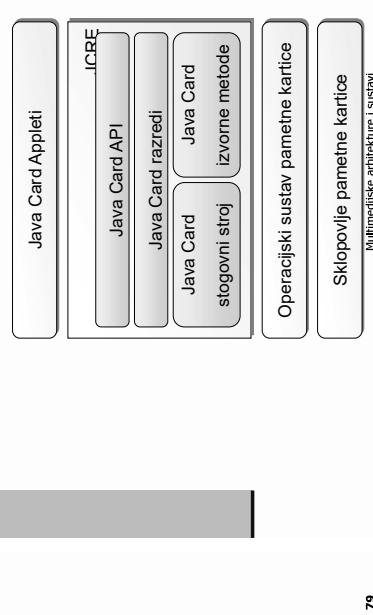
## Osobine Java Card OS-a

- temelji se na normi ISO 7816-4
- neovisnost kartične aplikacije o platformi na kojoj se izvodi - učitavanje aplikacija nakon proizvodnje kartice
- osnovne namjene operacijskog sustava:
  - upravljanje prijenosom podataka
  - kontrola izvođenja naredbi
  - izvođenje kriptografskih algoritama
  - stvaranje digitalnih potpisa

## Arhitektura Java Card tehnologije

- najmanja računalna platforma za Java
- tipična konfiguracija memorije: 1K RAM-a, 32K EEPROM-a i 24K ROM-a
- podrižan samo podskup osobina Java jezika

## Arhitektura Java Card tehnologije



Multimedijiske arhitekture i sustavi

79

## Java Card platforma

- platforma za izvršavanje Java appleta
- postojanje više aplikacija na kartici
- neovisnost i zaštićenost aplikacija
- dajevoi Java Card platforme:
  - Java Card Virtual Machine (JCVM)
  - Java Card Runtime Environment (JCREE)
  - Java Card Application Programming Interface (API)

Multimedijiske arhitekture i sustavi

81

## Svojstva Java Card jezika

- podrižan samo podskup osobina Java jezika
- očuvana objektno-orientirana svojstva Java jezika
- razlika između Java Card appleta i Java appleta
- podrižane osobine Java jezika
- primitivni tipovi podataka: byte, short, boolean, int(kod nekih platformi) Java paketi klase, iznimke jednodimenzionalna polja sučelja objektno-orientirane osobine
- nepodrižane osobine Java jezika primitivni tipovi podataka: char, long, float, double dreteve, dinamičko upravljanje memonjom, kloniranje objekata znakovi i stringovi višedimenzionalna polja upravljanje sigurnošću dinamičko učitavanje klasa

Multimedijiske arhitekture i sustavi

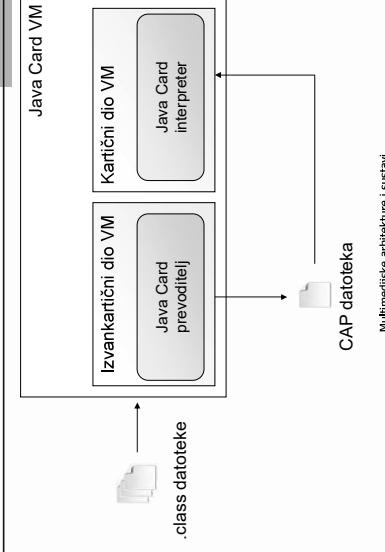
82

## Java Card stogovni stroj (2)

- podijeljen na dva dijela
- izvankartični dio (Java Card prevoditelj)
  - procesiranje i kreiranje CAP datoteke
  - inicijalizacija statičkih varijabli
  - optimiziranje Java bytecode-a
  - kreiranje podatkovnih struktura
- kartični dio (Java Card interpreter)
  - izvedba bytecode naredbi
  - upravljanje memorijom
  - kreiranje objekata

Multimedijiske arhitekture i sustavi

83



Multimedijiske arhitekture i sustavi

84

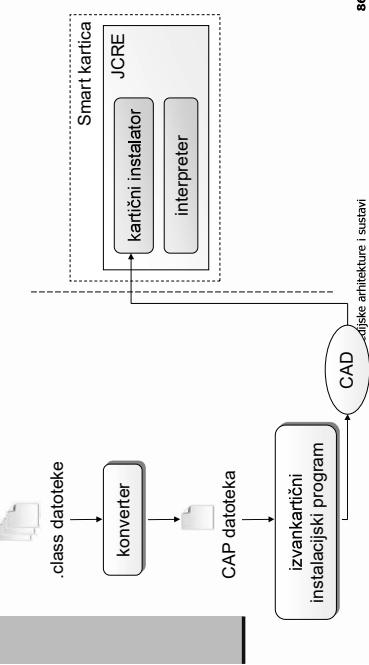
- Java Card interpreter – izvršava CAP datoteke
- instalator – mehanizmi preuzimanja i instaliranja
- CAD uređaj – prihvaca karticu
- CAP datoteka sadrži binarni kod
- podjela funkcionalnosti između interpretera i instalatora
  - smanjena veličina instalatora
  - fleksibilnost implementacije instalatora

## Java Card izvršna okolina (1)

- sistemski komponente – izvršavaju se unutar kartice
- upravljanje kartičnim resursima
- upravljanje mrežnom komunikacijom
- izvršavanje appleta
- sigurnost appleta
- najvažnija namjena izvršne okoline – uloga operacijskog sustava
- appleti se mogu učitavati na karticu nakon proizvodnje

87

## Java Card instalator (2)



86

Multimedijске arhitekture i sustavi

## Java Card izvršna okolina (1)

- sistemski komponente – izvršavaju se unutar kartice
- upravljanje kartičnim resursima
- upravljanje mrežnom komunikacijom
- izvršavanje appleta
- sigurnost appleta
- najvažnija namjena izvršne okoline – uloga operacijskog sustava
- appleti se mogu učitavati na karticu nakon proizvodnje

87

## Java Card izvršna okolina (2)

- dijelovi izvršne okoline:
  - Java Card stogovni stroj
  - Java Card aplikacijske klase
  - industrijsko-specifična proširenja
  - JCRE sistemske klase

Multimedijске arhitekture i sustavi

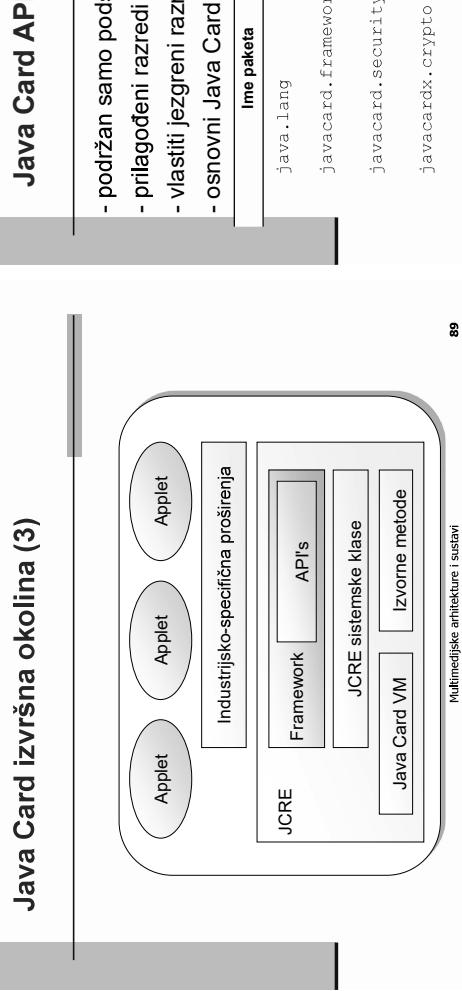
Ime paketa	Opis paketa
java.lang	osnove dizajna Java Card tehnologije
javacard.framework	osnovni skup klasa i sučelja potrebnih za izgradnju i rad s Java Card appletima.
javacard.security	razred i sučelja koja su potrebna za implementaciju sigurnosnih alata
javacardx.crypto	dodatajni paket koji sadrži sigurnosne klase koje pružaju podršku za kriptografske alate

88

Multimedijске arhitekture i sustavi

## Java Card izvršna okolina (3)

- podižan samo podskup Java paketa
- prilagođeni razredi
- vlastiti jezgreni razredi
- osnovni Java Card API (2.1.1.) paketi:



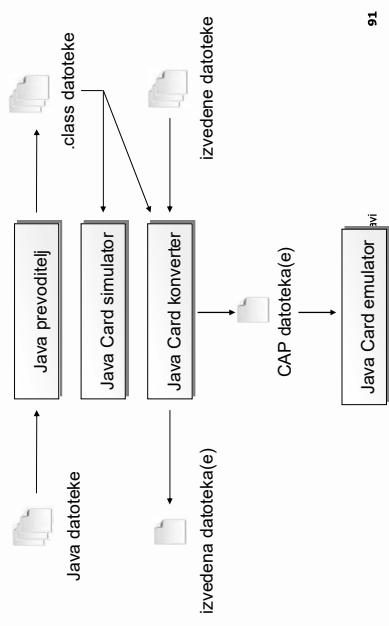
89

Multimedijске arhitekture i sustavi

## Java Card API

Multimedijске arhitekture i sustavi

## Razvojni proces appleta



91

Multimedijске arhitekture i sustavi

## Sigurnosne osobine Java Card platforme

- implementirane osobine Java Card platforme:
  - integritet transakcije
  - perzistentni i tranzientni objekti
  - aplikacijski vratovid
  - sigurnosne i kriptografske klase
  - izvorne metode u appletima

## Primjer

- Jednostavna aplikacija za izračun kripto funkcije

92

Multimedijске arhitekture i sustavi

# Multimedijiske arhitekture i sustavi

Algoritmi i njihova izvedba u rač.  
sustavima

- Do sada smo vidjeli na primjeru JPEG-a kako izvesti neki algoritam
- Jednostavno uzmem standard i raspisemo ga u nekom višem programskom jeziku
- Stvar će naravno raditi
- Brzina je upitna !!!

Prof.dr.sc. Mario Kovač  
Prof.dr.sc. Hrvoje Milinarić



Ova predavanja konste ječan do matenja koji je ustupljen od strane XILINX pod AUP uvjetima konstenja.

Multimedijiske arhitekture i sustavi

1

Algoritmi i njihova izvedba u rač.  
sustavima

- Slijedeći korak: provjeriti usko grio algoritma:
  - Kod JPEG-a
    - DCT/IDCT
    - kvantizacija
    - RGB2YUV i YUV2RGB
    - GetBits (vrlo jednostavna funkcija koja se jako puno puta poziva)

3

Multimedijiske arhitekture i sustavi

2

Algoritmi i njihova izvedba u rač.  
sustavima

- Slijedeći korak: provjeriti usko grio algoritma:
  - Kod JPEG-a
    - DCT/IDCT
    - kvantizacija
    - RGB2YUV i YUV2RGB
    - GetBits (vrlo jednostavna funkcija koja se jako puno puta poziva)

## Kako riješiti problem

### ■ RGB to YUV Conversion

- Prvi način programski:
- DCT/IDCT: da li postoji neki drugi pristup od standardnog

$$X_{k_1, k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1, n_2} \cos\left[\frac{\pi}{N_1}\left(n_1 + \frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2 + \frac{1}{2}\right)k_2\right].$$

- Pojednostavljeni algoritam putem FFT ili neki drugi pristup. AAN Algoritam
- Treba nam pomoći matematike.
- Problem znanja i primjene

Multimedijiske arhitekture i sustavi

4

### ■ RGB to YUV cijelobrojni

$$\begin{aligned} Y &= (66 * r + 129 * g + 25 * b + 128) >> 8 + 16; \\ U &= (-38 * r - 74 * g + 112 * b + 128) >> 8 + 128; \\ V &= (112 * r - 94 * g - 18 * b + 128) >> 8 + 128; \end{aligned}$$

### ■ YUV to RGB Conversion

$$\begin{aligned} B &= 1.164(Y - 16) + 2.018(U - 128) \\ G &= 1.164(Y - 16) - 0.813(V - 128) - 0.391(U - 128) \\ R &= 1.164(Y - 16) + 1.596(V - 128) \end{aligned}$$
$$\begin{aligned} Y &= (66 * r + 4 * 32 * g + 25 * b + 128) >> 8 + 16; \\ U &= (-38 * r - 74 * g + 112 * b + 128) >> 8 + 128; \\ V &= (3 * 38 * r - 3 * 32 * g - 18 * b + 128) >> 8 + 128; \end{aligned}$$

Multimedijiske arhitekture i sustavi

6

## GetBits

- Problem dohvata bitova iz memorije
- Nije zahtijevan potprogram ali se često poziva i zato može izazvati probleme.
- Nije efikasno dohvaćati bit po bit na 32-bitnom procesoru
- Problem protočne arhitekture
- Problem pričuvne memorije

Multimedijiske arhitekture i sustavi

7

## Kako rješiti problem?

- Aproksimacija funkcija polinomom n-tog reda...
- Prevođenje algoritama iz realne u cijelobrojnu domenu (fix-point)
- Dodavanje posebnog sklopovlja
  - ARM - XSCALE(200MHz) procesor prespor za dekodiranje MPEG4 video zapisa (320x240). Naknadno dodan dodatni sklop Maratton za pomoć u dekodiranju Video zapisa.
- Problem protične arhitekture
- Problem pričuvne memorije

Multimedijiske arhitekture i sustavi

8

## Kako rješiti problem?

- Ne postoji metodologija kako to jednostavnije ubrzati algoritam
- Što onda, kako ubrzati algoritam?
  - Programske
    - Aproksimacija, pojednostavljenje
    - Proučiti arhitekturu procesora
      - Napisati pretvorbu u strojnom jeziku
      - Možda postoje posebne naredbe u strojnom jeziku koje mogu ubrzati algoritam
        - HITACHI SH4 (množenje maticice 4x4 s vektorom) (97)
        - SIMD, ARM NEON, 3D naredbe
  - Sklopovsko rješenje

Multimedijiske arhitekture i sustavi

9

## Zaključak

- Pisanje efikasnog koda zahtijeva poznavanje i programske i sklopovske podrške. Moramo poznavati:
  - Kako radi prevodioci
  - Kako se koristi strojni kod
  - Arhitekturu sklopovlja periferija
  - Arhitekturu procesora

Multimedijiske arhitekture i sustavi

10

## Multimedijiske arhitekture i sustavi

- Kako ubrzati rad algoritma korištenjem sklopovlja:
- Korištenjem postojećih sklopova
  - Izrada vlastitih
  - Sklopovska integracija SoC
  - Hardware software codesign

## Hardware Software codesign

- Problematika izvedbe cjelokupnog rješenja
- Pogodnost izrade sustava u jednom okruženju
  - Jednostavniji dizajn
  - Naročito pogodan u ugradbenim sustavima

Multimedijiske arhitekture i sustavi

11

Multimedijiske arhitekture i sustavi

12

## Hardware Software codesign

- Ugradbeni sustavi imaju sljedeće karakteristike:
  - Uglavnom imaju jednu funkcionalnost
  - Koja je unaprijed definirana
  - Zadovoljavaju
  - Projektirani da se smanjiv potrošnja
    - Što manji broj komponenti
    - Moraju zadovoljiti brzinom rada
  - Rad u stvarnom vremenu (Real-time)
    - Mora kontinuirano pratiti događaje i reagirati na promjene
    - Sklopovska programska ispretezenost

Multimedijeke arhitekture i sustavi

13

## Ugradbeni računalni sustavi

- Primjeri:
  - Mobilni telefoni
- Auto aplikacije
  - Sustav kočnica, kontrola proklizavanja, zračni jastuci, it.d.
  - Avionska industrija
    - Sustav kontrole leta, kontrola motora, sustav za samostalno letenje, sustavi za zabavu putnika
  - Obrambeni sustavi
    - Radarski sustavi, sustav za kontrolu borbenih avion, sustavi avion, ...

Multimedijeke arhitekture i sustavi

14

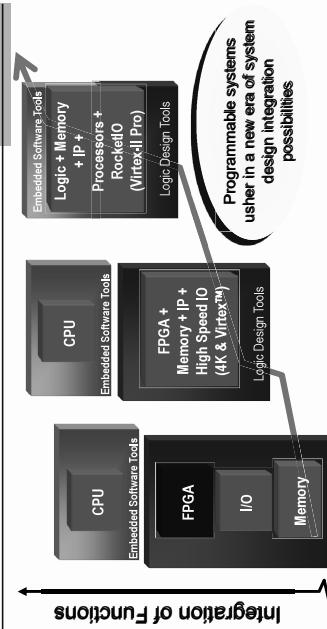
## Postojeća tehnologija

- Mikrokontroler zasnovani sustavi
- DSP procesor zasnovani sustavi
- ASIC tehnologija
- FPGA tehnologija

Multimedijeke arhitekture i sustavi

15

## Integration in System Design



Multimedijeke arhitekture i sustavi

16

## Ugradbeni sustavi korištenjem FPGA

- Osnove smo vidjeli na URSt-u
- Sastoje se od:
  - FPGA sklopovske arhitekture
  - Generiranje upravljačkih programa i biblioteka
  - Programske aplikacije
    - Potprograma
    - Prekidnog sustava
    - Operating System (OS) ili Real Time Operating System (RTOS) (optional)

## Što je u planu za ovaj dio semestra?

- Dobiti neki osnovni uvid u problematiku i metode riješavanja H/S codesigna.
- S obzirom da već od prije poznajete programabilnu logiku, a u prvom dijelu predavanja upoznali ste se s JPEG standardom. Ovaj dio će se pozabaviti problemom izrade sustava digitalnog foto aparata

Multimedijeke arhitekture i sustavi

17

Multimedijeke arhitekture i sustavi

18

## Primjer 22 Mpixel digitalni aparat

- Phase One H25
- Digitalna kamera
- 2004 godina
- 22 MegaPixel
- Cijena:
  - \$29,990



Multimedijeke arhitekture i sustavi

19

## Phase One

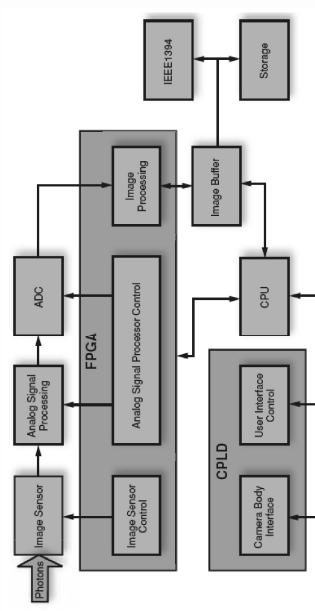
- Velika veličina datoteke: 128 MB/staci  
od 16 bits po boji, ukupno 48 bits za red/green/blue (RGB)
- Brzina prijenosa podataka 7 Gbps
- Velika memorija za dugacke burst sequences
- Pohrana nekomprimiranih ili komprimiranih slika
- 30 slike/min u punoj rezoluciji
- 400 Mb IEEE1394 (Firewire)
- Napredni "power management"
- Male dimenzije (relativno gledano)
- Danas:
  - CCD Full frame
  - Rezolucija: 80 mega pixels
  - CCD size effective 49.1 x 36.8 mm
  - Pixel size 6.8 x 6.8 micron
  - Image ratio 4:3
  - Cijena: \$49,990



Multimedijeke arhitekture i sustavi

21

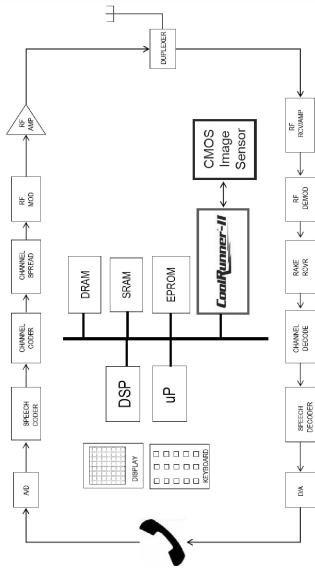
## Phase One



Multimedijeke arhitekture i sustavi

22

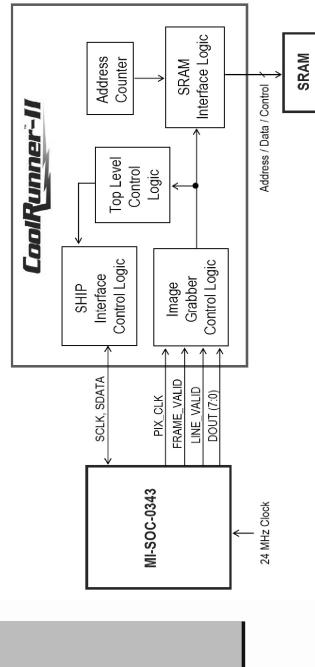
## Mobilni telefoni - primjer



Multimedijeke arhitekture i sustavi

23

## CoolRunner-II



Multimedijeke arhitekture i sustavi

24

## Prototyping

- “A prototype is an early sample or model built to test a concept or process or to act as a thing to be replicated or learned from.”
- Male serije

## Mikroprocesori protiv FPGA

- Izvršavaju prevedeni kod napisan u asembleri i/ili višem programskom jeziku
- Program se nalazi na samom mikroprocesoru i/ili vanjskoj memoriji
- Procesor dohvaca naredbe, dekodira ih, obrađuje podatke i kontrolira I/O
  - Slijedno izvođenje
  - Zahtijeva velik broj taktova za obavljanje operacija
  - Lagana optimizacija potrošnje (Power save mode)
  - Procesor proradi samo na određene događaje
  - Ograničena I/O sučelja

## Mikroprocesori

- Izvršavaju prevedeni kod napisan u asembleri i/ili višem programskom jeziku
- Program se nalazi na samom mikroprocesoru i/ili vanjskoj memoriji
- Procesor dohvaca naredbe, dekodira ih, obrađuje podatke i kontrolira I/O
  - Slijedno izvođenje
  - Zahtijeva velik broj taktova za obavljanje operacija
  - Lagana optimizacija potrošnje (Power save mode)
  - Procesor proradi samo na određene događaje
  - Ograničena I/O sučelja

## FPGA

- ‘Sea of gates’ moguće konfigurirati prema potrebi
- Pogodni za parallelno izvođenje operacija (mreža, multimedija,...)
- Velik broj I/O pinova i podržanih standarda
- Jednostavna podjela u funkcijске blokove
- Nisu pogodni za tokovne aplikacije i kontrolu

## Mikroporcessor naspram FPGA

- Zašto ne bi koristili najbolje od jedno i drugoga.

■ Povežimo ih zajedno u jednu cjelinu.

## FPGA embedded processor

- FPGA računalni sustav ima mnoge prednosti u odnosu na standardni računalni sustav:
  - 1) modularnost i nadogradnja
  - 2) jednostavna migracija
  - 3) smanjena cijena i potrošnja (redundantnost)
  - 4) sklopovsko ubrzanje

## Modularnost i Nadogradnja

- Potpuna fleksibilnost u izradi računalnog sustava.
- Dizajn može zahtijevati potpuno novu nepostojćeće sučelje koje je vrlo jednostavno dodat
  - Na primjer ne možete naci procesor sa 10 UART sučelja, ali svaki FPGA možete prilagoditi bez ikakvih problema da ima 10 UART potrova.

## Migracija

- Mnoge kompanije su razvile cijeli sustav kojem je životni vijek puno veći nego, životni vijek pojedinih komponenti
- FPGA soft-processori su izvanredan odabir jer su opisani pomoću HDL jezika i lako ih je prebaciti i na nove platforme
- Npr. Končar Elektronik

## Smanjena cijena i potrošnja

- Sklop koji je prije zahtjevalo više komponenti danas se može zamijeniti s jednim FPGA sklopotom
- Smanjivanjem broja komponenti, možemo smanjiti veličinu proizvoda i potrebnih komponenti
- Sve to dovodi do smanjenja cijene i potrošnje.

## Sklopovsko ubrzanje

- Perhaps the most compelling reason to choose an FPGA embedded processor is the ability to make tradeoffs between hardware and software to maximize efficiency and performance.
- If an algorithm is identified as a software bottleneck, a custom co-processing engine can be designed in the FPGA specifically for that algorithm.
- This co-processor can be attached to the FPGA embedded processor through special, low-latency channels, and custom instructions can be defined to exercise the co-processor.
- With modern FPGA hardware design tools, transitioning software bottlenecks from software to hardware is much easier since the software C code can be readily adapted into hardware with only minor changes to the C code.

## Nedostatci

- Za razliku od off-the-shelf procesora, sklopovska arhitektura treba biti opisana za FPGA i testirana.
- Zbog povezanosti sklopova i programske podrške alati su složeniji.
- Kako je područje relativno novo alati imaju neke dječje bolesti ☺.
- Cijena samih komponenti.
- Cijena off-the-shelf višestrukog jeftinija od FPGA sklopova.

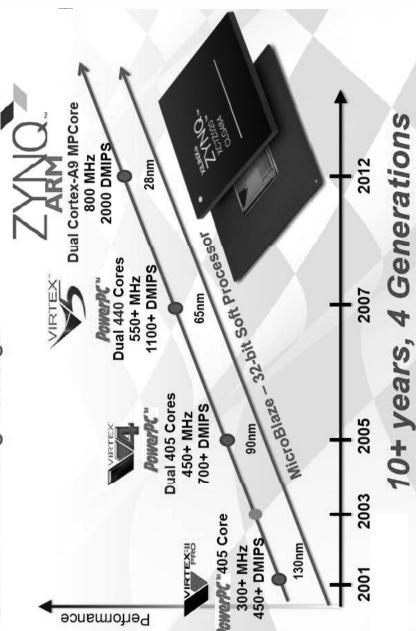
## Soft – Hard procesor

- Soft procesor – koristi FPGA blokove
- Hard procesori – Izvedeni u siliciju
- ARM922T -Altera Excalibur
- PowerPC 405 - Xilinx Virtex-II Pro and Virtex-4.
- Za razliku od hard procesora, soft procesor mora biti sintetiziran i implementiran u FPGA sklopu.

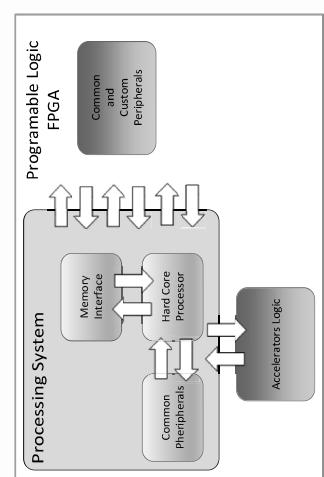
## FPGA embedded processor

- FPGA
- Soft core processor
  - PicoBlaze, MicroBlaze (Xilinx)
  - Nios II (Altera)
  - LEON3, 8051, C68000, ...
  - ARM Cortex M1
- Hardcore processor
  - PowerPC 405

## Xilinx Processing Heritage



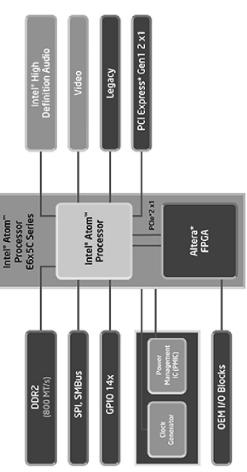
## Što imamo danas?



## What We Have Today ?

- Xilinx**
  - Virtex II Pro, Virtex 4, Virtex5 (PowerPC 405)
  - ZYNQ 7000
- Altera**
  - Excalibur (ARM922T)
  - Cyclone V SoC, Arria V SoC
- Actel**
  - SmartFusion
  - SmartFusion 2

## Intel Atom Processor E6x5C Series



## Xilinx Zynq 7000 family The All Programmable SoC

- Nije običan FPGA!**
- Nije običan mikroprocesor!**
- Jedinstvena simbioza i jednog i drugog**
  - Dual ARM Cortex™-A9 procesor+ caches + robustni sustav periferija
  - Velika povezanost između procesora i logike
- To više nije FPGA ili procesor to je procesorski sustav s programabilnim logikom – “All Programmable SoC”**

## Smanjenje troška

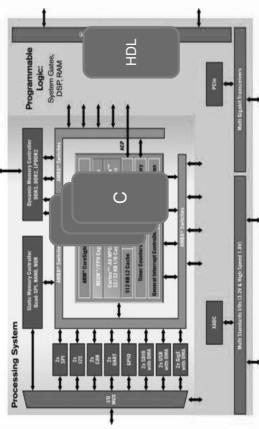
- Smanjen broj komponenti po uređaju**
  - Procesor
  - Programabilni sklop
  - DSP
  - Napajanje
  - Smanjena PCB složenost
  - Manje vodova => manje sjejava
  - Brži dizajn
  - Jako bitno
  - In-System Reconfiguration

Multi-chip Zynq-7000

## Povećanje performansi

### Meet HW and SW Processing Performance Needs

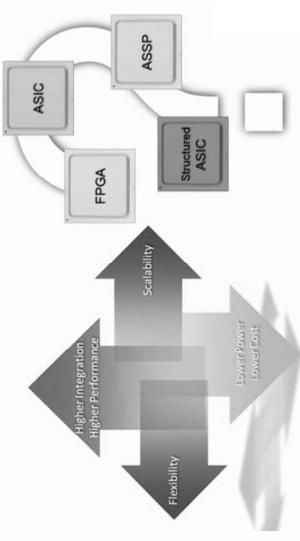
- Programabilna logika
- Ogromna DSP snaga
- Sabirnice velike propusnosti –on chip
  - Over 3000 Processing System to Programmable Logic direct connections
- High performance I/Os
- Gigabit transceivers



### Optimized & Simplified HW/SW Partitioning

## Povećanje performansi

### Koju tehnologiju izabratи?



## Današnji zahtjevi

## Što odabratи?

	ASIC	ASSP	2 Chip Solution	FPGA SoC
Performance	+	+	-	+
Power	+	+	-	+
Unit Cost	+	+	-	+
TCO	-	+	+	+
Risk	-	+	+	+
TM	-	+	+	+
Flexibility	-	-	+	+
Scalability	-	-	+	+

## Što odabratи?

	ASIC	ASSP	2 Chip Solution	FPGA SoC
Performance	+	+	-	+
Power	+	+	-	+
Unit Cost	+	+	-	+
TCO	-	+	+	+
Risk	-	+	+	+
TM	-	+	+	+
Flexibility	-	-	+	+
Scalability	-	-	+	+

## Što odabratи?

## Naš sustav

- Xilinx Zynq SoC FPGA
- SPARTAN-6 E2LP



## Komponente sustava

- Zynq SoC FPGA
  - ZedBoard - »Zynq™-7000 Soc XC7Z020
  - Dual core ARM Cortex A9
  - Spartan-6 E2LP
    - MicroBlaze procesor
- OmniVision OV7670
  - Digitalni video senzor OmniVision OV7670
  - I2C (SCCD)
  - Zoom Video Port – Digital video port

Multimedijeke arhitekture i sustavi

50

## Zynq-7000 Family Highlights

- ARM procesorski sustav
  - Application Processor Unit (APU)
  - Dual ARM Cortex™-A9
  - Prijavna memorija
  - Fully integrated memory controllers
  - I/O sučelja
- Povezano s programabilnom logikom
  - Koristi se zaporširivanje funkcionalnosti procesora
  - Skalabilnost i povećanje performansi
  - Fleksibilno I/O sučelje
  - Veliki raspon I/O standarda
  - Seniski prijenos visokih performansi
  - Analogno digitalni pretvornici

Multimedijeke arhitekture i sustavi

51

## ZYNQ 7000

- The Zynq-7000 AP SoC arhitektura sastoji se od dva dijela

### PS: Procesorskog sustava

- Dual ARM Cortex-A9 procesor
  - Mnogobrojne periferije
- PL: Programabilne logike
  - Dijele arhitekturu najnovijih Xilinx FPGA sklopova
    - Artix™-2 zasnovani uređaji: Z-7010 i Z-7020
    - Kintex™-2 zasnovani uređaji: Z-7030, Z-7045, and Z-7100

- The Zynq-7000 AP SoC arhitektura sastoji se od dva dijela

### ARM Cortex-A9 izveden u ARMv7-A arhitekturi

- ARMv7 definira ARM Instrukcijski set (ISA)
  - ARMv7-A: oznaka A znači da podržava - Memory Management Unit (MMU)
- ARMv7 ISA osim osnovnog seta naredbi podržava sljedeće naredbe
  - Thumb naredbe: 16 bits; Thumb-2 naredbe: 32 bits

- NEON: ARM's Single Instruction Multiple Data (SIMD)

Multimedijeke arhitekture i sustavi

53

## ZYNQ – dual core processor

- ARM Cortex-A9 izveden u ARMv7-A arhitekturi

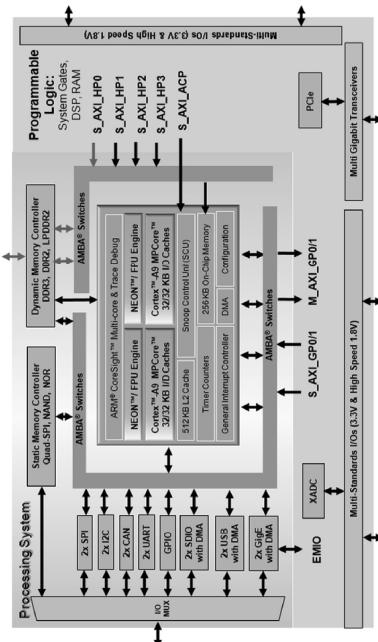
- ARMv7 definira ARM Instrukcijski set (ISA)
  - ARMv7-A: oznaka A znači da podržava - Memory Management Unit (MMU)
- ARMv7 ISA osim osnovnog seta naredbi podržava sljedeće naredbe
  - Thumb naredbe: 16 bits; Thumb-2 naredbe: 32 bits

- NEON: ARM's Single Instruction Multiple Data (SIMD)

Multimedijeke arhitekture i sustavi

54

## ZYNQ 7000



Multimedijeke arhitekture i sustavi

51

ZYNQ - dual core processor

- ARM Advanced Microcontroller Bus Architecture (AMBA®) sabirnica
    - AXI3: ARM sučelle treće generacije
    - AXI4: nadogradnja AXI3 sabirnice
  - Cortex je najnovija porodica procesora ARM

Multimedejske arhitekture i sustavi

56 Multimedijiške arhitekture i sustavi

ZYNQ - dual core processor

- 2.5 DMIPS (Dhrystone)
  - Harvardska arhitektura
  - 32KB L1 instrukcijske i podatkovne pričuvne memorije
  - 512KB L2 pričuvne memorije maksimalna frekvencija 1GHz

55

Multimedijске arhitekture i sustavi

57

ZYNQ Procesor - memorija

- On-chip memorija(OCM)
  - RAM
  - Boot ROM
  - DDRx sučelje
  - podizava LPDDR2, DDR2, DDR3
  - Flash/static, memorjsko sučelje
  - Podržava SRAM, QSPI, NAND/NOR FLASH

Multimedijiške arhitekture i sustavi

59

ZYNQ – Procesor – IO sučelja

- ## ■ NAND, NOR/SRAM, Quad SPI

Multimedijiške arhitekture i sustavi

59

60

Multimedijске arhitekture i sustavi

## PS – PL Sučelje

- AXI high-performance slave ports (HPO-HP3)
  - 32-bit i 64-bit
  - pristup OCM i DDR
  - AXI FIFO sučelje (AFI)
  - ■ AXI general-purpose ports (GP0-GP1)
    - Dva "master" PS - PL
    - Dva "slave" PL - PS
    - 32-bit širine

Multimedijeke arhitekture i sustavi

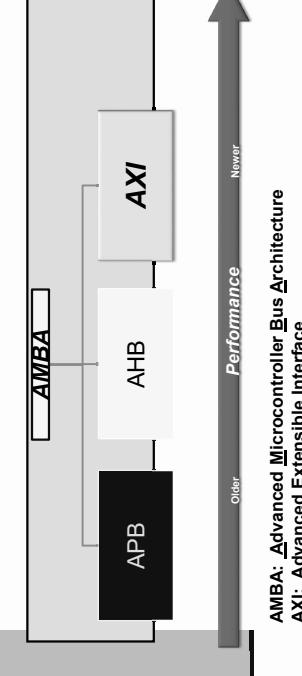
61

## AXI

- 64-bit accelerator coherence port (ACP) AXI slave interface to CPU memory
- DMA, interrupt, events signals
- Extended multiplexed I/O (EMIO) – omogućava spajanje procesorskih periferija s PL logikom
- Clock i reset
- Četiri odvojena signala vremenskog vodenja
- Četiri odvojena reset signala

Multimedijeke arhitekture i sustavi

62



Multimedijeke arhitekture i sustavi

63

## AXI

Interface	AMBA	Enhancements for FPGAs	
Memory Map (AXI)	APB AHB AXI ATB	AMBA 3.0 (2003)	Similar to PLB/46, PCI
Streaming (AXI-Stream)	AXI-4 Memory Map	AXI-4 Lite (2010)	Local Link / DSP Interfaces / FIFO / FSL
Lite (AXI-Lite)	Traditional Address/Data – No Burst (single address, single data)	PLB/46-single OPB	276 GMAC operacija u sekundi ADC – 2 x 12 bita – 17 ulaza

Multimedijeke arhitekture i sustavi

64

## Programibilna logika

- Artx-7 FPGA
- 87K logičkih blokova, oko 1.3M logičkih vrata
- 53.200 logičkih vrata
- 106.400 registara (bistabila)
- 560 KB BRAM memorija
- 220 MACC blokova
- 276 GMAC operacija u sekundi
- ADC – 2 x 12 bita – 17 ulaza

Multimedijeke arhitekture i sustavi

65

## Spartan 6

- Spartan-6:
  - Spartan-6 LX FPGA: velik broj logičkih vrata
  - Spartan-6 LXT FPGA: komunikacijski zahajevni sklopovi
  - Namijenjeni za izradu jeftinjih ugradbenih sustava
  - Velik broj različiti blokovi koji imaju različite funkcionalnosti
  - Velik broj IO podržanih normi
  - Jeftina izvedba pakiranja

## Spartan 6

- Različite naponske razine, podrška za različite standarde - SelectIO™
- Do 1.080 Mb/s podataka
- Mogućnost konfiguracije izlazne struje, do 24 mA po pinu
- 3.3V do 1.2V I/O
- "Hot swap" podrška
- High-speed GTP serijska komunikacija - LXT obitelj do 3.2 Gb/s
- High-speed podrška za: Serial ATA, Aurora, 1G Ethernet, PCI Express, OBSAI, CPRI, EPON, GPON, DisplayPort,

## Spartan 6

- DSP48A1 Blokovi
  - Sklopovi pogodni za aritmetičke operacije i digitalnu obradu signala
  - 18 x 18 Množenje i 48-bit MAC operacije
  - Pogodni za protočni ili kaskadni
  - Dodatno pred zbrajalo
  - Memory Controller blokovi
    - DDR, DDR2, DDR3 i LPDDR podrška
    - Brzine do 800 Mb/s
    - "Multi-port" sabirnička struktura sa neovisnim FIFO spremnicima

## Spartan 6

- CLB
  - Sadrže dvije polovice (Slices)
  - Slices
    - Postoje tri tipa
      - SLICEM
      - SLICEI
      - SLICEX
    - Upravljanje signalom vremenskog vođenja
      - DCM
      - PLL
    - Blok RAM
    - Množila, IO blokovi, memoriski skloppovi

## SPARTAN<sup>6</sup>

### SPARTAN<sup>6</sup>

Spartan-6 LX FPGAs Optimized for Lowest Cost Logic, DSP, and Memory (1.2 Vol. 0.8W)									
Part Number	XCS544	XCS640	XCS616	XCS620	XCS615	XCS610	XCS6100	XCS6105	XCS6107
Logic Resources	3,840	4,112	4,479	5,014	5,453	5,936	6,522	7,062	7,622
QCAPIs	4,930	5,140	5,324	5,728	6,107	6,522	7,028	7,528	8,022
Memory Demands	73	90	138	210	401	602	748	10,281	14,443
Memory Resources	12	32	32	92	116	172	268	92	401
Clock Resources	216	251	251	456	3,098	4,924	4,425	866	2,038
I/O Resources	2	2	2	4	6	6	2	4	6
Multifunction Pins	120	200	202	266	394	430	470	290	520
Configurable Switches	8	16	32	38	59	102	160	98	189
Hard IP Resources	-	-	-	-	-	-	1	1	1
Memory Controller Blocks	0	2	2	1	4	4	4	2	4
Configurable Memory Blocks	-	-	-	-	-	-	2	4	6
Serial Circuits	4	4	4	4	4	4	4	4	4
Configuration	27	37	44	77	148	171	240	44	101

### SPARTAN<sup>6</sup>

Spartan-6 LX FPGAs Optimized for Lowest Cost Logic, DSP, and Memory (1.2 Vol. 0.8W)									
Part Number	XCS614	XCS619	XCS615	XCS620	XCS610	XCS6100	XCS6105	XCS6107	XCS6108
Logic Resources	3,840	4,112	4,479	5,014	5,453	5,936	6,522	7,062	7,622
QCAPIs	4,930	5,140	5,324	5,728	6,107	6,522	7,028	7,528	8,022
Memory Demands	73	90	138	210	401	602	748	10,281	14,443
Memory Resources	12	32	32	92	116	172	268	92	401
Clock Resources	216	251	251	456	3,098	4,924	4,425	866	2,038
I/O Resources	2	2	2	4	6	6	2	4	6
Multifunction Pins	120	200	202	266	394	430	470	290	520
Configurable Switches	8	16	32	38	59	102	160	98	189
Hard IP Resources	-	-	-	-	-	-	1	1	1
Memory Controller Blocks	0	2	2	1	4	4	4	2	4
Serial Circuits	-	-	-	-	-	-	2	4	6
Configurable Memory Blocks	-	-	-	-	-	-	1	1	1
Serial Circuits	4	4	4	4	4	4	4	4	4
Configuration	27	37	44	77	148	171	240	44	101

### MicroBlaze Procesor

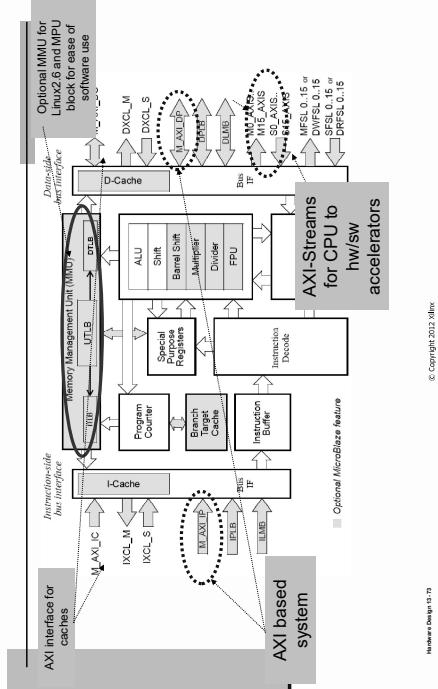
32-bitna prilagodljiva Jezgra									
Postrojena aritmética	3 stupnjeva (smajena koltčina logike) ili 5 stupnjeva (veća brzina rada)								
Pričuvana memorija za naredbe i podatke – AXI ili XCL sučelje	■ Direktno mapiranje 1-way asociativno								
■ Po izboru "Memory Mgt" ili "Memory Protection Unit"	■ Potrebno za Linux OS (Linux 2.6 podržan)								
■ Floating-point unit (FPU)	■ IEEE 754 format								
Barrel Shifter	■ Sklopovsko množilo								
■ 32x32 množilo koje daje 64 bitni rezultat	■ Sklopovsko dijeljenje								
■ AXI4 Stream/Lite/Full podrška za direktni pristup programabilnom sklopu	■ Podrška za debagiranje								

© Copyright 2012 Xilinx

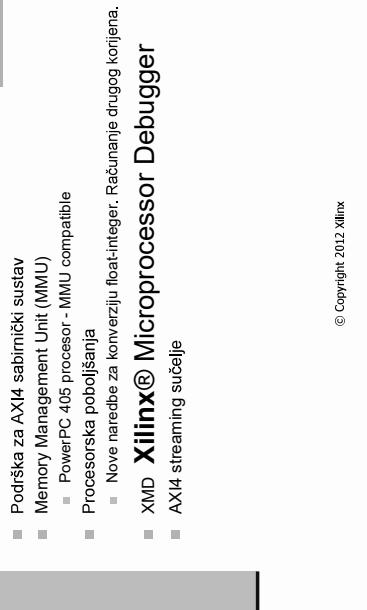
Hardware Design 13-72

## MicroBlaze Block Diagram

### AXI4 System



## MicroBlaze dodatne funkcije



## AXI Streaming Interface

- Jednostranska komunikacija, "point-to-point", koristi FIFO spremnike
- Programabilna dubina FIFO spremnika
- Direktna veza s jezgrom procesora
  - Do 16 komunikacijskih kanala
  - Specializirani regitri za čitanje i pisanje u/z FIFO spremnika

© Copyright 2012 Xilinx

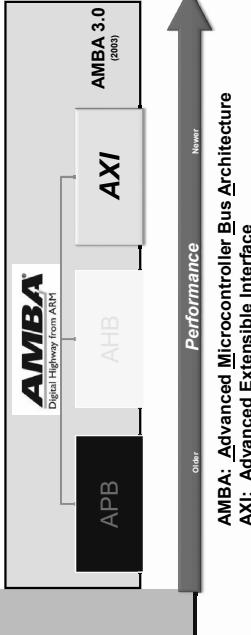
Hardware Design 13-76

## Podrška za više jezgri

- Multicore arhitektura
  - Mailbox: komunikacija između dviju jezgri
  - Podrška AXI4-Lite, AXI4-Stream and FSL
  - Mutex core: Synchronizing dva ili više procesora
  - Supports AXI4-Lite and PLBV46
  - Processor Version Register (PVR)
    - Sadrži: Processor ID, Configuration/User/Processor info (e.g., cache size etc.), version number and other internal information

© Copyright 2012 Xilinx

## AXI is Part of ARM's AMBA

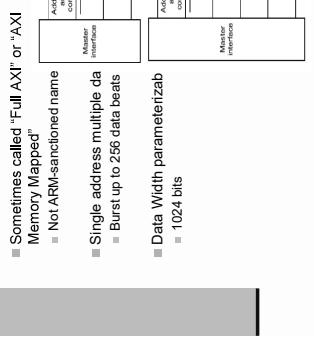
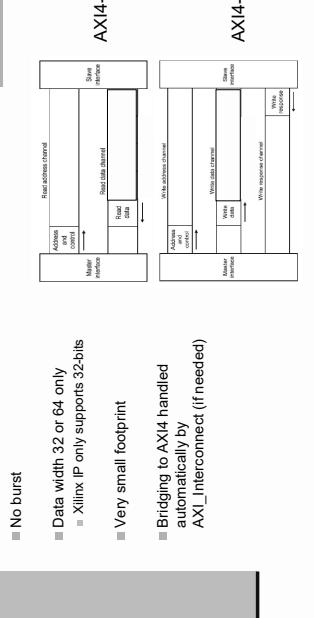
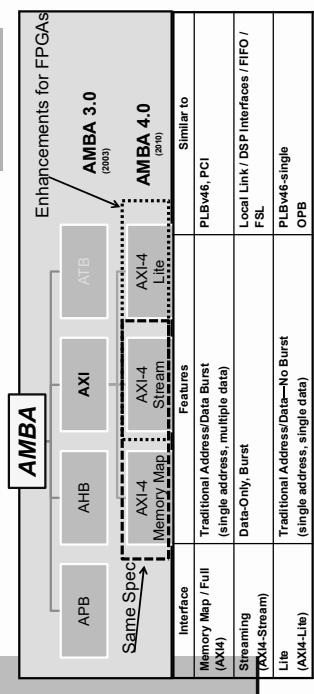


© Copyright 2012 Xilinx

Hardware Design 13-78

## AXI is Part of AMBA: Advanced Microcontroller Bus Architecture

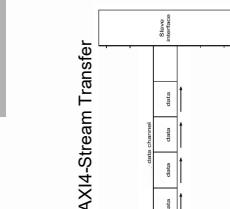
### The AXI Interface—AXI4-Lite



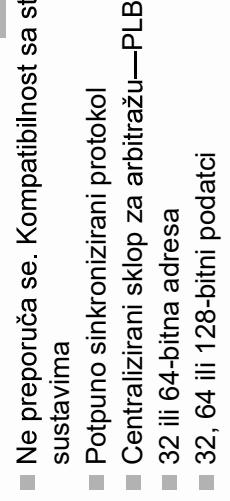
### The AXI Interface—AXI4

## The AXI Interface—AXI4-Stream

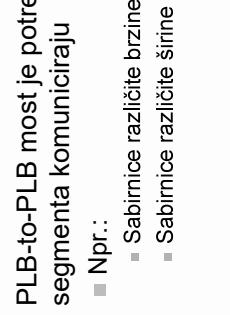
- No address channel; no read and write, always just master to slave
- Effectively an AXI4 “write data” channel
- Unlimited burst length
- AXI4 max 256
- AXI4-Lite does not burst
- Virtually same signaling as AXI Data Channels
- Protocol allows merging, packing, width conversion
- Supports sparse continuous, aligned, unaligned streams



- Ne preporuča se. Kompatibilnost sa stariim sustavima
- Potpuno sinkronizirani protokol
- Centralizirani sklop za arbitražu—PLB arbiter
- 32 ili 64-bitna adresa
- Mogućnost dijeljenja sabrmice ili point-to-point komunikacija
- Protočna struktura (2 nivoa)



- PLB-to-PLB most je potreban kada dva PLB segmenta komuniciraju
- Npr.:
- Sabrmice različite brzine
- Sabrmice različite širine



## AXI Most

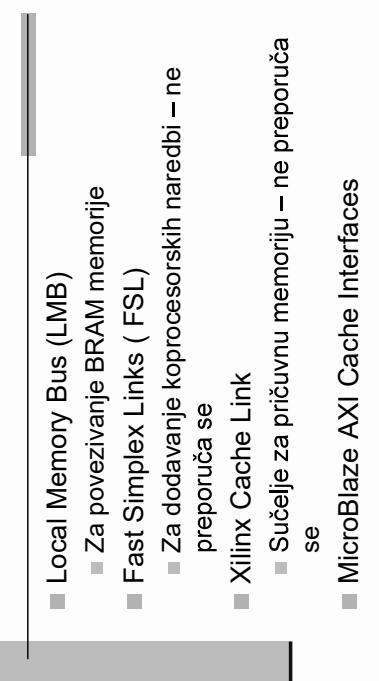
- AXI\_to\_PLBv46 | PLBv46\_to\_AXI Most
  - Used in system having two standards (Core-Connect and AMBA)
  - Supports multi-master/multi-slave connections
  - Designed to support existing customer PLBv46-based cores in an AXI system
  - AXI\_to\_APB Bridge
    - Designed to support 3<sup>rd</sup> party slave IP talking to an AXI4-Lite master
    - The bridge is slave on the AXI4Lite side and master on the APB peripheral side
    - APB3/APB4 peripherals are supported

© Copyright 2012 Xilinx

Hardware Design 13-65

## MicroBlaze Cache to External Memory Datapath

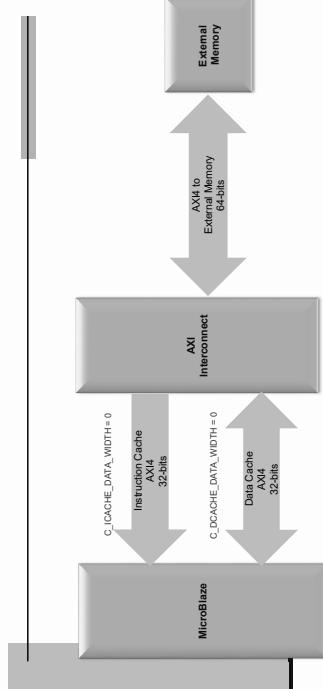
*Default Configuration, same as pre-AXI*



© Copyright 2012 Xilinx

## MicroBlaze Cache to External Memory Datapath

*Default Configuration, same as pre-AXI*



© Copyright 2012 Xilinx

# Multimedijiske arhitekture i sustavi

Prof.dr.sc. Mario Kovač  
Izv.prof.dr.sc. Hrvoje Mlinarić



Ova predavanja koniste jedan dio materijala koji je dostupan na stranici XILINX pod AUP uvjetima konstenca.

## Video Senzor

- Dvije vrste - tehnološki različite
  - CCD (charge coupled device)
  - CMOS (complementary metal oxide semiconductor)
- Svaka tehnologija ima svoje prednosti i mane u ovisnosti o primjeni. Niti jedan nema superiornu prednost pred drugim iako često možete naći da proizvođači tvrde suprotno.

Multimedijiske arhitekture i sustavi

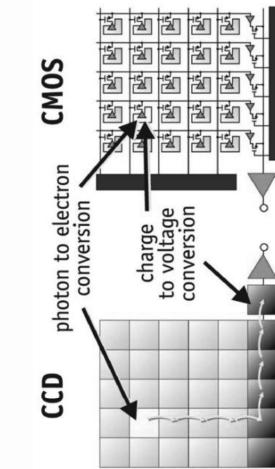
1

## Video Senzor

- CCD senzor
  - nabori svakog piksela prenosi se vrlo kratkim putovima i kroz mali broj čvorova prije nego se pretvoriti u naponsku razinu i pošalje izvan čipa kao analogni signal (koristi se samo jedan ili mali broj pretvarača). Svi pikseli dohvataju se istovremeno i uniformno se obrađuju (osnovni uvjet kvalitetne slike).

Multimedijiske arhitekture i sustavi

3

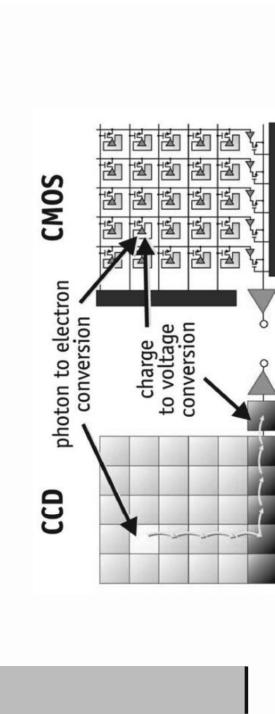


## Video Senzor

- CMOS senzor:
  - Svaki piksel ima svoj zasebni sklop za pretvorbu naboja u napon. Senzori obično posjeduju i pojačala, filtre šuma, i digitalni sklop sve na jednom čipu.
  - Sve navedeno povećava fleksibilnost dizajna.
  - Pošto svaki piksel ima svoj pretvarač narušava se uniformnost konverzije, ali zato čip može biti izrađen da treba mali broj vanjskih komponenti.

Multimedijiske arhitekture i sustavi

4



## Video Senzor

- CCD photon to electron conversion
- charge to voltage conversion
- CMOS

Multimedijiske arhitekture i sustavi

6

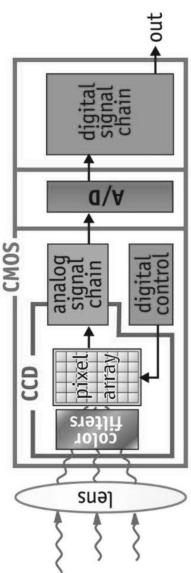
## Video Senzor

- CCD i CMOS izumljeni su kasnih '60 i početkom '70 (osnivač DALSA Dr. Savvas Chamberlain).
- Zbog svoje strukture i jednostavnije izvedbe (čitaj cijene ☺) CCD senzori su postali dominantni.
- Napretkom tehnologije '90 godina ponovo se pojavljuje interes za CMOS senzorima

Multimedijeke arhitekture i sustavi

7

## Video Senzor



Multimedijeke arhitekture i sustavi

8

## Video Senzor

- OmniVision's OV7670 SINGLE-CHIP CMOS VGA COLOR DIGITAL CAMERA



Multimedijeke arhitekture i sustavi

9

## OV7670

- 640x480 – VGA format
- 24 pina
- Podržan format – YUV 4:2:2, GRB 4:2:2, RGB Raw Data 565/555
- 8 video data: ITU-601, ITU-656, ZV port
- Automatska eksponicija/gain/kontrola bijele boje (WB)
- Operacije nad slikom – svjetloća, kontrast, gamma, saturation, oština, uzimanje dijela slike, i.t.d.
- Vanjska i unutarnja sinkronizacija

Multimedijeke arhitekture i sustavi

10

## OV7670

- Frame exposure/line exposure option (za foto aparate)
- 3.3V Volt operation, low power dissipation
- < 80 mW active power
- < 20 uA in power-save mode
- I2C kontrolno sučelje (400 kb/s):
  - Color saturation, brightness, contrast, white balance, exposure time, gain

Multimedijeke arhitekture i sustavi

11

## OV7670

- Postoji i crno-bijela verzija OV7171
- CMOS Video senzor
- Ukupan broj pikslea 656 x 488
- Maksimalno do 60 slika po sekundi

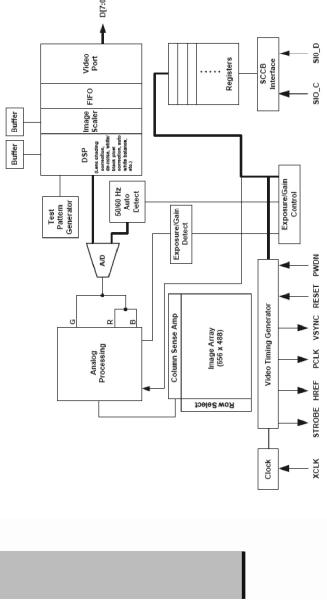
Multimedijeke arhitekture i sustavi

12

OV7670

- Predviđena za sljedeće funkcije:
    - Video konferencije
    - Video telefonija
    - Video pošta
    - Foto aparati na mobilnim telefonima
    - PC Multimedia
    - i.t.d.

0V7670

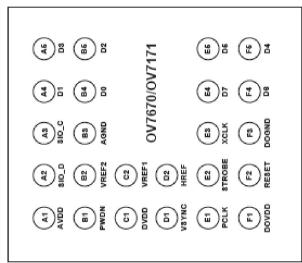


Multimedijске arhitekture i sustavi

Multimedejske arhitekture i sustavi

14

0V7670



Multimedijске arhitekture i sustavi

15

OmniVision

- **Image Sensors**
  - 16-megapixel
  - 14-megapixel
  - 13-megapixel
  - 12-megapixel
  - 10-megapixel
  - 9-megapixel
  - 8-megapixel

0V7670

- Dva komunikacijska sučelja
  - Kontrolno sučelje SCCB
  - Podatkovno sučelje: ITU-601, ITU-656, ZV port

SCCB - Serial Camera Control Bus

- Omogućava komunikaciju jednog master i više slave uređaja
  - Sukladno s IIC (I2C) sučeljem

Multimedejske arhitekture i sustavi

Multimedijiske arhitekture i sustavi

17

18 Multimedijске arhitekture i sustavi

## I2C

- I2C - Inter-Integrated Circuit
- Početkom '80ih Philips Semiconductors razvio je dvosmjernu "2-wire" komunikacijsku sabircnicu.
- Osnovna namjena joje bila da omogući jednostavnu komunikaciju između procesora i ostalih komponenti unutar televizora.
- Philips Labs u Eindhoven (Nizozemskoj)
- Danas I2C široko primijenjen i u drugim uređajima.

Multimedijeke arhitekture i sustavi

19

## I2C

- Generalno je prihvaćena kao standard
- Podržana od mnogih proizvođača:
  - Xicor
  - ST Microelectronics
  - Infineon Technologies
  - Intel
  - Texas Instruments
  - Maxim
  - Atmel
  - Analog Devices
  - i.t.d.

Multimedijeke arhitekture i sustavi

20

## I2C

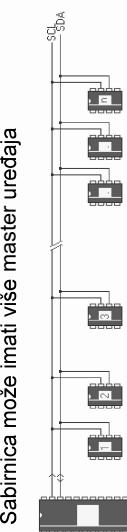
- Fizički je izvedena sa dvije žice (spojna put)■ SDA – Serial DAta
- SCL – Serial CLock
- Obije linije su dvosmjerne
- Svaki uređaj spojen na sabircnicu mora imati svoju jedinstvenu adresu na toj sabircnici
- Svaki od uređaja može slati ili primati podatke

Multimedijeke arhitekture i sustavi

21

## I2C

- Spojeno I sabirnica
- Sabirnica je slobodna kada su SDA i SCL u visokom.
- Koriste se pull-up otpornici



Multimedijeke arhitekture i sustavi

22

## I2C

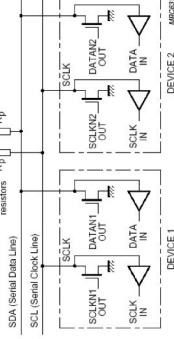
- Brzina komunikacije
  - 100 kbps (standardni način rada)
  - 400 kbps(brzi način rada)
  - 3.4 Mbps (vrlo-brzi načina rada)
- Broj uređaja na sabircnici je limitiran s maksimalnim kapacitetom od 400 pF

Multimedijeke arhitekture i sustavi

23

## I2C

- Svojstva sabircnice
  - Više uređaja može započeti komunikaciju
  - Uredaji koji započinje komunikaciju nazivaju se master na sabircnici
  - Suktadno tome svj ostali u tom trenutku su slave uređaji
  - Master na sabircnici je obično procesor
  - Sabirnica može imati više master uređaja

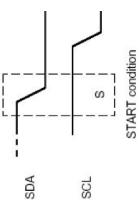


Multimedijeke arhitekture i sustavi

24

## I2C

- Komunikacija
  - Master generira START stanje
    - Dobjavljuje ostalim na sabirnici da zahtijeva pozornost
    - Postavlja SDA signal u nisko, a zatim SCL signal u nisko

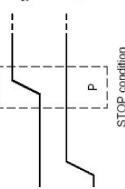


Multimedijeke arhitekture i sustavi

25

## I2C

- Nakon što master primi potvrdu može početi slati podatke.
- Na kraju master šalje STOP
  - Otpušta SCL i SDA signal koji idu u visoko stanje



Multimedijeke arhitekture i sustavi

26

## I2C

- Master
  - Kontrolira SCL
  - Šalje start i stop sekvencu
  - Kontrolira adresu na sabirnici
- Slave
  - Adresiran je od strane master-a
  - Ako master čita onda šalje bitove

Multimedijeke arhitekture i sustavi

27

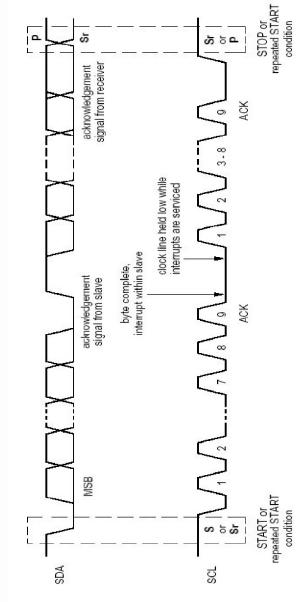
## I2C

- Prijenos podataka
  - Master spušta SCL u nisko i počinje generirati impulse za svaki bit
  - Generira se 8 pulsova za podatak i jedan puls za potvrdu od slave uređaja
  - Master započinje slanje slijedećeg bajta
  - Slave može odgoditi slanje slijedećeg ili primanje bajta držanjem signala SCL u niskom

Multimedijeke arhitekture i sustavi

28

## I2C



Multimedijeke arhitekture i sustavi

29

Multimedijeke arhitekture i sustavi

30

## I2C

- Prijenos podataka
  - Prenosi se osam bitova i potvrda od primatelja
  - Uredaj koji šalje podatke nakon 8 pulsova osloboda SDA
  - Onaj koji je primao podatke spušta SDA u nisko da bi potvrdio primitak podataka
  - Nakon toga opušta SDA signal

Multimedijeke arhitekture i sustavi

31

## I2C

- Multi master sabirnica
  - Više uređaja može kontrolirati sabirnicu
  - Moguća situacija je da više mastera istovremeno pokrenu start sekvencu
  - Potrebna je sinkronizacija na SCL signali
  - Potrebna je arbitraža na SDA signali
  - Problem se rješava korištenjem ožičenog I povezivanja

Multimedijeke arhitekture i sustavi

32

## I2C

- Sinkronizacija na SCL signalu
  - Započinje komunikacija
  - SCL je u 1 i prvo što se radi master postavlja SCL u 0 – START
  - Nakon toga Master otpušta SCL
    - Ako je SCL = 0 netko je još na sabirnici odi u stanje čekanja
    - Ako je SCL = 1 sabirnica je tvola
  - Master vraća SCL u 0
- Master vraca SCL u 0

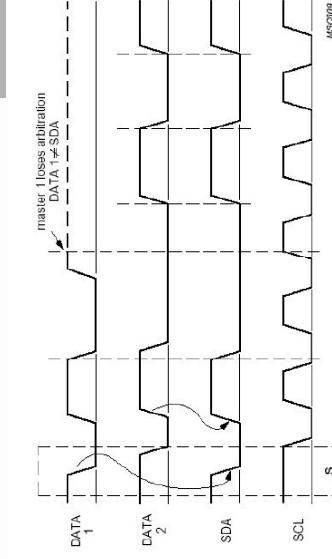
Multimedijeke arhitekture i sustavi

36

- Arbitraža na SDA signalu
  - Započinje komunikacija šalje se START
  - Izvrši se sinkronizacija. SCL signala i na visoku razinu SCL postavlja se podatak
  - Svaki master generira svoj podatak
  - Master prestaje slati ako razina na SDA signalu ne odgovara onome što je on postavio
  - Oslobađa SDA i pokušava ponovo poslati podatak kada je sabirnica slobodna

Multimedijeke arhitekture i sustavi

34



## I2C

- Adresiranje:
  - Prvi bajt uvijek šalje master
  - 7 bitova čine adresu
  - 1 bit daljnji smjer komunikacije
    - 0 – master piše podatke
    - 1 – master čita podatke
  - Prijenos podataka završava STOP stanjem, može prenijeti više od jednog bajta
  - Adresa se sastoji od fiksнog dijela i promjenjivog
  - Fiksni dio određuje I2C odbor za dodjelu adresa

Multimedijeke arhitekture i sustavi

35

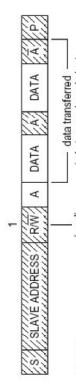
# I2C

## Okvir komunikacije

master-transmitter



master-receiver (since second byte)



Multimedijске arhitekture i sustavi

37

## Rezervirane adrese za posebnu namjenu:

### Ima 7 bitnu adresu

CS[2:0]	000	001	010	011	100	101	110	111
WRITE ID (hex)	C0	C4	C8	CC	D0	D4	D8	DC
READ ID (hex)	C1	C5	C9	CD	D1	D5	D9	DD
	*	*	*	*	*	*	*	*

Multimedijске arhitekture i sustavi

38

# SCCB

## Podržava brzinu prijenosa do 400 kbps

### Ima 7 bitnu adresu

CS[2:0]	000	001	010	011	100	101	110	111
WRITE ID (hex)	C0	C4	C8	CC	D0	D4	D8	DC
READ ID (hex)	C1	C5	C9	CD	D1	D5	D9	DD
	*	*	*	*	*	*	*	*

Multimedijске arhitekture i sustavi

39

# SCCB

## Nama bitni registri unutar senzora:

Register	Address	Default	Description
CLKRC	0x11	0x80	Bit[6]: 0: Apply prescaler on input clock 1: Use external clock directly Bit[0-5]: Clock prescaler Finternal clock = $F_{input} \cdot 2^{(Bit[0-5] + 1)}$ Range [0x000] to [1111]
DBLV	0x6B	0x0A	Bit[7]: PLL control 0: Bypass PLL 1: Input clock x4 10: Input clock x16 11: Input clock x8 Regulator control 0: Enable internal regulator 1: Bypass internal regulator

Multimedijске arhitekture i sustavi

42

# SCCB

- Čitanje podataka po I2C protokolu ne omogućava pod adresiranje pojedinih registara unutar video senzora.

- Zato se čitanje vrši sa zadnje adresiranog registra u procesu pisanja

- Moramo koristiti jedan prazan ciklus pisanja u kojem ne upisujemo podatak u registar, odnosno šaljemo samo dva bajta – adresu senzora i adresu registra

# SCCB

- Osim osnovnog adresiranje preko kojeg se odabire senzor s kojim se komunicira SCCB podržava i pod adresiranje internih registara unutar video senzora

- Tijekom slanja podataka nakon adresе senzora u drugom bajtu se šalje adresa registra kojem se želi pristupiti

- Treći bajt koji se šalje zapisuje se u odabrani registar
- U koliko se nastavi slati bajtove podatci se zapisuju u sljedeće registre koji slijede

Multimedijске arhitekture i sustavi

40

Multimedijске arhitekture i sustavi

41

## OV7670

### Video sučelje

#### ■ Nama bitni registri unutar senzora:

Register Address	Default	Description
COM3	0x0C	Bit[0]: 1: Swap the data MSB and LSB Bit[5]: On powerdown 0: Invert the output clock 1: Do not invert the output clock On powerdown Bit[4]: 0: Tristate the output data 1: Do not tristate the output data  Bit[3]: 0: Disable scaling 1: Enable scaling Bit[2]: 0: Disable downsampling, cropping, windowing 1: Enable downsampling, cropping, windowing  Bit[7]: 0: Nothing 1: Reset all the registers to default values  Bit[5]: & Nothing 1: Use QIF format  Bit[4]: & Nothing 1: Use QVGA format  Bit[3]: & Nothing 1: Use QCIF format  Bit[1]: 0: Disable color bar 1: Enable color bar Bit[2, 0]: 00: YUV 01: RGB 10: Bayer raw 11: Processed bayer raw
COM7	0x12	

43

#### ■ Video senzor podržava

- Digitalno sučelje
  - 8 bita
  - Video format CCIR601, CCIR656, ZV port
  - Format podataka – YUV 4:2:2, RGB 4:2:2, RGB raw dana 565/555

45

Multimedije arhitekture i sustavi

Register Address	Default	Description
COM3	0x0C	Bit[0]: 0: Nothing 1: Swap the data MSB and LSB Bit[5]: On powerdown 0: Invert the output clock 1: Do not invert the output clock On powerdown Bit[4]: 0: Tristate the output data 1: Do not tristate the output data  Bit[3]: 0: Disable scaling 1: Enable scaling Bit[2]: 0: Disable downsampling, cropping, windowing 1: Enable downsampling, cropping, windowing  Bit[7]: 0: Nothing 1: Reset all the registers to default values  Bit[5]: & Nothing 1: Use QIF format  Bit[4]: & Nothing 1: Use QVGA format  Bit[3]: & Nothing 1: Use QCIF format  Bit[1]: 0: Disable color bar 1: Enable color bar Bit[2, 0]: 00: YUV 01: RGB 10: Bayer raw 11: Processed bayer raw
COM7	0x12	

43

Multimedije arhitekture i sustavi

## ITU656

- Pravi naziv BT.656 predložen od strane ITU (International Telecommunication Union) stoga se cesto naziva ITU656 ili CCIR656
- Definira jednostavni video protokol za prijenos ne komprimiranog digitalnog video signala bio PAL ili NTSC (522 ili 625 linija)
- Standar je nastao na BT.601(CCIR601) koji definira prijenos podataka u formatu 4:2:2 interfaced u YUV (YCbCr)
- Standard definira 8 i 10 bitnih podataka serijski ili paralelno.
- Koristi se za prijenos podataka u televizorima između čipova

Multimedije arhitekture i sustavi

## ZV Port

- Korištenjem ZV porta prijenosna računala dobivaju performanse desktop računala u video svijetu
- ZV Port je jednosmjerna komunikacija između PC Card-a i VGA kontrolera
- U potpunosti je kompatibilno s CCIR601

Multimedije arhitekture i sustavi

## ZV Port

- Video format YUV 4:2:2, RGB 4:2:2, RGB raw dana 565/555
- Format podataka – YUV 4:2:2, RGB 4:2:2, RGB raw dana 565/555
- 8 bita
- Video format CCIR601, CCIR656, ZV port
- Format podataka – YUV 4:2:2, RGB 4:2:2, RGB raw dana 565/555

47

Multimedije arhitekture i sustavi

## ZV Port

- Više detalja u PC Card Standard
- Dodano u standard od verzije PC Card Standard 5.04 Update
- Trenutna verzija (koja je ujedno i konačna)
  - PC Card Standard 8.0
  - PC Card je narušten
- Trenutno se razvija Express Card koji podržava sve što i PC Card
- Fizički nisu kompatibilni

Multimedijске arhitekture i sustavi

## OV7670

- Signali
  - VSYNC
    - Postavljanjem u visoko dojavljuje se početak okvira slike. Generira se ječnom za svaku sliku
  - HREF
    - Postavlja se visoko i ostaje u visokom kada se šalju jedan red slike po završetku slanja jednog reda postavljaju se u nisko
  - PCLK
    - Na padajući ili rastući brid signala podatak na sabirnicu je valjan
    - Opcija rastući ili padajući brid može se podešiti preko I2C-a

Multimedijске arhitekture i sustavi

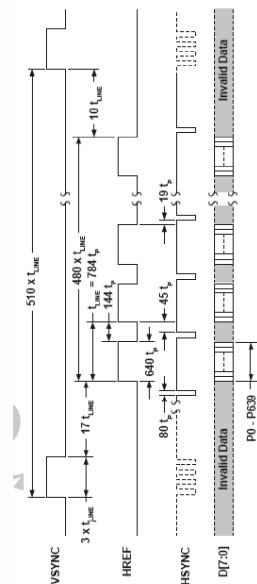
50

## ZV Port

- Signalni
  - VSYNC
    - Postavljanjem u visoko dojavljuje se početak okvira slike. Generira se ječnom za svaku sliku
  - HREF
    - Postavlja se visoko i ostaje u visokom kada se šalju jedan red slike po završetku slanja jednog reda postavljaju se u nisko
  - PCLK
    - Na padajući ili rastući brid signala podatak na sabirnicu je valjan
    - Opcija rastući ili padajući brid može se podešiti preko I2C-a

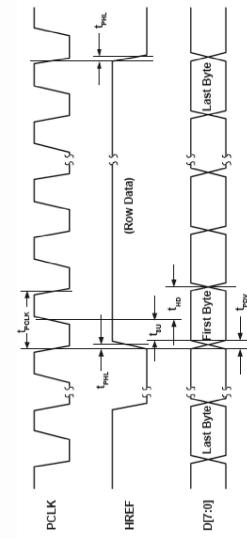
Multimedijске arhitekture i sustavi

51



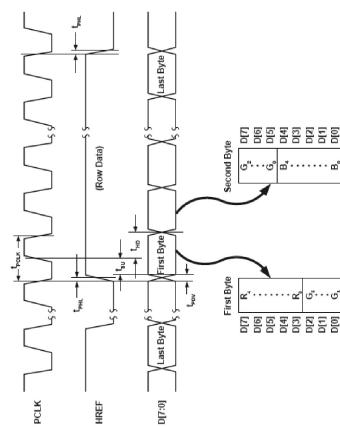
Multimedijске arhitekture i sustavi

## ZV Port



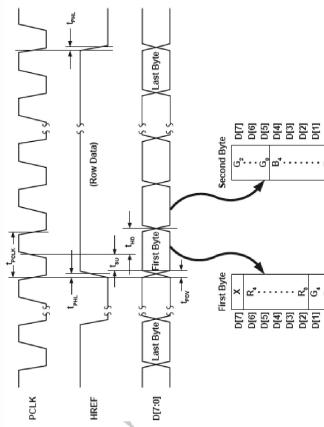
Multimedijске arhitekture i sustavi

## OV7670 – RGB565



Multimedijске arhitekture i sustavi

## OV7670 – RGB555



53

54

## OV7670

- Ostali signali koje koristimo
  - RESET
  - Sklopovski reset kamere
  - PWDN
  - Postavljanje kamere u "Power Down Mode"
    - Način rada smanjene potrošnje
    - U tom načinu rada senzor zaustavlja obradu video signala.

Multimedijeke arhitekture i sustavi

55

## OV7670

- Video senzor radi kao MASTER uređaj
  - Moguće ga je postaviti da radi i kako SLAVE uređaj
  - Tada procesor mora kontrolirati i generirati signale CHSYNC, VSYNC, PCLK. On se mora brinuti o horizontalnoj i vertikalnoj sinkronizaciji

Multimedijeke arhitekture i sustavi

56

## OV7670

- Frame execution mode
  - Predviđen za uređaje koji imaju mehaničku kontrolu otvora blende
  - Digitalni foto aparati

Multimedijeke arhitekture i sustavi

57

## OV7670

- Vaš zadatak
  - Kreirati vaš IP za čitanje sa ZV porta ili koristiti postojeći IP-ove (GPIO)
  - Koristimo jedno od dva I2C sučelje koje se nalazi na ARM procesor
  - Napisati program za čitanje s ZV porta
  - Napisati program za konfiguraciju kamere preko I2C porta
    - moramo usporiti kameru 60 fps je malo prebrzo za nas ☺, a možda i nije – ovisi o izvedbi

Multimedijeke arhitekture i sustavi

58

## Xilinx Processor IP Library

- iicps v1\_03\_a
  - Uređaj može biti master ili slave
  - Omogućeno je korištenje prekida prilikom čitanja i pisanja ili prozivanje
  - lako sklopolje podržava 7 i 10 bitnu adresu
  - izvedeni upravljački programi podržavaju samo 7 bitno adresiranje

Multimedijeke arhitekture i sustavi

59