

# SACCDMM - Curs 08

## Standardul JPEG 2000

# Subband coding: motivation

---

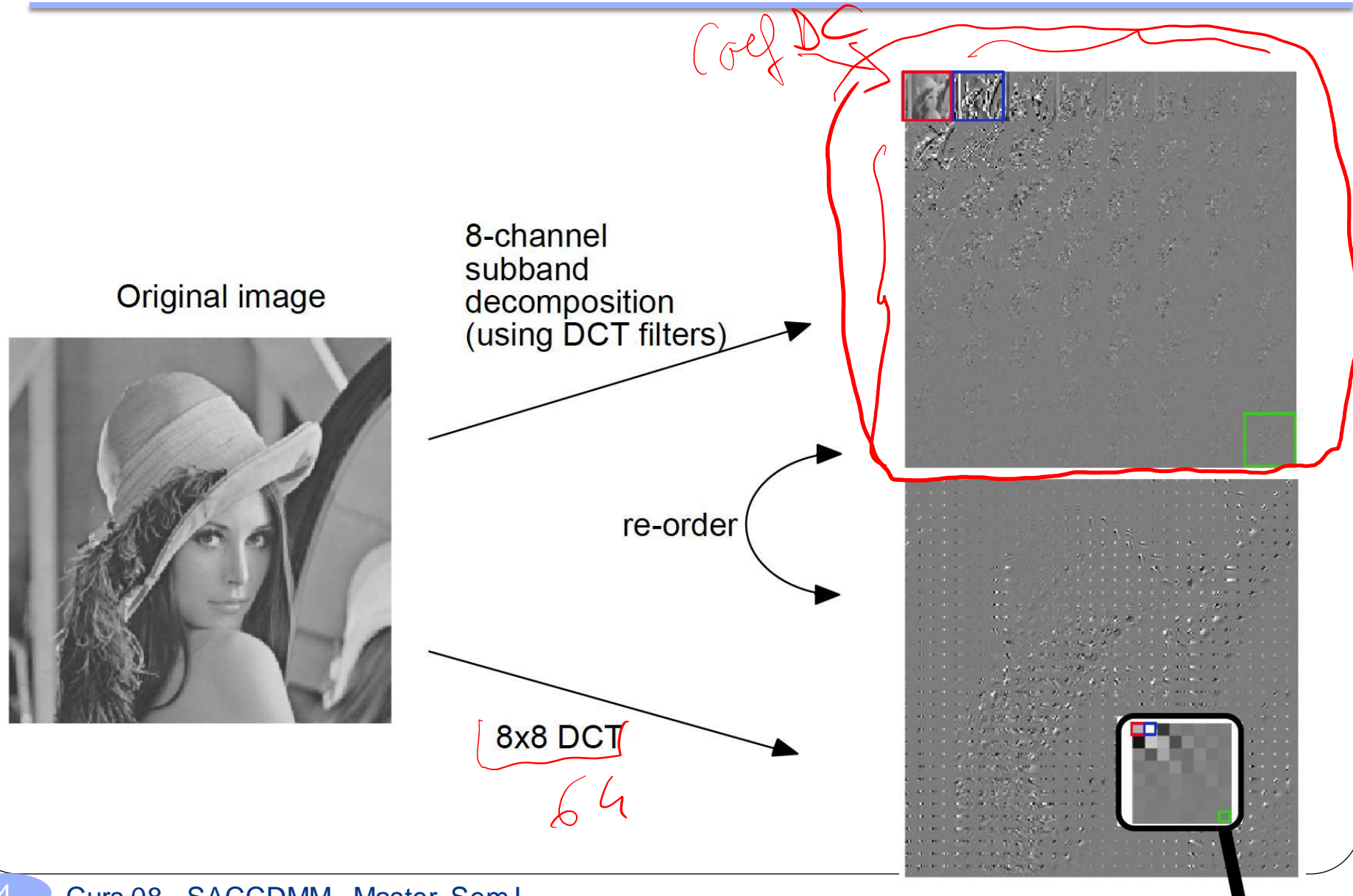
- Coding with block-wise transform introduces visible blocking artifacts, as bit-rate decreases.
- Can we, somehow, overlap adjacent blocks,
  - thereby smoothing block boundaries,
  - but without increasing the number of transform coefficients?
- Solution: subband transform.

# Subbands vs. block-wise transform

---

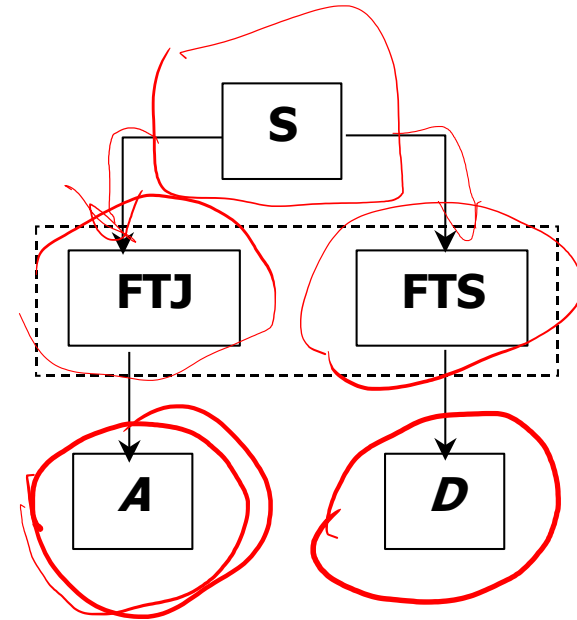
- Blockwise transforms are a special case of subband decompositions with:
  - Number of bands  $m =$  order of transform  $N$
  - Length of impulse responses of analysis/synthesis filters  $\leq m$
- Filters used in subband coders are **not** in general orthogonal.
- Linear phase is desirable for images.

# Subbands vs. block-wise transform (cont.)



# Transformata Wavelet discreta (DWT - Discrete Wavelet Transform)

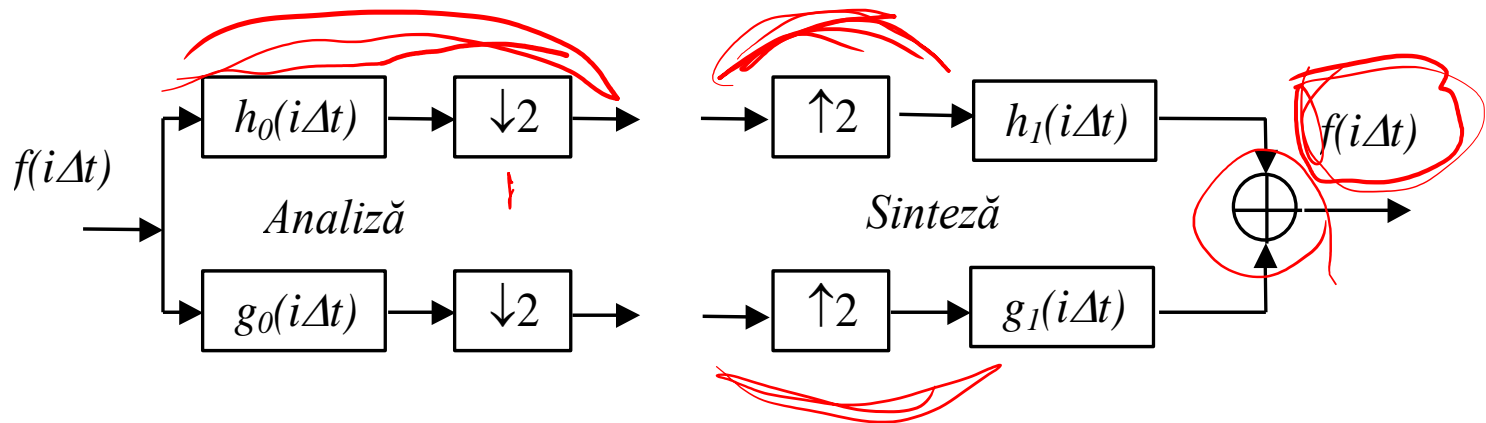
- calculul coeficientilor Wavelet pentru orice scala -> costisitor
- aplica DWT – set de scale si pozitii
- metoda diadica  $\Leftrightarrow$  factor de scala si pozitie = putere a lui 2
  - permite realizarea unei analize mult mai eficiente și mai corecte
- putem implementa DWT folosind filtrele
  - prin descompunerea in subbenzi folosind două canale
    - conținutul de joasă frecvență – definește „identitatea” semnalului.
    - conținutul de frecvență înaltă aduce îmbunătățiri în descrierea semnalului.
- Ex. - vocea umană
  - eliminăm frecvențele înalte, aceasta se va auzi diferit dar va fi inteligibilă
  - eliminăm frecvențele joase, vocea este posibil să fie distorsionată.



- 
- reprezentarea cu filtre a DWT
    - avem două componente:
      - Aproximările (A) – sunt componente cu factor de scală mare, adică componente de frecvență joasă
      - Detalii (D) – componente cu factor de scală mic, adică componente de frecvență înaltă

# DWT

- semnal original trecut prin DWT (trecut prin două filtre complementare)
  - $\Rightarrow$  2 ori numărul de esantioane (două componente: A – aproximări, D – detalii)
  - trebuie făcută o subesantionare  $\Rightarrow$  la ieșirea bancului de filtre se obține aceeași cantitate de informație ca la intrare
- la reconstrucție se aplica operațiunea inversă



# DWT bidimensionala

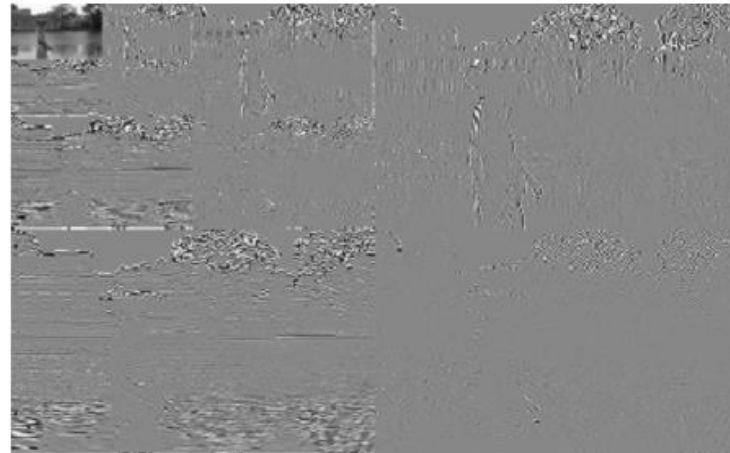
- Aplicarea transformatei wavelet unidimensionale discrete de două ori:

- pe linii
- pe coloane

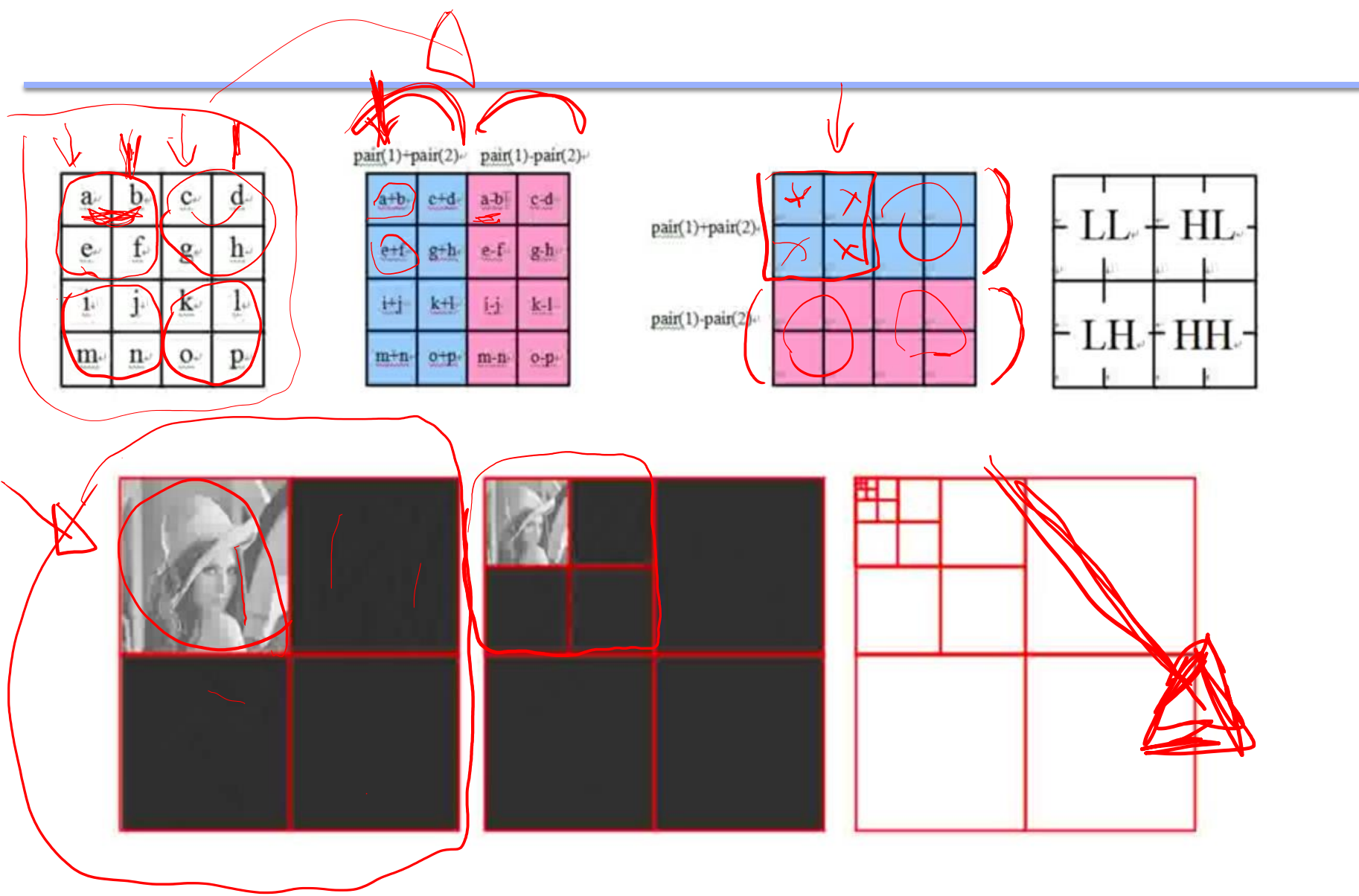
LL – informatii de joasa frecventa; HL – high vertical, low orizontal;  
LH – low vertical, high orizontal; HH – high vertical, high orizontal

LL	LH
HL	HH

1	2	5	8
3	4		
6		7	
9			







# JPEG 2000

---

- Noiembrie 1997 – evaluare a 20 algoritmi
- alegerea transformatei Wavelet – transformare de baza
- JPEG 2000 – 6 parti
  - partea 1 – modul de baza
    - complexitate minima
    - acopera 80% din aplicatii
    - defineste un format de fisier
    - devine standard international in Decembrie 2000
  - partea 2-6 –
    - extensii de algoritm
    - extensii de format de fisier
    - diferite stadii de dezvoltare

# JPEG 2000 “parts”

- partea 2
  - imbunatatirea performantelor
  - complexitate mare
  - IPR (intellectual property rights)
- partea 3
  - Motion JPEG 2000 - MJP2
- partea 4
  - testare
- partea 5
  - implementare software
    - implementare JAVA – grup JJ2000 (format din Canon, EPFL, Ericsson)
    - implementare C – Image Power, University of British Columbia
- partea 6
  - fisier complex – aplicatii de scanare de documente, fax

# JPEG 2000 – de ce?

- nu numai pentru a imbunatati compresia
- o noua reprezentare a imaginii
- extinderea domeniului de aplicabilitate
- facilitati:
  - imbunatatirea eficientei codarii
  - compresie cu fara pierderi
  - reprezentari la rezolutii multiple
  - sisteme de tip “embedded bit-stream”
    - decodare progresiva ←
    - scalabilitate SNR ←
  - impartirea in blocuri (tile component - tiling) ←
  - codarea ROI →
  - rezistenta la erori
  - accesul si prelucrarea aleatoare a sirului codat
  - fisier mai flexibil

*se aplica pe  
intreaga img.*

JPEG → JPEG 2000

# JPEG 2000 – Cum?

- DCT -> DWT
  - compactare energetica mai buna
  - decorelare
  - reprezentare imaginii la mai multe rezolutii
  - compresie cu/fara pierderi – acelasi sir codat
- codorul Huffman - > codare aritmetica adaptiva (codorul MQ)
- planurile de biti sunt codate independent (blocurile de codare)
- introducerea sistemului de coordonate – faciliteaza:
  - operatii de rotatii
  - inversare

# Pașii codării JPEG 2000 :

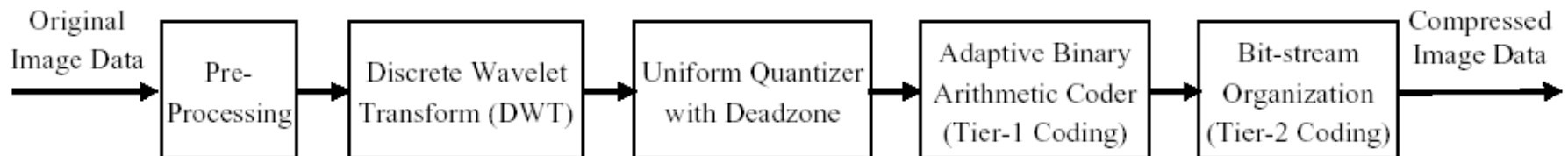
- descompunerea imaginii din spatiul de reprezentare RGB într-un alt spatiu de reprezentare care realizeaza o decorelatie intre componenta de luminanta si cele de crominanta (ex. formatele YUV, YC<sub>r</sub>C<sub>b</sub>, etc.)
- Componentele pentru reprezentarea imaginii sunt împărțite în blocuri de imagine elementare (tiling) - acest proces nu este obligatoriu
- se aplică transformata pe fiecare bloc elementar de imagine - transformata folosită în standardul JPEG2000 este transformata Wavelet discretă 2D
- subbenzile de coeficienți sunt cuantizate și grupate în „blocuri de codare”
- planurile de biți ale coeficienților dintr-un bloc de codare sunt codate entropic
- codarea poate fi implementată astfel încât să luăm în considerare ROI
- adăugarea elementelor de marcare suplimentare pentru reducerea erorilor la transmisie și salvare
- obținerea șirului codat JPEG 2000

# Acești pași pot fi grupați în 3 etape mari:

- etapa de **pre-procesare**
- etapa de **procesare de bază**
- etapa de **generare a șirului codat**

## II. Arhitectura JPEG 2000

- pre-procesare
- transformata Wavelet
- cuantizorul uniform
- codorul aritmetic adaptiv – nivelul 1 de codare
- organizarea sirului codat – nivelul 2 de codare





# Pre-procesarea

- imagini – 3 (RGB,  $Y C_r C_b$ ) ->  $2^{14}$  (16384) componente
- valorile esantioanelor: intregi cu/fara semn - esantion reprezentat cu B biti
- reprezentarea fara semn:  $(0, 2^B - 1)$  - sunt translatate simetric fata de zero (prin scaderea valorii  $2^{B-1}$ )  $\Rightarrow [-128, 127]$   
 $B=8, \{0, 255\}$
- reprezentarea cu semn:  $(-2^{B-1}, 2^{B-1} - 1)$  - nu sunt translatate
- impartirea imaginii in zone drept. de dim. egale (exceptie marginile)
- dimensiunea zonelor – arbitrara (poate fi chiar toata imaginea)
- fiecare zona este comprimata independent de celelalte (putand folosi un setul propriu de parametrii)  $\Rightarrow$
- avantaj – la codare cand memoria disponibila este redusa

# Pre-procesarea

- valoare esantion (fara semn) –  $2^{B-1}$  (simetrie fata de zero)
- esantioanele cu semn nu sunt “deplasate”
- transformarea spatiului de culoare
  - ICT – irreversible color transform RGB –  $YC_bC_r$  (lossy compresion)

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.500 \\ 0.500 & -0.41869 & -0.08131 \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0 & 1.402 \\ 1.0 & -0.34413 & 0.71414 \\ 1.0 & 1.772 & 0 \end{pmatrix} \times \begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix}$$

- RCT – reversible color transform RGB (lossless and lossy coding)

$$Y = \left\lfloor \frac{R + 2G + B}{4} \right\rfloor, \quad U = R - G, \\ V = B - G,$$

$$G = Y - \left\lfloor \frac{U + V}{4} \right\rfloor, \quad R = U + G, \quad B = V + G.$$

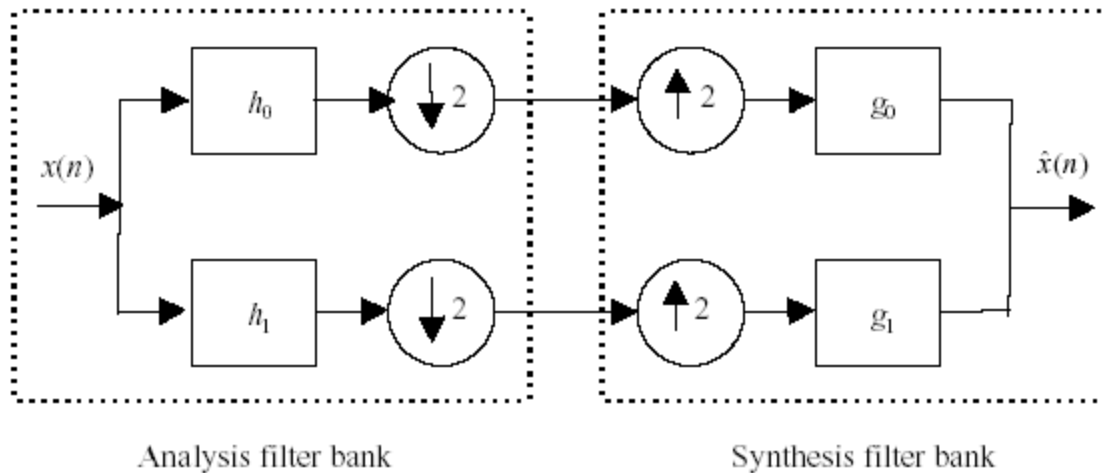
- Dacă pentru standardul JPEG se proceda la o operație de subeșantionare a componentelor de cromaticitate  $C_r$ ,  $C_b$  acest lucru nu este recomandabil pentru JPEG2000.
- Practic în JPEG2000 se realizează un prim pas al DWT pe componentele de cromaticitate. *Detalii*
  - Componentele LH, HL și HH se elimină și practic se obține o subeșantionare cu 2 atât în plan vertical cât și în plan orizontal.

# Transformata Wavelet Discreta (DWT)

- DCT pe blocuri -> DWT pe fiecare "tile" (sau pe intreaga imagine)
- caracteristici:
  - reprezentarea multi-rezolutie
  - eliminarea artefactelor (efect blocking) la comp. mare
  - folosirea filtrelor DWT intregi -> putem implementa:
    - codarea cu pierderi
    - codarea fara pierderi

# 1D –DWT

- succesiune de perechi de filtre FTJ, FTS + subesantionare cu 2



- perechea FTS, FTJ – analiza bancurilor de filtre
- FTJ – pastreaza frecventele joase (imagine incetosata)
- FTS – pastreaza frecventele inalte (contururi, texturi, detalii etc.)

# 1D-DWT multiplu

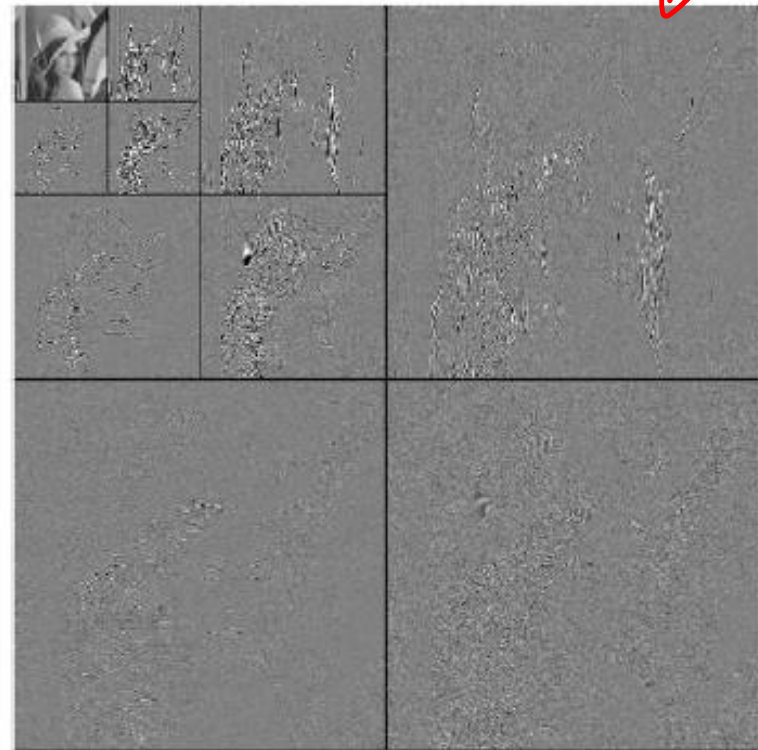
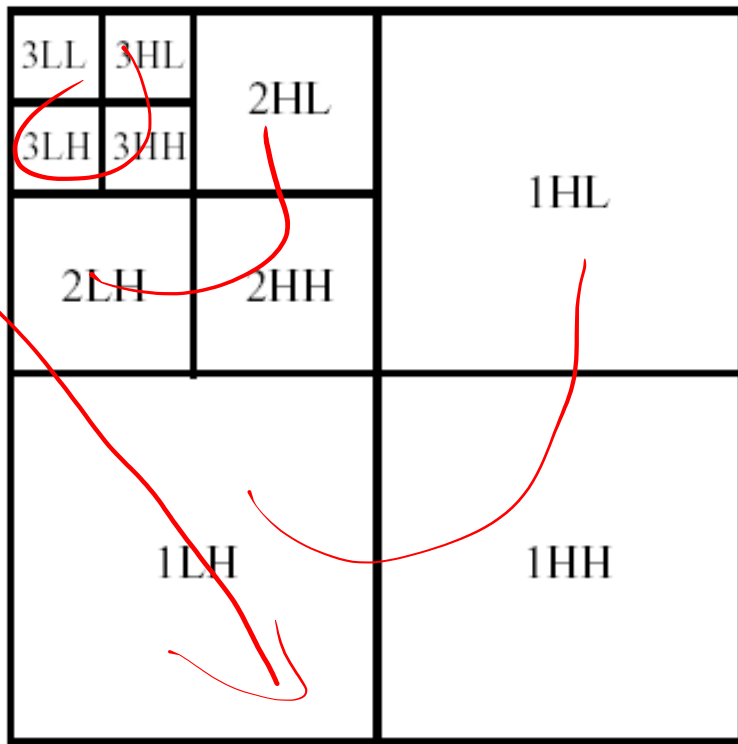
- primul pas DWT – secventa low-pass este inca corelata
- se aplica DWT pe secventa *low-pass* - *dyadic*
- DWT pe secventa *high-pass* – nu aduce castig!
- JPEG 2000 (part 1) – descompunerea dyadic
- JPEG 2000 (part 2) – + descompunerea pe secventa high-pass

# DWT – 2D

- 1D – DWT pe randuri ( $h_0, h_1$ )
- 1D – DWT pe coloane ( $h_0, h_1$ )
- dupa 1 pas 2D – DWT -> 4 subbenzi
  - subbanda LL
  - subbanda HL (H – pe orizontala; L verticala)
  - subbanda LH (L – orizontala; H – verticala)
  - subbanda HH
- subbenzile AC – adauga ~~128~~ pentru vizualizare mai buna

# Exemplu 2D –DWT

- bancuri de filtre DWT





# Implementarea DWT

---

- buffer pentru intreaga imagine
- impartirea imaginii in blocuri (tiles) – reduce necesarul de memorie
- folosirea schemei de lift-are – *lifting scheme*
  - reducerea cantitatii de memorie
  - reducerea complexitatii de calcul
  - rapiditate de calcul
  - adoptata in JPEG 2000

# Schema de *lift-are*

---

- impartirea semnalului de intrare
  - secvente de esantioane **pare**  $d_i^0$
  - secvente de esantioane **impare**  $s_i^0$
- alternarea pasilor:
  - predictie
  - update

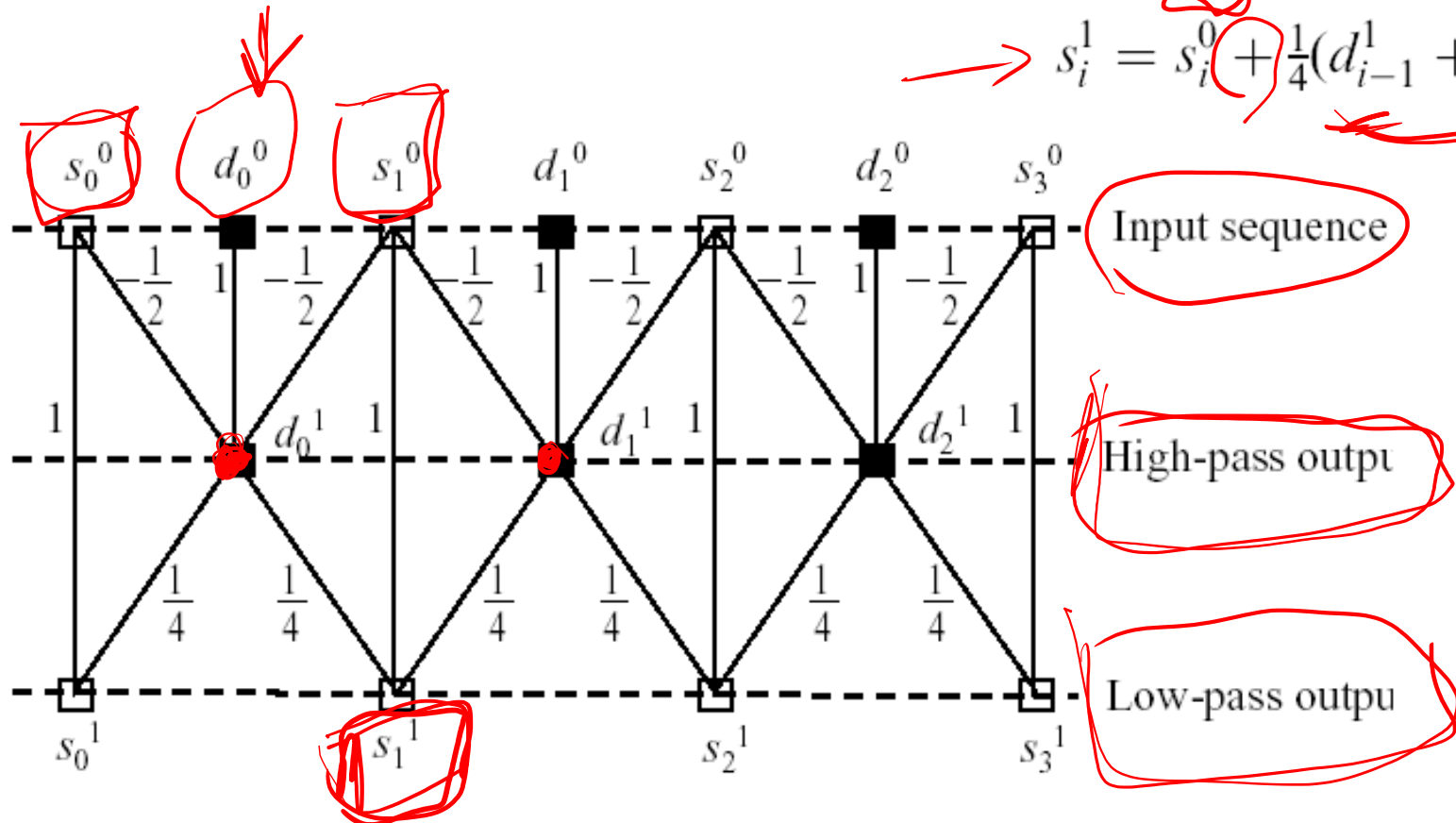
# Schema de lift-are pentru filtru (5,3)

- predictia – fiecare esantion par se exprima
- update – fiecare esantion impar se exprima
- suficient un singur pas

$$I = I_{FS} + I_{FS}$$

$$\rightarrow d_i^1 = d_i^0 - \frac{1}{2}(s_i^0 + s_{i+1}^0).$$

$$\rightarrow s_i^1 = s_i^0 + \frac{1}{4}(d_{i-1}^1 + d_i^1).$$



# Cuantizarea

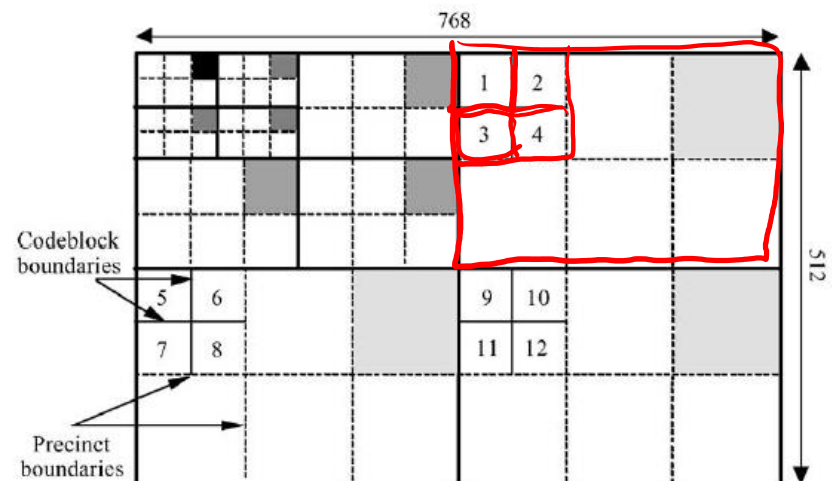
- Se folosește o cuantizare uniformă pentru fiecare bandă de coeficienți DWT.
- Această etapă induce pierderi în imaginea reconstruită.
- Fiecare coeficient  $a_b(u,v)$  al subbenzii  $b$  este cuantizat la valoarea  $q_b(u,v)$  conform formulei:

$$q_b(u,v) = \text{semn}[a_b(u,v)] \cdot \left\lfloor \frac{|a_b(u,v)|}{\Delta b} \right\rfloor$$

unde  $\Delta b$ , reprezintă factorul de cuantizare pentru subbanda  $b$

# Codarea entropica

- fiecare subbanda este codata separat
- fiecare subbanda este impartita in *blocuri de codare*
- fiecare bloc de codare este codat independent
- dimensiunea blocului de codare:
  - intreg, putere a lui 2
  - numarul maxim de coeficienti dintr-un bloc  $\leq 4096$
  - nivelul blocurilor de codare nu poate fi mai mic de 4



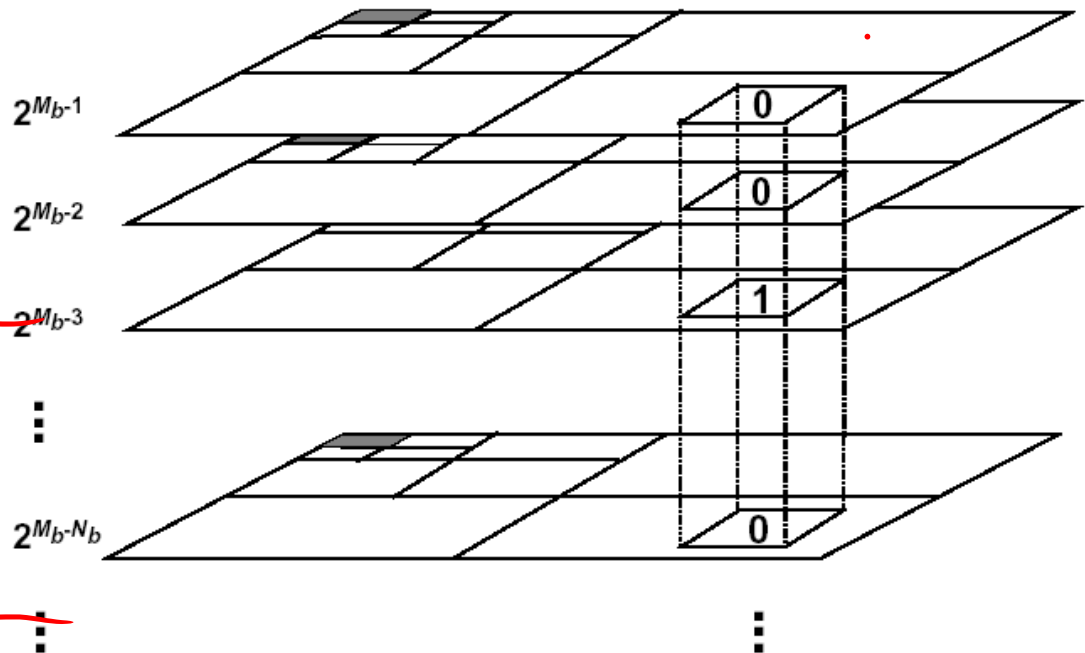
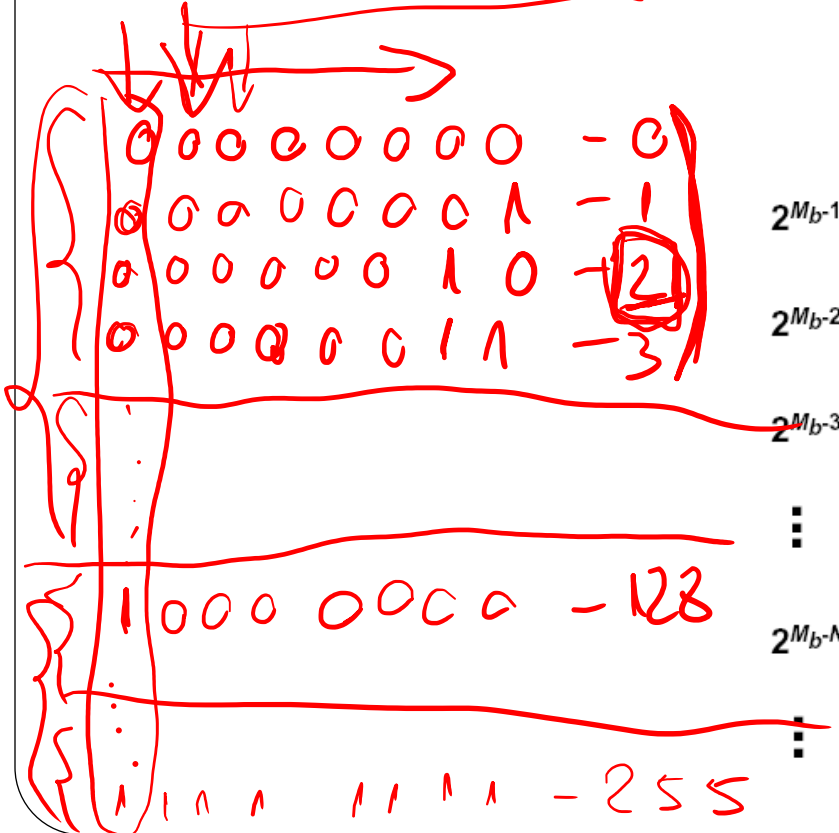
# Avantaje ale codarii independente

---

- accesul aleator la imagine
- implementarea pe arhitecturi paralele
- functionalitati de prelucrare a imaginii
- rezistenta la erori
- eficienta controlului ratei de bit
- flexibilitate maxima pentru aranjarea ordinii de aparitie

# Codarea pe plane de biti

- coeficienti
  - nesemnificativi – bitul este zero
  - semnificativi – bitul este 1 – incepe codarea  
coeficientului



# Codarea aritmetica si MQ

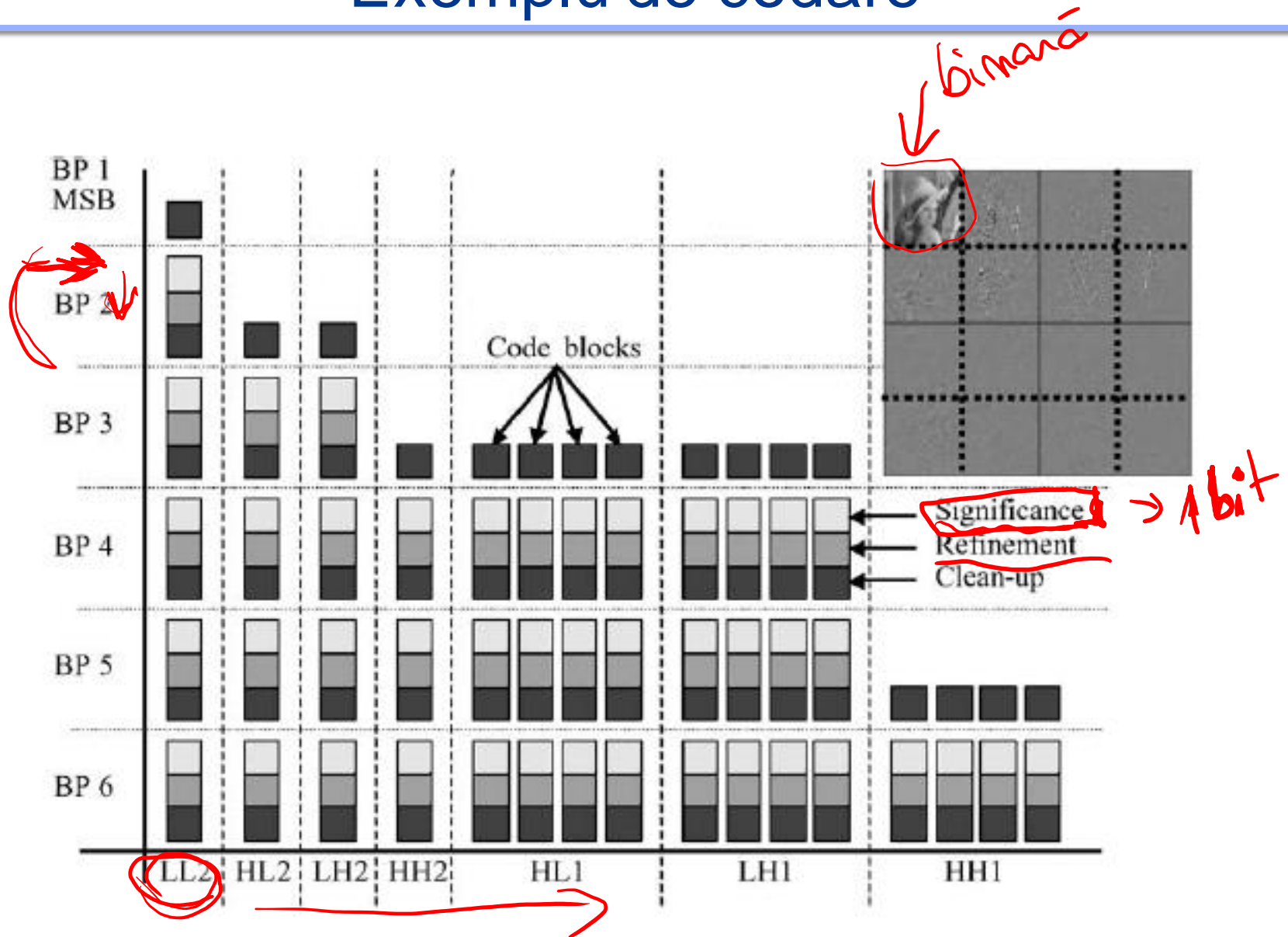
- codeaza o intreaga secventa de simboluri
- cuvantul de cod – prin impartirea recursiva a  $(0,1)$  dupa probabilitati
- codare adaptiva



# Etapele codarii pe planuri de biti

- fiecare plan de biti este codat independent folosind 3 etape (sub-bitplanes) cu “intreruperea” sirului codat dupa fiecare etapa
- avantaje:
  - pentru optimizarea afisarii imaginilor la diferite rezolutii
  - minimizarea distorsiunilor care pot aparea in sirul codat
- pasul I – “significance propagation”
  - codarea coeficientilor care au probabilitatea sa devina semnificativi
- pasul II – “refinement”
  - imbunatatirea codarii coeficientilor semnificativi
- pasul III – “cleanup”
  - codarea restului coeficientilor din plan care nu vor deveni semnificativi

# Exemplu de codare



# Organizarea sirului codat in JPEG2000

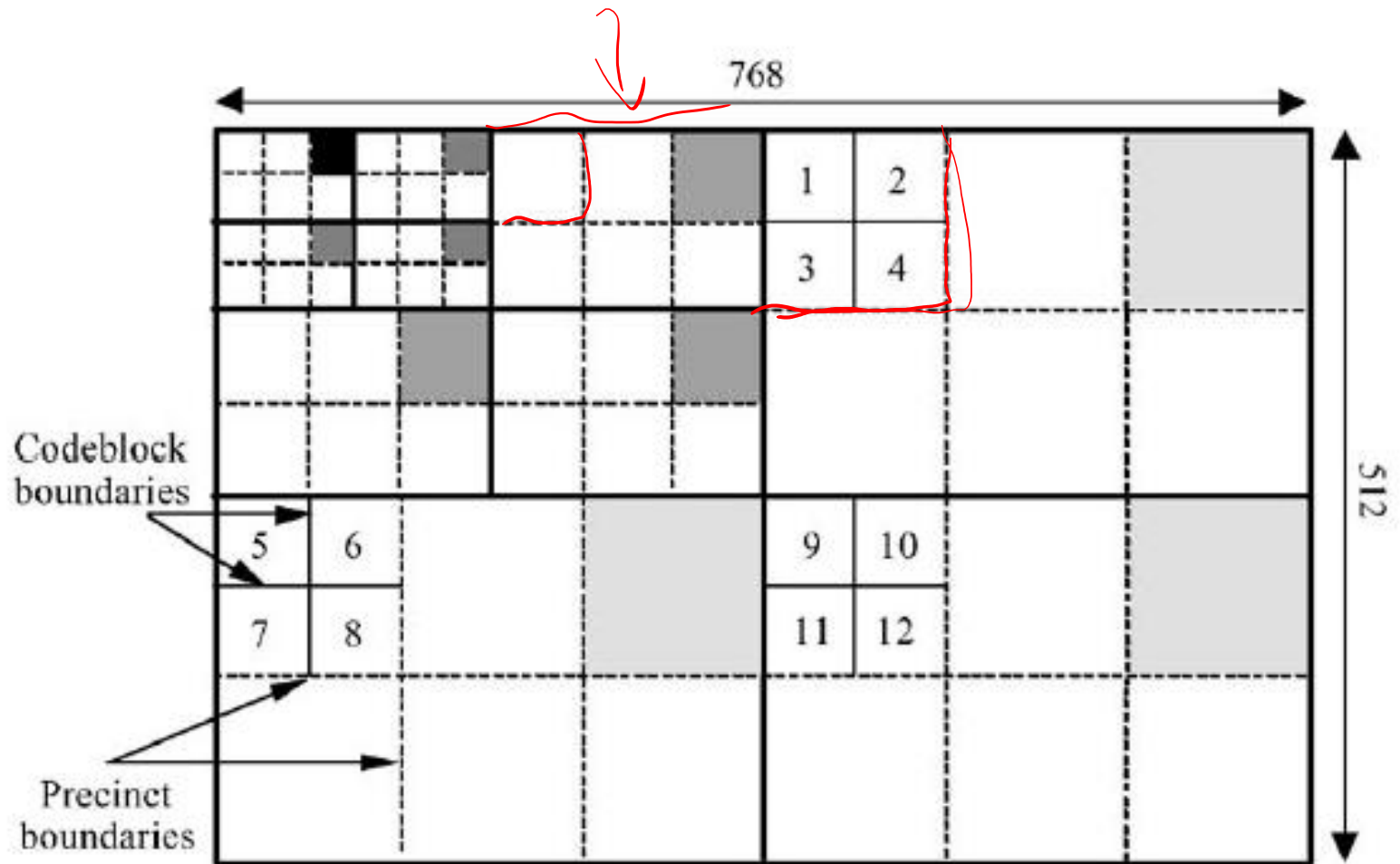
- de ce?
  - acces aleator la imagine
  - ROI
  - scalabilitate
- impartirea imaginii
  - canale de culoare (componente)
  - regiuni spatiale (tiles) → *usual intreg imagine*
  - regiuni de frecventa (subbenzi ale DWT)
  - regiuni spatiu-frecventa (blocurile de codare)

*YCbCr / YUV*

# Alte structuri de date utilizate

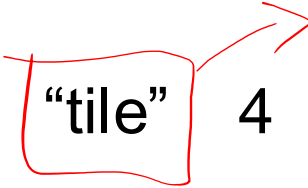


- **precinct** – colectie de blocuri de codare (aproximativ uniforme) din toate subbenzile cu acelasi nivel de descompunere DWT
- Fiecare precinct este împărțit în blocuri elementare de codare
- **pachete** – este un segment continuu din sirul codat care contine un numar din nivelele de codare pe planuri de biti pentru fiecare bloc de codare din precinct
- pachetele din fiecare precinct la toate rezolutiile sunt grupate in **layer-e**

## Exemplu de structuri “precinct” si blocuri de codare



# Ordinea de aparitie a pachetelor din sirul codat

---

- progression order
- pentru o componenta  4 componente pentru identificarea pachetului:
  - componenta 
  - rezolutia
  - nivelul (layer)
  - pozitia (precinct)
- pachetele pentru o componenta specifica sunt generate prin scanarea precinct-ului intr-o ordine bine stabilita 

# Algoritmi de citire ordonata

- Layer-resolution-component-position – LRCP
  - aplicatii de BD de imagini
- Resolution-Layer-Component-Position – RLCP
  - aplicatii client-server – clientii acceseaza imagini la diferite rezolutii
- Resolution-Position-Component-Layer – RPCL
  - aplicatii unde este necesara accesarea imaginilor la diferite rez. si poz.
- Position-Component-Resolution-Layer – PCRL
  - imbunatatirea calitatii imaginii pentru o pozitie specifica
- Component-Position-Resolution-Layer – CPRL
  - obtinerea celei mai bune calitati a unei imagini pentru o pozitie particulara

# Performantele diferitelor algoritmi JPEG 2000

- **parametrii:**
  - eficienta codarii
  - viteza
  - complexitatea implementarii
- **compresia cu pierderi:**
  - dimensiunea componentei "tile"
  - dimensiunea blocului de codare
  - filtre DWT
  - nivelelor de descompunere DWT
  - optiuni de codare entropica
  - efectul ciclurilor multiple de codare
- **compresia fara pierderi:**
  - transformari de culoare reversibile
  - optiuni ale codorului fara pierderi
  - optiuni de codare entropica



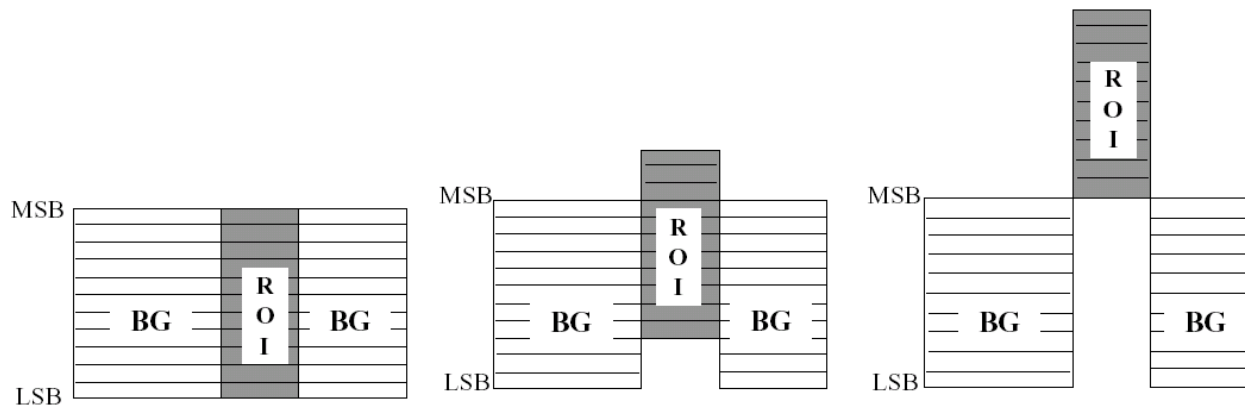
# Imbunatatiri ale standardului JPEG 2000

---

- definirea ROI
- scalabilitatea spatiala si dupa RSZ
- imunitate la erori
- posibilitatea de protejare IPR

# ROI

- se pot defini diferite zone din imagine
- importanta mai mare sau mai mica a zonelor
- principiu:
  - translatarea coeficientilor din ROI
  - se codeaza mai intai planurile de biti MSB
  - apoi se codeaza planurile BG



# Scalabilitatea

- **dupa RSZ**
  - pentru aplicatii unde este utila accesul la imagini cu multiple nivele de calitate
  - aceeasi rezolutie spatiala la **RSZ** diferiti
- **spatiala**
  - aplicatii unde avem nevoie de imagini la rezolutii multiple dar la aceeaasi calitate
  - codare imagine de rezolutie mica + transmitere versiune de rezolutie superioara pentru imbunatatirea primei versiuni

# Imunitate la zgomot

---

- pentru aplicatii pe sisteme mobile si Internet
- doua nivele de imunitate la zgomot:
  - la nivelul codorului entropic
    - folosirea blocurilor de codare
    - folosirea terminatoarelor in procesul de codare aritmetica
  - la nivelul blocului de formare a sirului de biti
    - folosirea pachetelor scurte
    - folosirea marker-ilor de resincronizare

# Comparatie JPEG <-> JPEG 2000

Imagine	Tip	Rezoluție	Observații	Compresie	RSZ(dB)
<b>Barb.bmp</b>	8biți/pixel	512x512	-		-
Barb_1_27_2000.jpc	8biți/pixel	512x512	1:27	JPEG2000	29.22
Barb_1_4_2000.jpc	8biți/pixel	512x512	1:4	JPEG2000	38.55
Barb_1_27.jpg	8biți/pixel	512x512	1:27	JPEG	23.28
Barb_1_4.jpg	8biți/pixel	512x512	1:4	JPEG	36.68
<b>Bogdan.bmp</b>	24biți/pixel	256x256	-		-
Bogdan_41_2000.jpc	24biți/pixel	256x256	1:41	JPEG2000	26.62
Bogdan_7_2000.jpc	24biți/pixel	256x256	1:7	JPEG2000	27.67
Bogdan_41.jpg	24biți/pixel	256x256	1:41	JPEG	23.05
Bogdan_7.jpg	24biți/pixel	256x256	1:7	JPEG	34.8*

factor de comp.  
mare

# JPEG <-> JPEG 2000



barb.bmp



barb\_1\_27\_2000.jpc



barb\_1\_27.jpg



bogdan.bmp



bogdan\_1\_41\_2000.jpc



bogdan\_1\_41.jpg

# Comparație între schema de compresie bazată pe DCT și cea bazată pe wavelet

