



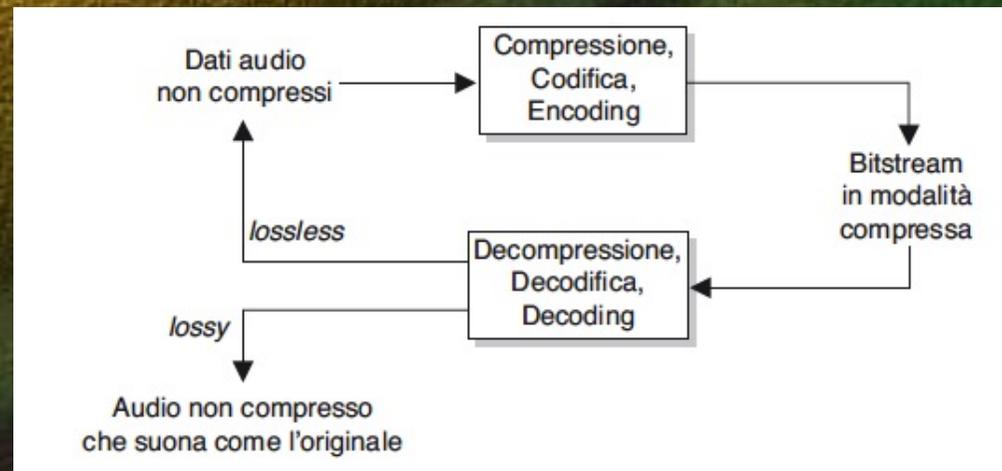
# Tecnologie digitali per il suono e l'immagine 2021/22

Vincenzo Lombardo  
Corso di Laurea in DAMS  
Università di Torino

Mutuato in parte da Elaborazione audio e musica  
(Laurea Magistrale di Informatica)

# Compressione del suono

Come risparmiare spazio e tempo lavorando sulla forma d'onda



# La compressione audio

L'arte di minimizzare le risorse per i dati audio

- ridurre la memoria occupata
- ridurre i costi di trasmissione
- Obiettivo: buona qualità in confronto all'audio non compresso

# Riproduzione trasparente

- Audio che anche “orecchie” sensibili non riescono a distinguere dall'originale
- Numero minimo di bit per riproduzione trasparente?

# Audio digitale

- Vantaggi
  - Alta fedeltà delle copie del segnale
  - Robustezza
  - Gamma dinamica estesa
- Svantaggio: alto tasso di trasferimento dati
  - Campionamento: 44.1 kHz (CD), 48 kHz (DAT)
  - Quantizzazione: PCM lineare a 16 bit
  - Molto spazio occupato
  - Ampia banda occupata sulla trasmissione

# Esempio: 1 min stereo, qualità CD

- Parametri

- campionamento = 44,100 c/s
- quantizzazione = 16 bit = 2 byte
- canali = 2

- Memoria =  $44.100 \text{ c/s} * 2 \text{ can} * 2 \text{ byte/c} * 60 \text{ s}$   
~ 10 Mbyte

- Tempo =  $10 \text{ MB} * 8 \text{ bit/byte} / (2 \text{ Mbps} * 60 \text{ s})$   
~ 40 sec

## Inoltre ...

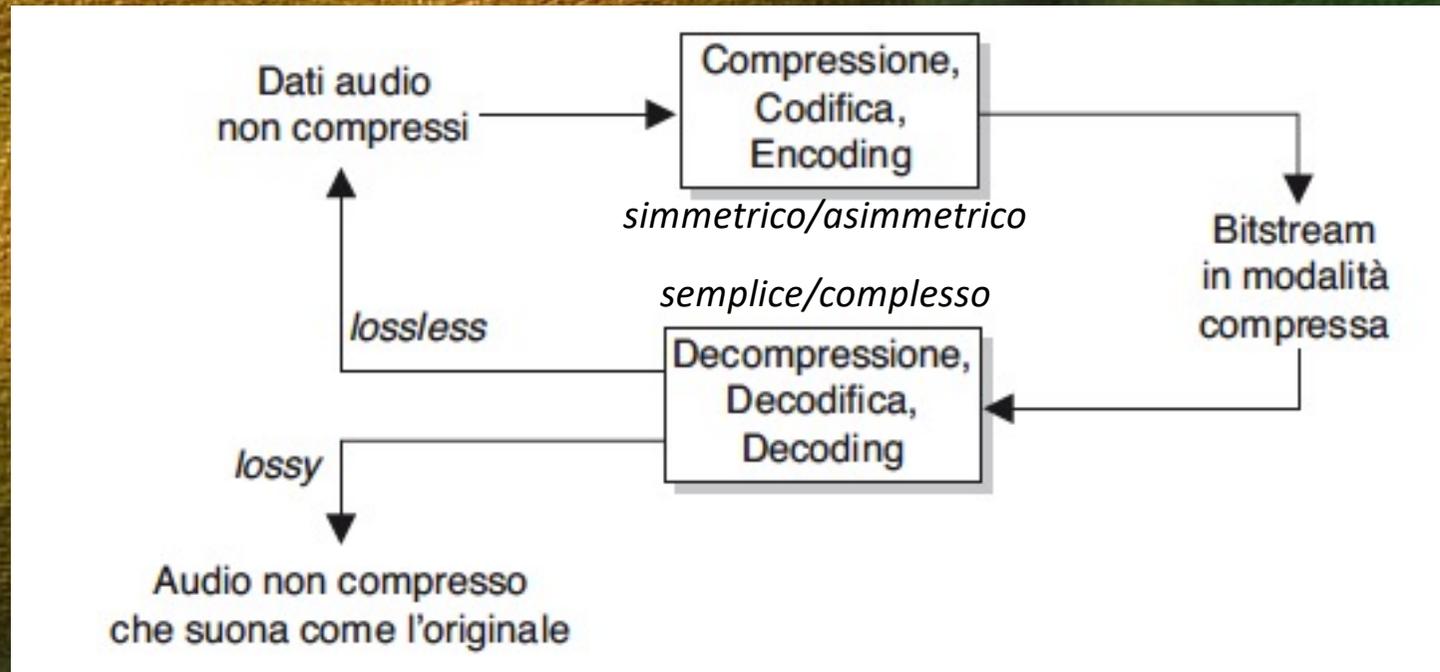
- Problemi di banda per applicazioni multimediali
- Accesso e trasferimento dati sempre più veloce, ma ...

... immagini, video, testo, e audio di alta qualità sincronizzati tutti assieme

# Due livelli di applicazioni

- Bassa qualità
  - Telefonia
- Alta qualità
  - colonne sonore per giochi
  - memorie allo stato solido per la musica
  - audio su Internet
  - broadcast di audio digitale (Radio e TV)

# Schema generico co/dec





# Schemi di compressione semplici

# Compressione del silenzio

- Utile per intensità vicine allo 0
- Silenzio = sequenze di campioni di intensità nulla

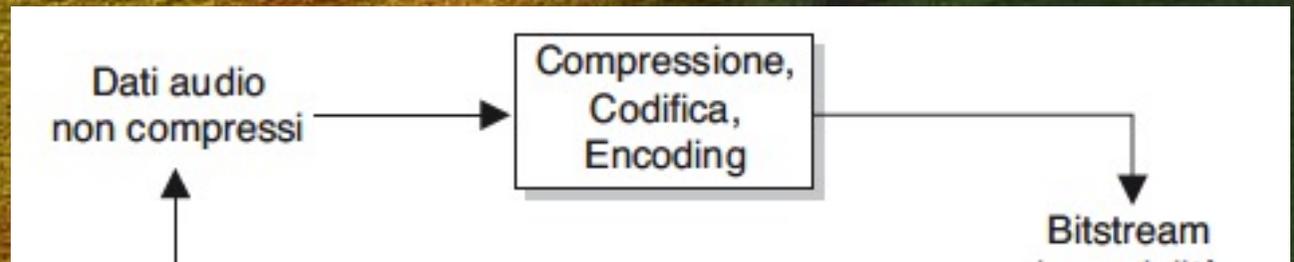
# RLE (Run Length Encoding)

- Compressione lossless
- Codifica un run

Esempio:

- “ddddddhyyyyyy” → “7d8h9y”

# RLE encoder

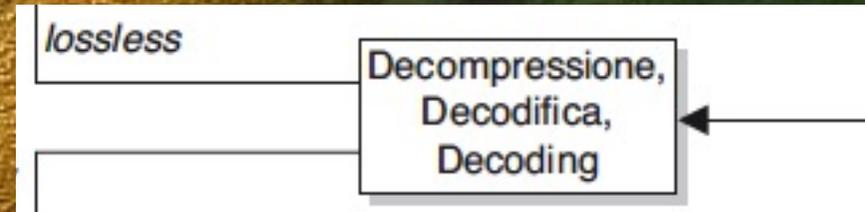


- Init stream;
- Init valore attuale x a un valore impossibile X;
- Init contatore a 0
- While not end-of-stream do
  - Leggi prossimo valore y
  - if x = y then
    - Incrementa contatore di 1
  - Else
    - Scrivi <contatore, x>;
    - init contatore a 1;
    - Valore attuale x diventa y

“ddddddhhhhhhhhhyyyyyyy” → “7d8h9y”

# RLE decoder

- `inizializza lo stream;`
- `Init contatore a 0`
- `While not end-of-stream do`
  - `Leggi prossima coppia <c,x>`
  - `For c volte do`
    - `Scrivi x`

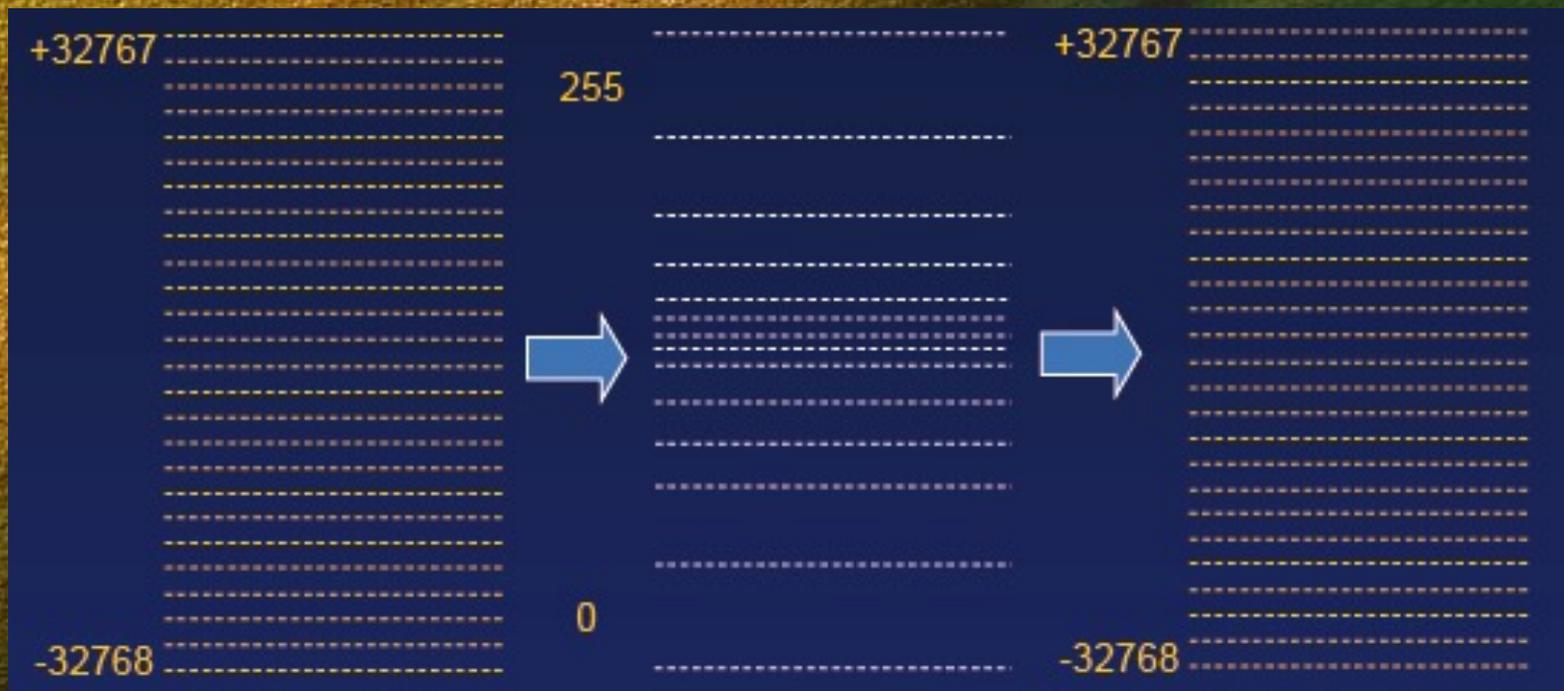


`“ddddddhhhhhhhyyyyyyy” ← “7d8h9y”`

# Compressione del silenzio (lossy): RLE + soglie

- Due casi
  - soglia di intensità sonora sotto la quale il segnale è interpretato come silenzio
  - minimo numero di campioni abbastanza forti che fungano da chiusura di una sequenza di silenzio
- Esempio: 15 deboli + 2 forti + 13 deboli

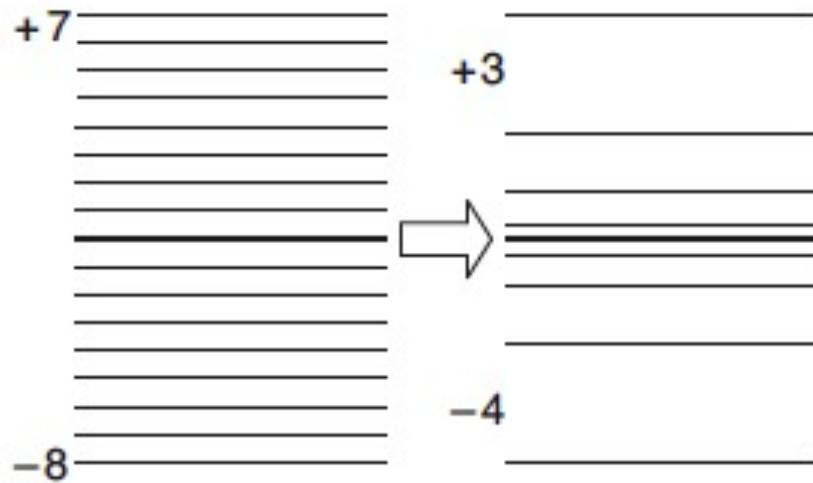
# ITU-T G.711, $\mu$ -law e A-law



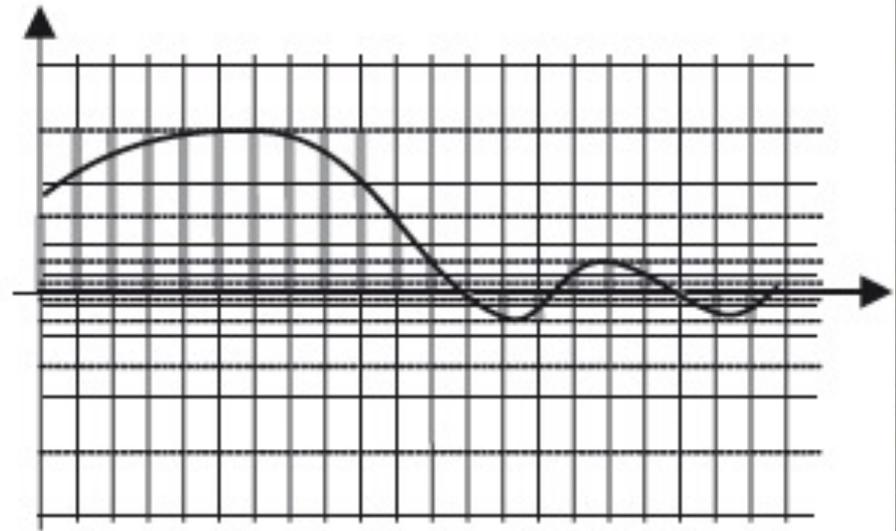
# $\mu$ -law

- Quantizzazione logaritmica
- Gamma dinamica di 14 bit con una codifica a 8-bit
- Per i servizi voce ISDN in Nord America e Giappone
- Semplice il computo della codifica

# Principio $\mu$ -law

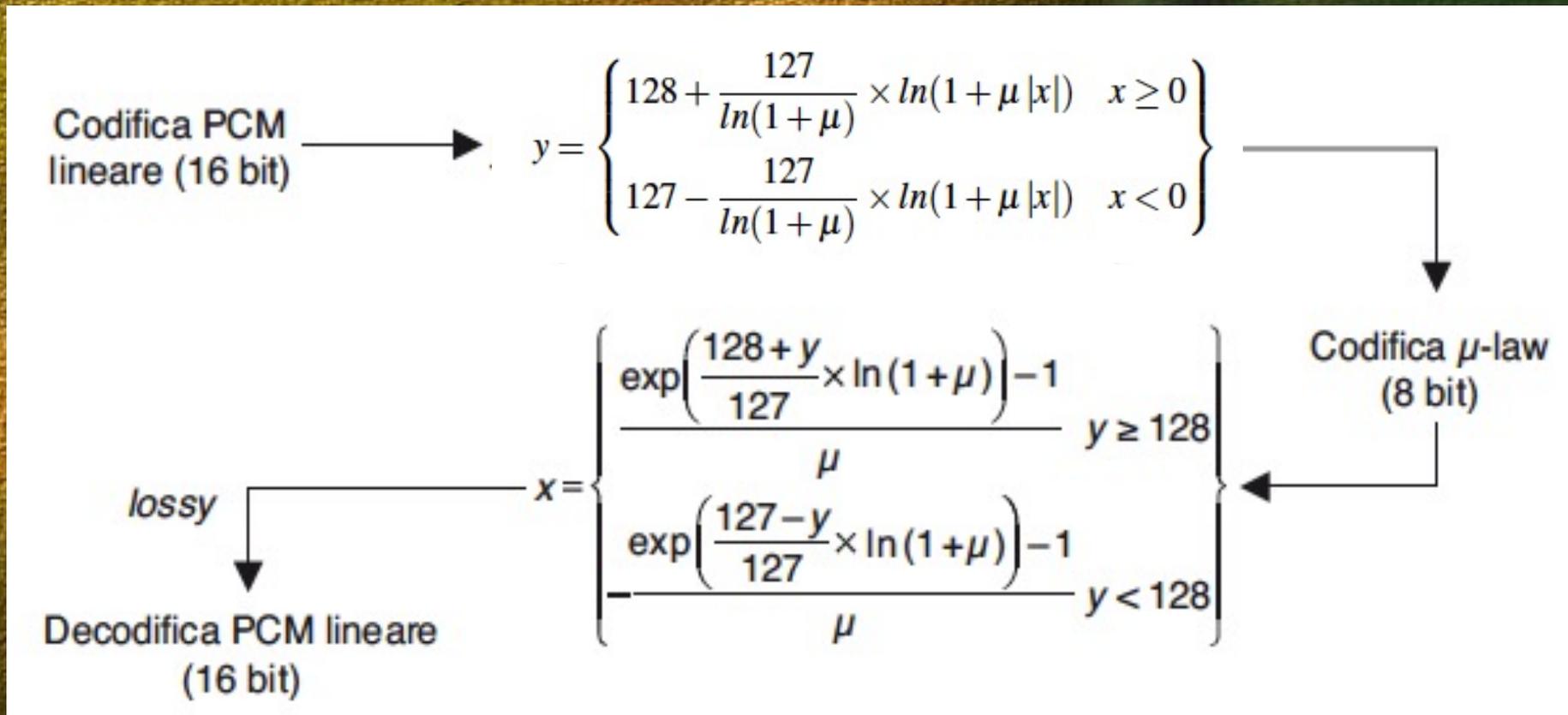


(a)



(b)

# Schema di codifica $\mu$ -law

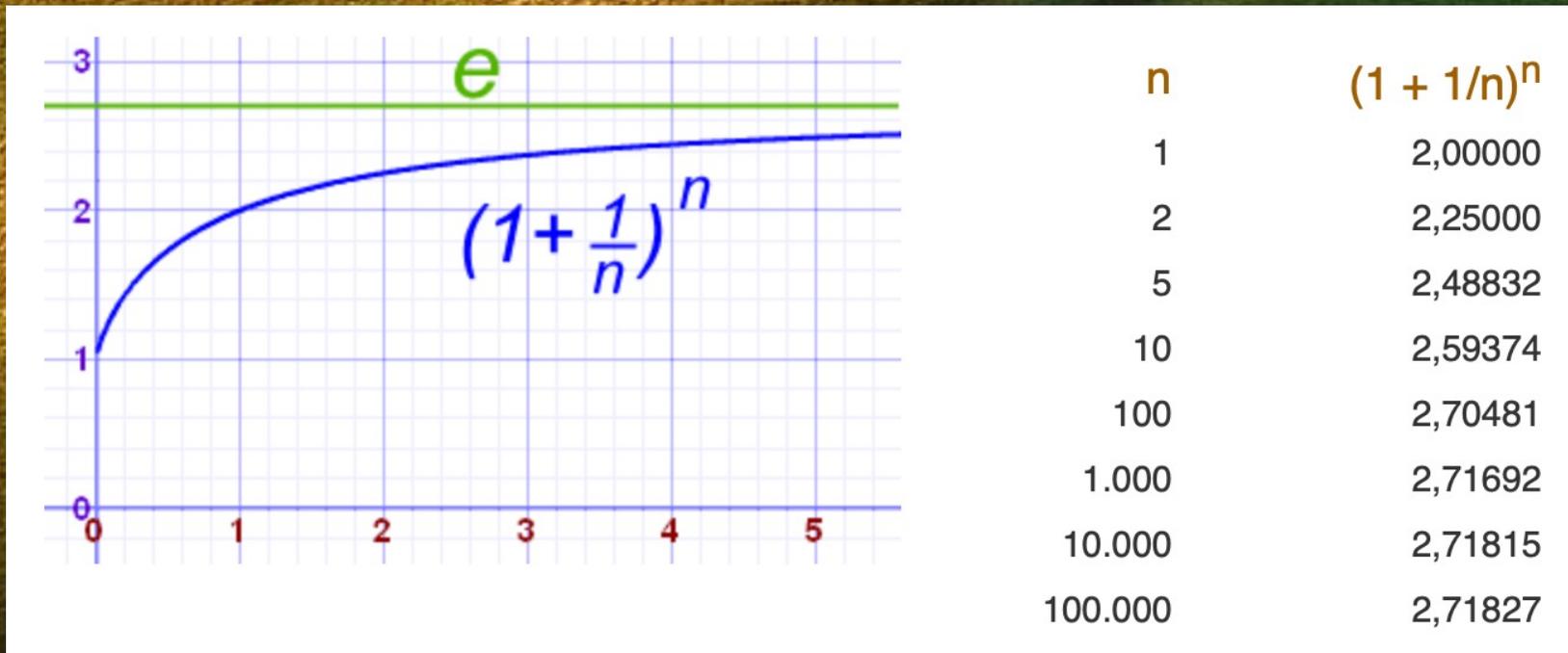


$\ln$  = logaritmo naturale, in base  $e = 2,7182818284590452353602874713527\dots$

$\exp$  = esponenziale naturale,  $e^n = (2,7182818284590452353602874713527\dots)^n$

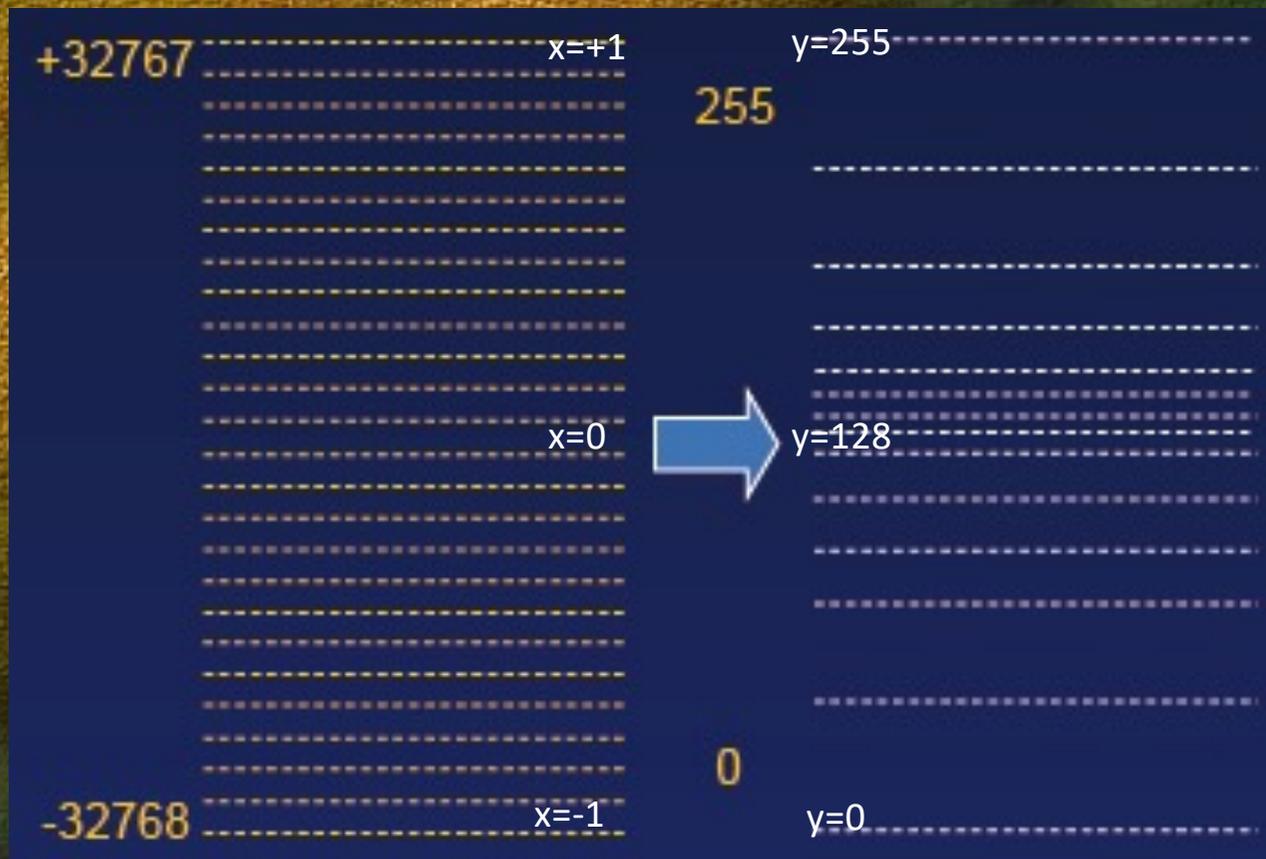
# Numero e

ln = logaritmo naturale, in base e = 2,7182818284590452353602874713527...



# Spiegazione intuitiva

$$y = \begin{cases} 128 + \frac{127}{\ln(1+\mu)} \times \ln(1+\mu|x|) & x \geq 0 \\ 127 - \frac{127}{\ln(1+\mu)} \times \ln(1+\mu|x|) & x < 0 \end{cases}$$



## Dettagli caso positivo

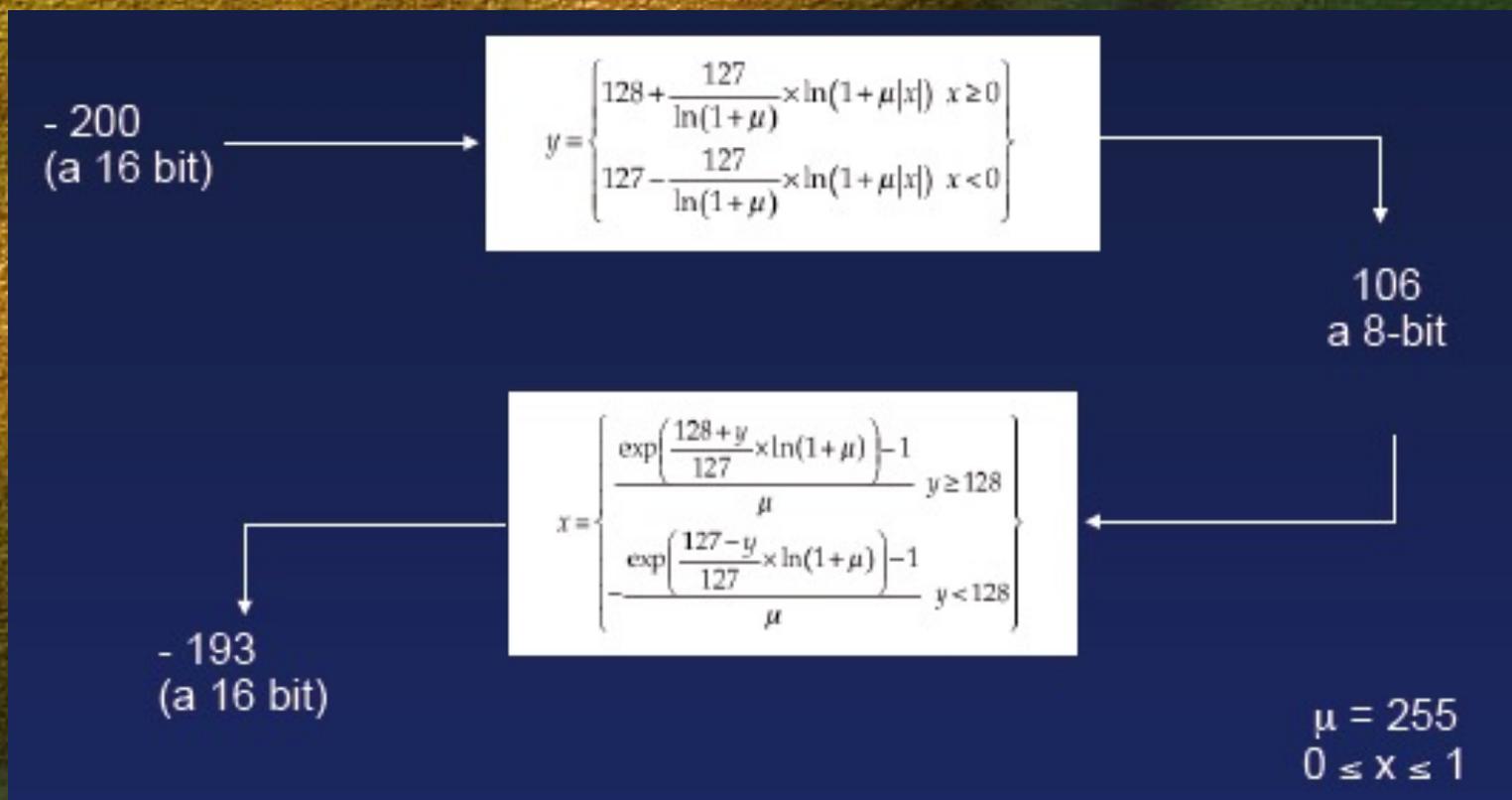
$$y = \begin{cases} 128 + \frac{127}{\ln(1+\mu)} \times \ln(1+\mu|x|) & x \geq 0 \\ 127 - \frac{127}{\ln(1+\mu)} \times \ln(1+\mu|x|) & x < 0 \end{cases}$$

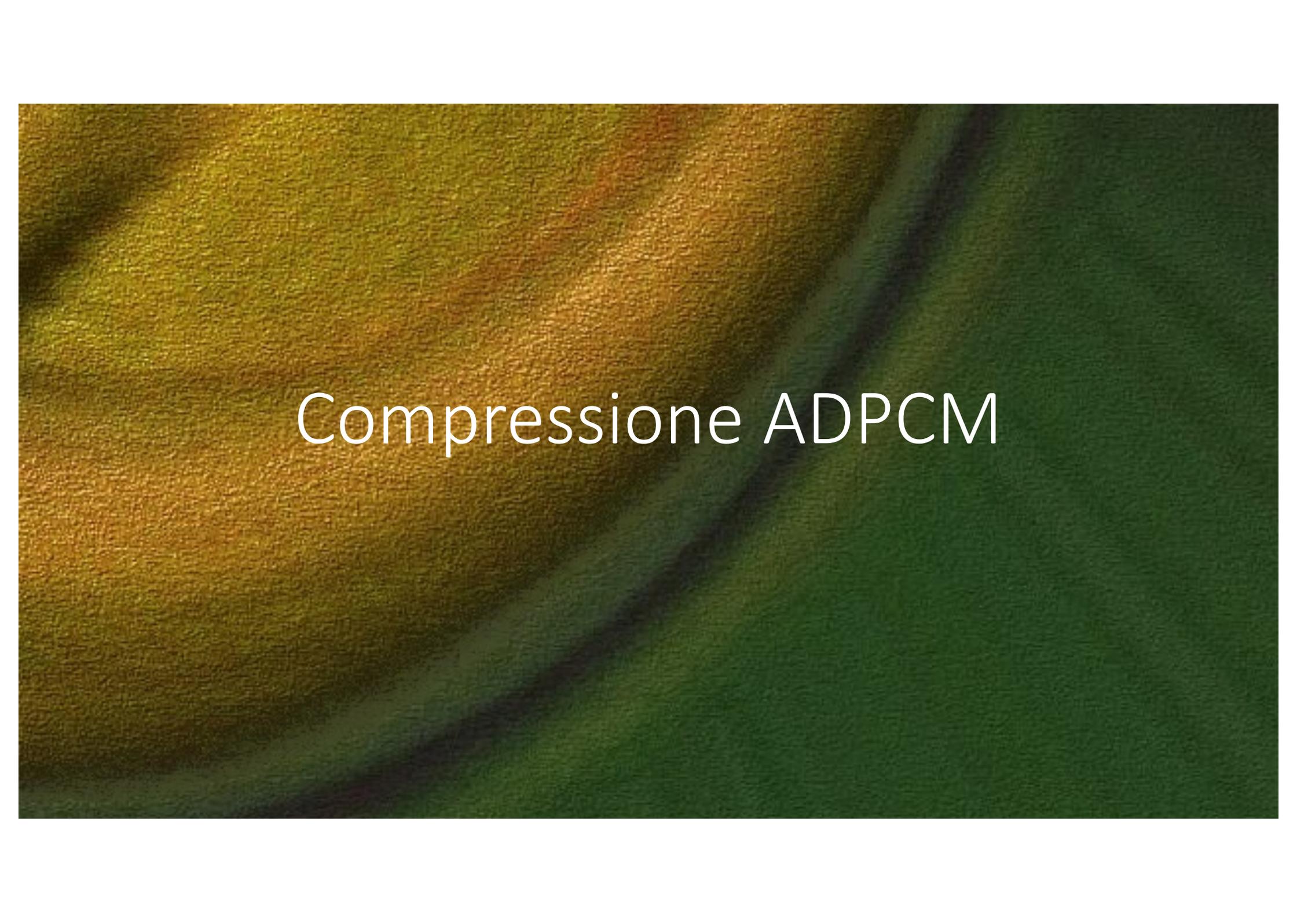
- Range variabili: x da 0 a 32767 (normalizzato da 0 a 1), y da 128 a 255
  - x=0 (silenzio SIGNED) → y=128 (silenzio UNSIGNED)
- Andamento logaritmico dei valori
  - A 32768 (=128x128) valori x, corrispondono 128 y, se fosse quantizzazione uniforme 128 x per 1 y
  - vicino al silenzio (x vicino a 0): pochi valori x (< 128) corrispondono a 1 y
  - lontano dal silenzio (x vicino a 32767): molti valori x (> 128) corrispondono a 1 y
  - Logaritmo  $\ln$ : logaritmo base e = 2,718281... =  $\sum(1+1/n)^n$
  - $0 \leq \ln(1+\mu|x|) / \ln(1+\mu) \leq 1$  dà una quota di 127, che si comporta in modo logaritmico

# Operazioni di conversione

	Campione originale	x in (-1,+1)	Nuovo campione
	-32768	-1	0
	...	...	...
$y = \begin{cases} 128 + \frac{127}{\ln(1+\mu)} \times \ln(1+\mu x ) & x \geq 0 \\ 127 - \frac{127}{\ln(1+\mu)} \times \ln(1+\mu x ) & x < 0 \end{cases}$	-32100	-0,9796142	0
	-32000	-0,9765625	1
	...	...	...
	-200	-0,00610351	106
	-100	-0,0030517	113
	...	...	...
	0	0	128
$x = \begin{cases} \frac{\exp\left(\frac{128+y}{127} \times \ln(1+\mu)\right) - 1}{\mu} & y \geq 128 \\ \frac{\exp\left(\frac{127-y}{127} \times \ln(1+\mu)\right) - 1}{\mu} & y < 128 \end{cases}$	...	...	...
	100	0,0030517	141
	200	0,00610351	149
	...	...	...
	32000	0,9765625	254
	32100	0,9796142	255
	...	...	...
	32767	1	255

# Lossy $\mu$ -law



The background of the slide features a diagonal gradient. The upper-left portion is a bright, shimmering gold, which transitions through a dark, almost black diagonal band to a deep, forest green on the lower-right. The overall texture is slightly grainy, suggesting a digital or printed surface.

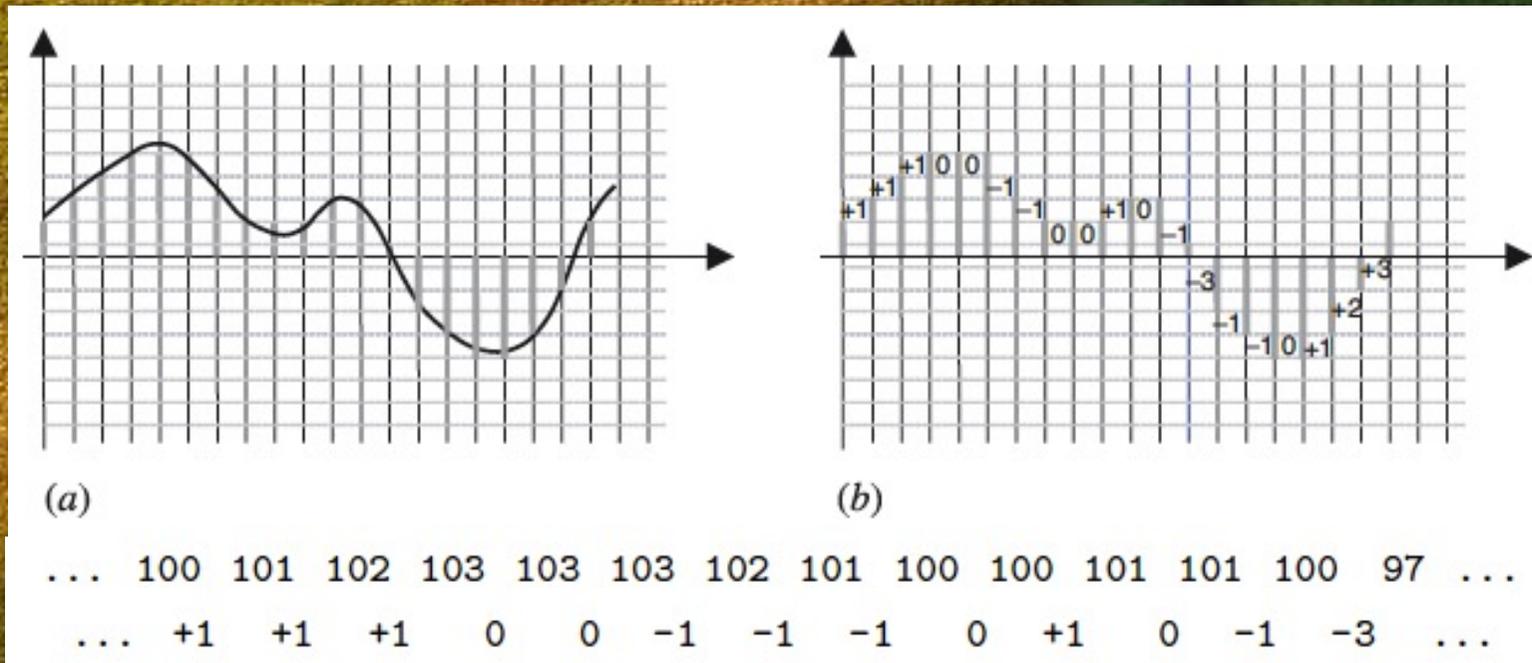
# Compression ADPCM

# ADPCM = Adaptive Differential PCM

## CCITT G.721, G.723, e ITU-T G.726

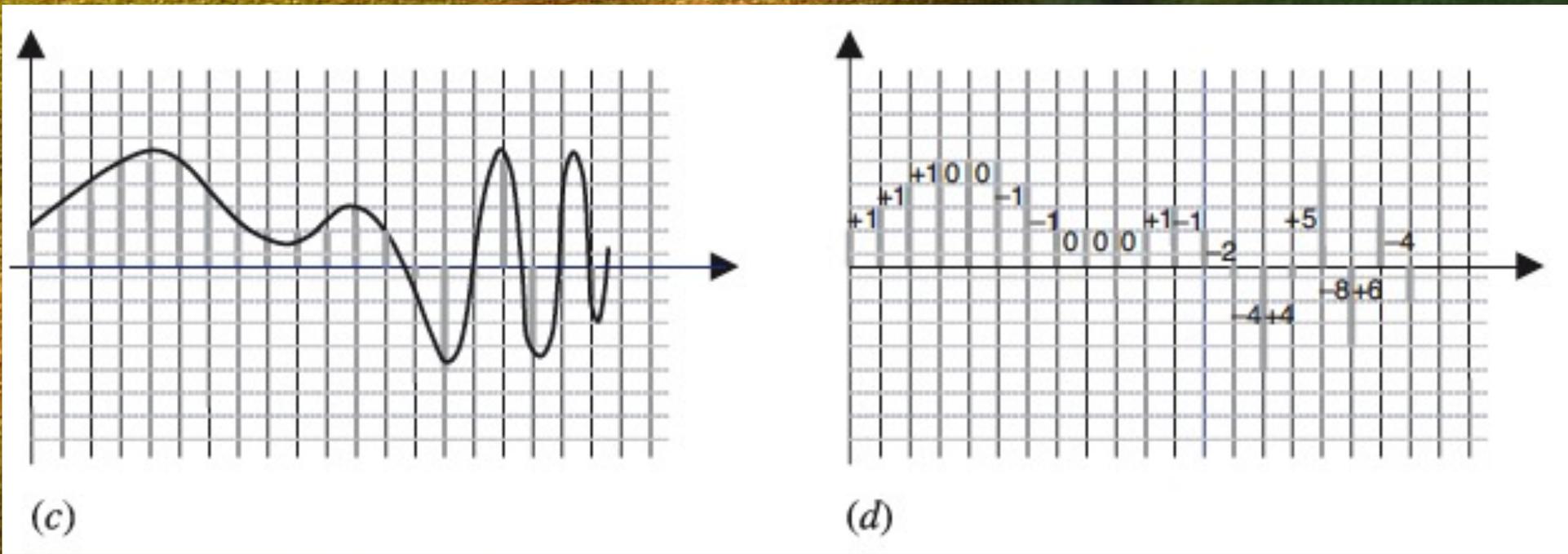
- Standard per la compressione di dati vocali
  - CCITT G.721 (ADPCM a 32 Kbps)
  - CCITT G.723 (ADPCM a 24 e 40 Kbps)
- metodo comune di compressione
- buon compromesso tra
  - velocità di elaborazione
  - tasso di compressione
  - decodifica di qualità

# Codifica della differenza (DPCM)



- Ridondanze temporali tra i campioni
- Differenza tra due campioni a x-bit con meno di x bit
- Si memorizza la differenza (non il campione)

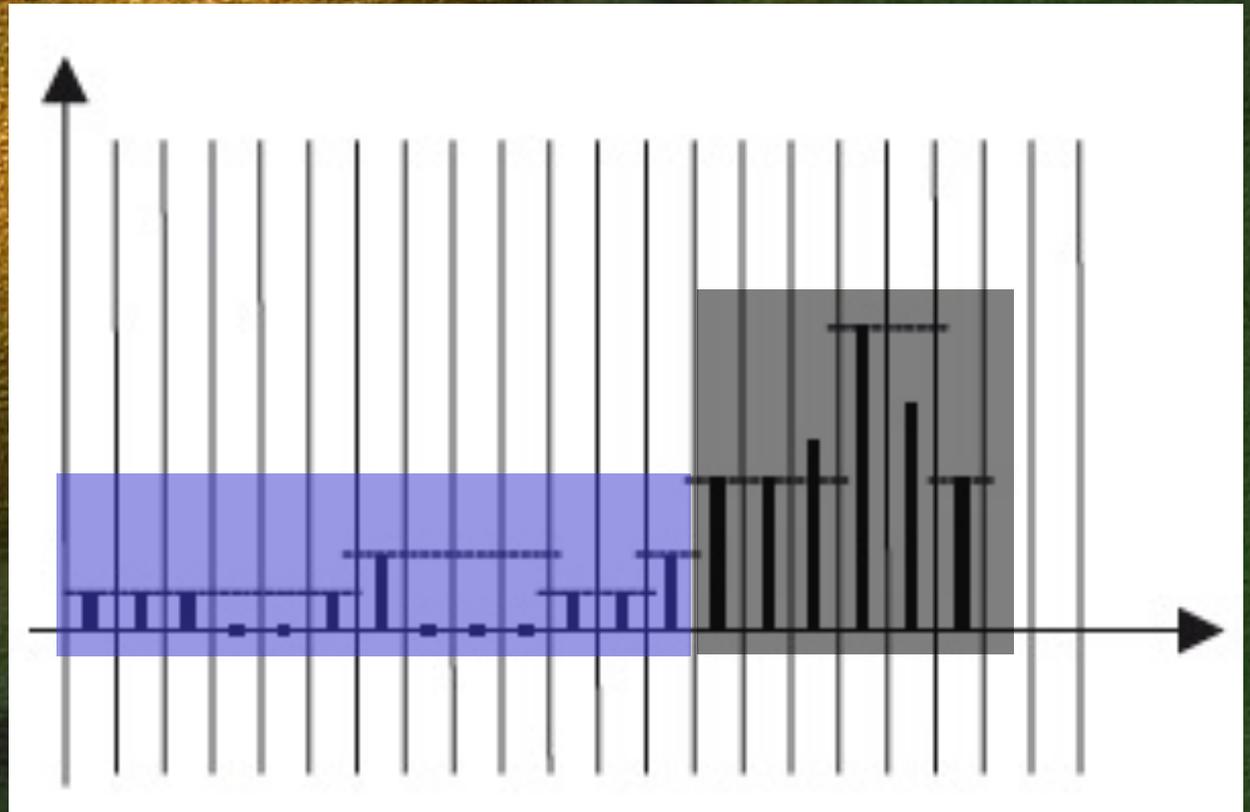
# Problema Slope Overload



- Differenze elevate (alte frequenze) non si possono rappresentare con pochi bit
- Errori introdotti porterebbero distorsioni sulle alte frequenze

# Adaptive PCM (ADPCM)

- Differenze grandi
- Passi di quantizzazione grandi
- Differenze piccole
- Passi di quantizzazione piccoli

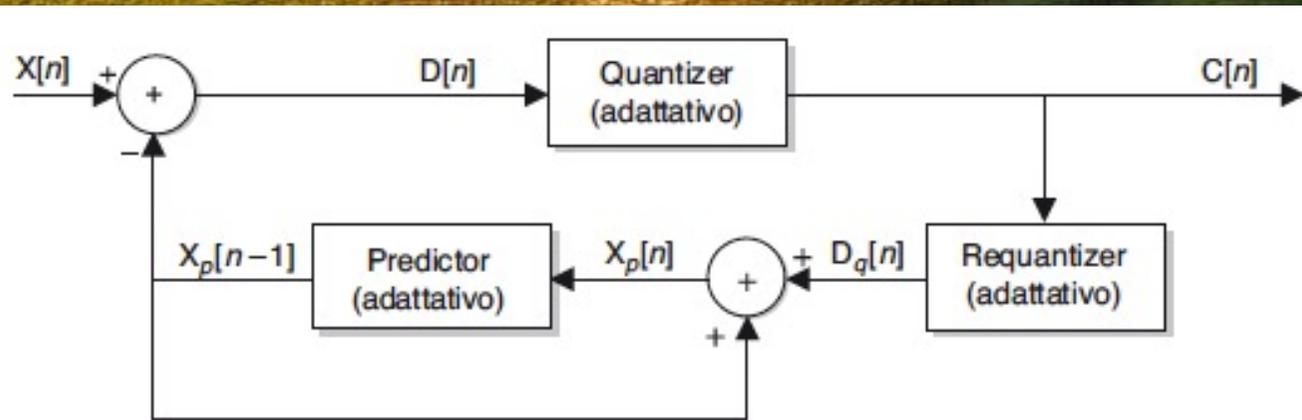


# La predizione

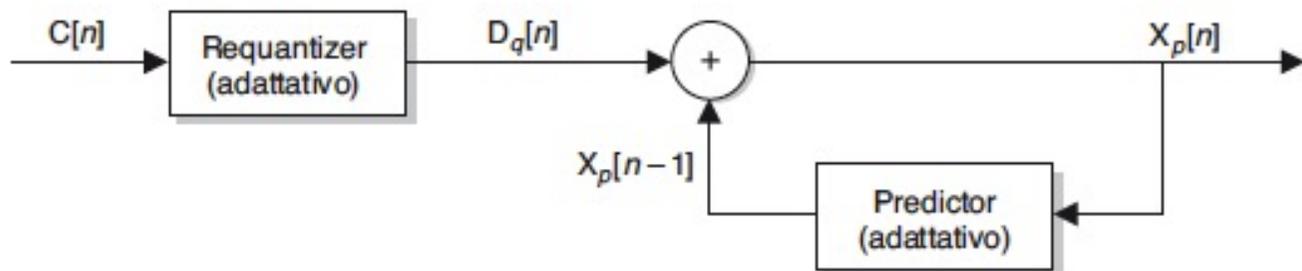
- campione precedente usato come base di partenza
- si sfrutta la conoscenza della porzione precedente del segnale
- Esempio: +1 e -1

```
... 100 101 102 103 103 103 102 101 100 100 101 101 100 97 ...  
... 100 101 102 103 104 103 102 101 100 101 100 101 100 99 ...  
... 0 0 0 0 -1 0 0 0 0 -1 +1 0 0 -2 ...
```

# Schema ADPCM



(a)



(b)

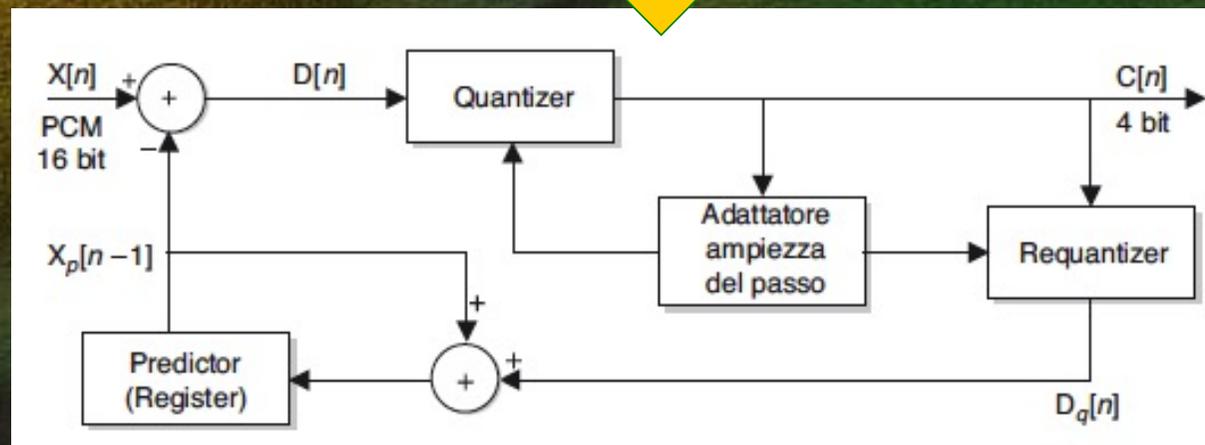
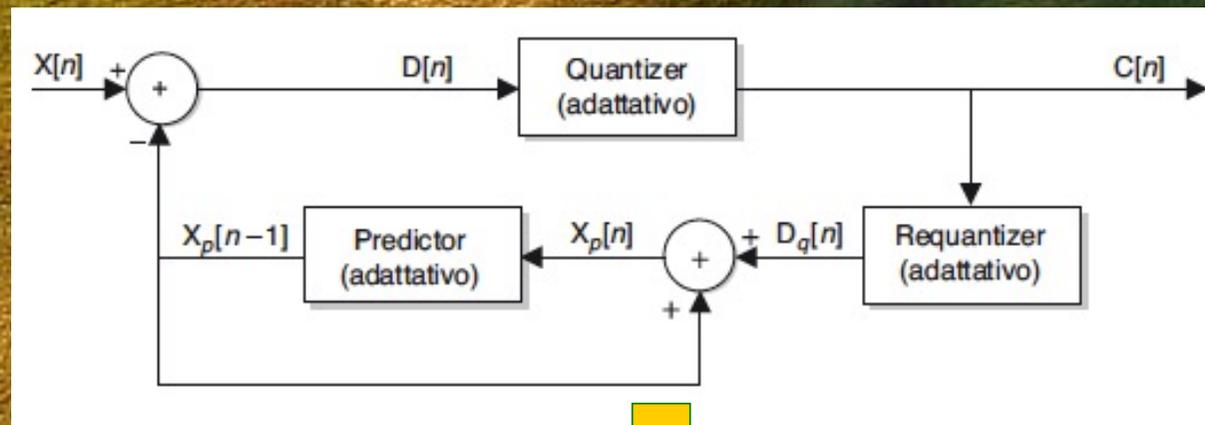
# Funzionalità aggiuntive

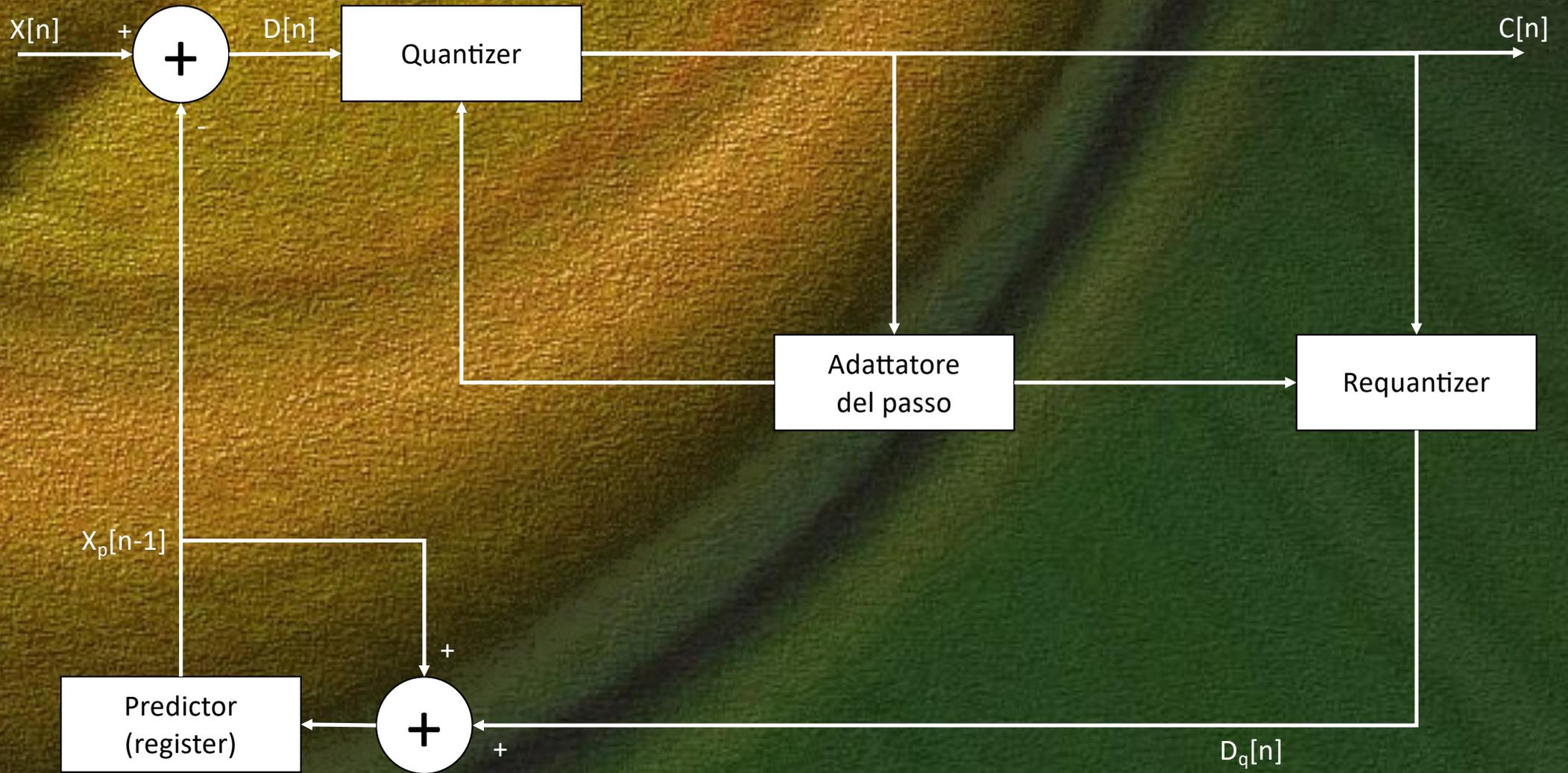
- Formattazione del bitstream  $C[n]$ 
  - Utile nel recupero degli errori di decodifica
  - Problema propagazione dell'errore
  - Ridistribuzione pseudocasuale della sequenza
- Informazioni aggiuntive
  - indicare i livelli di quant. di Predictor e/o Quantizer
  - fornire informazioni contestuali al decoder per recupero errori o accesso diretto al bitstream

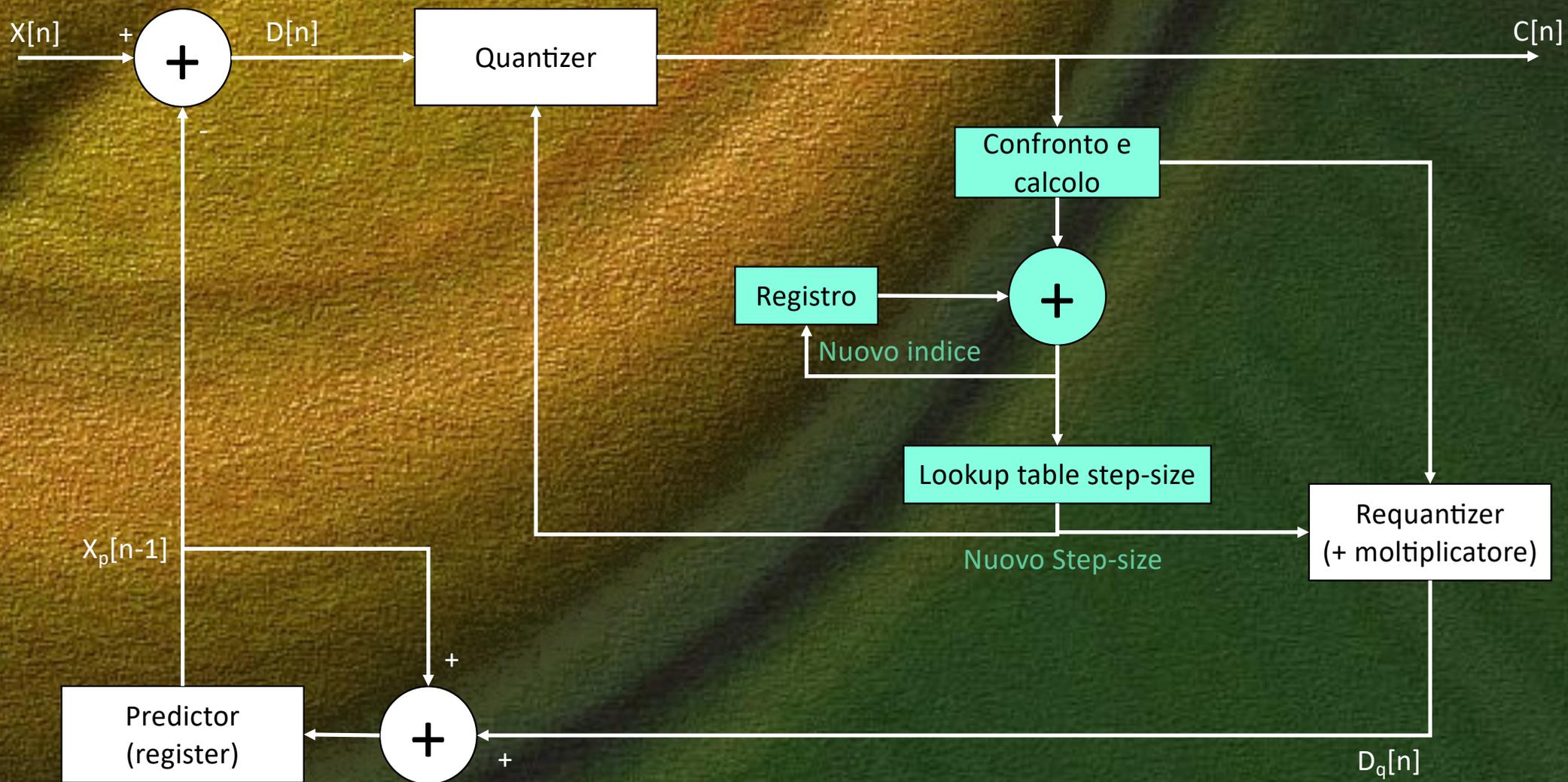
# IMA ADPCM

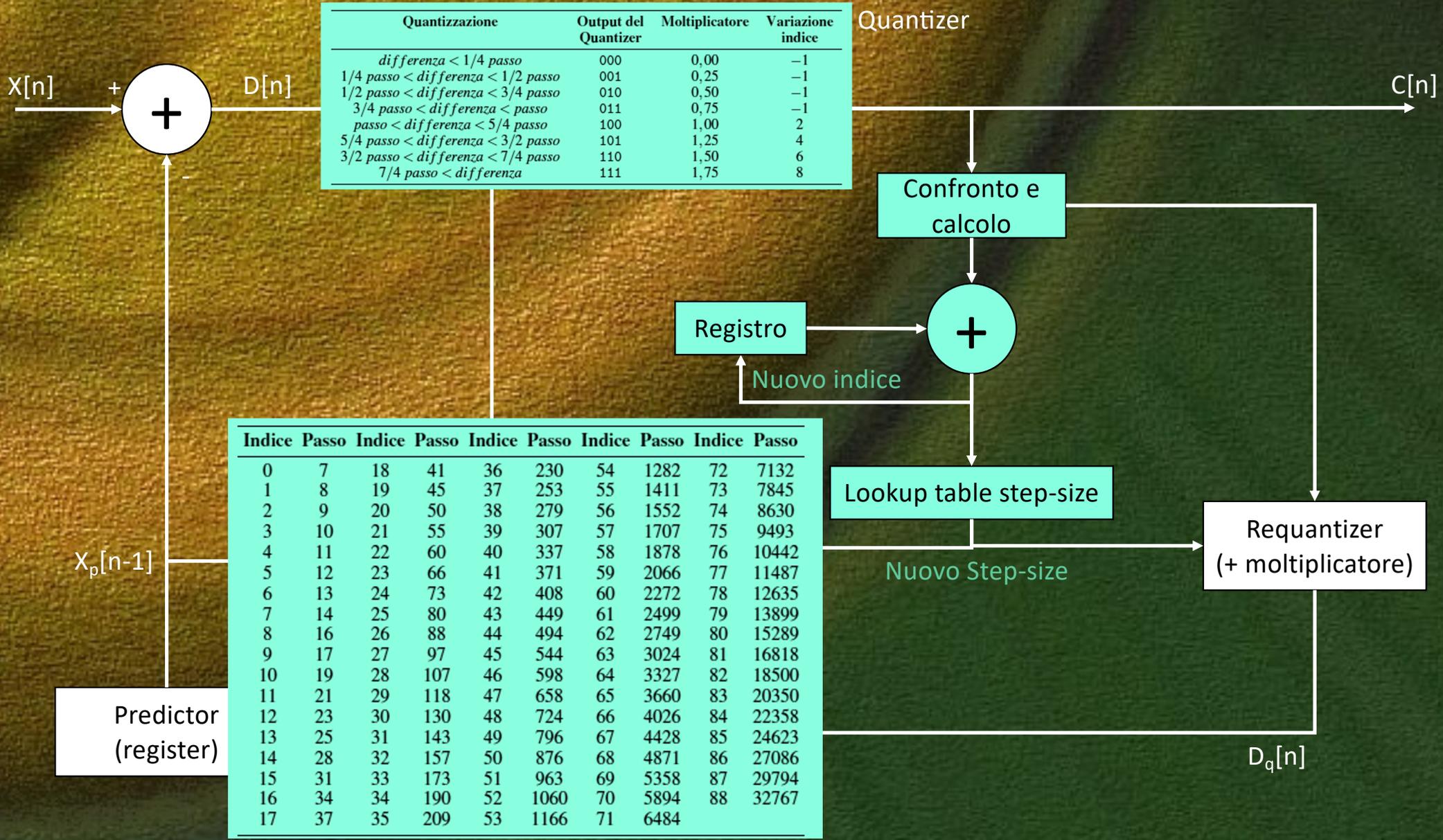
- Predictor non adattativo
- Passo di quantizzazione aumenta in modo logaritmico
- Variante IMA/DVI ADPCM: comprime da 16-bit a 4 bit
- Variante Microsoft ADPCM: da usare con i file WAV (da 16 a 4 bit)

# Schema IMA





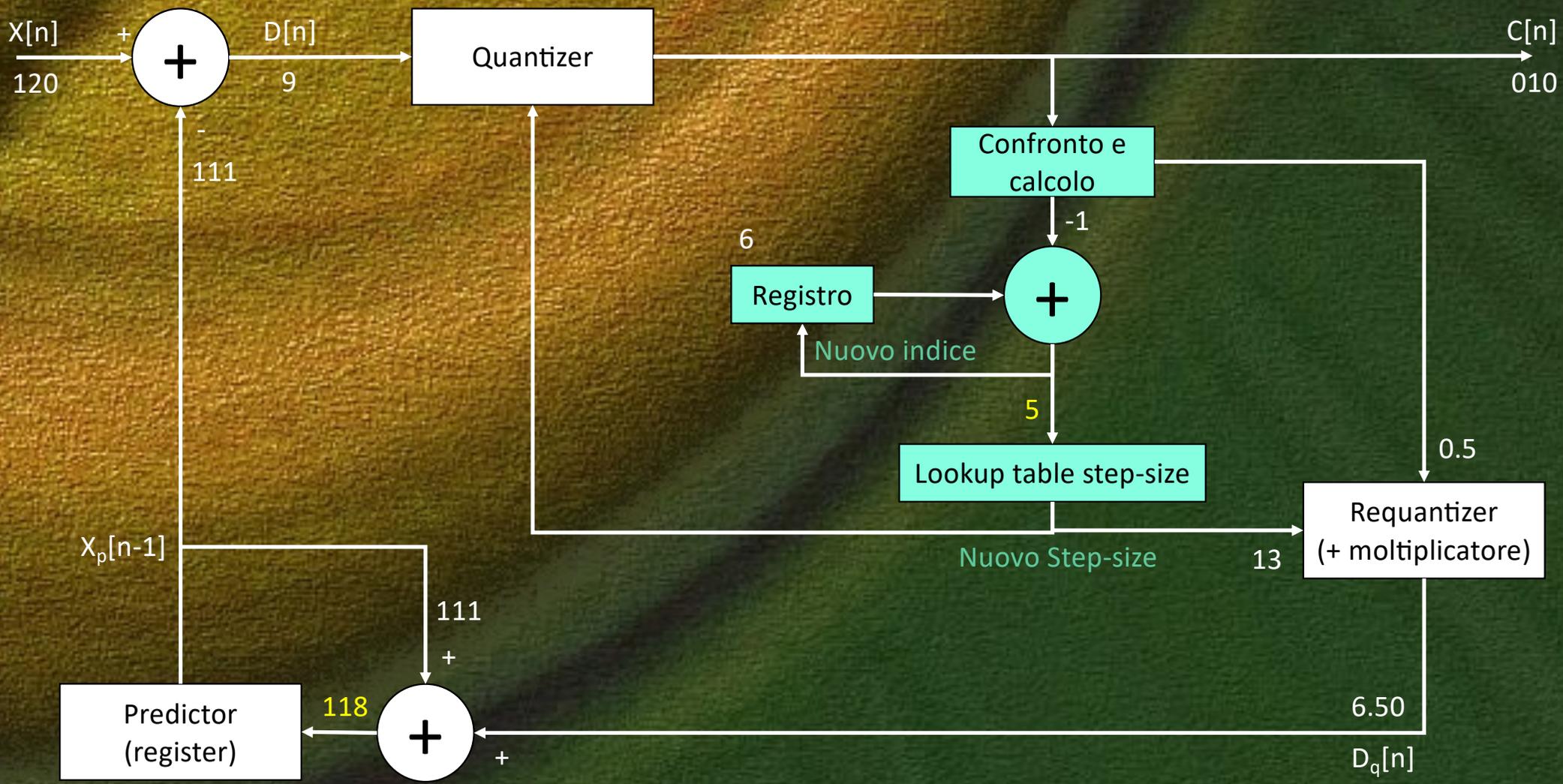




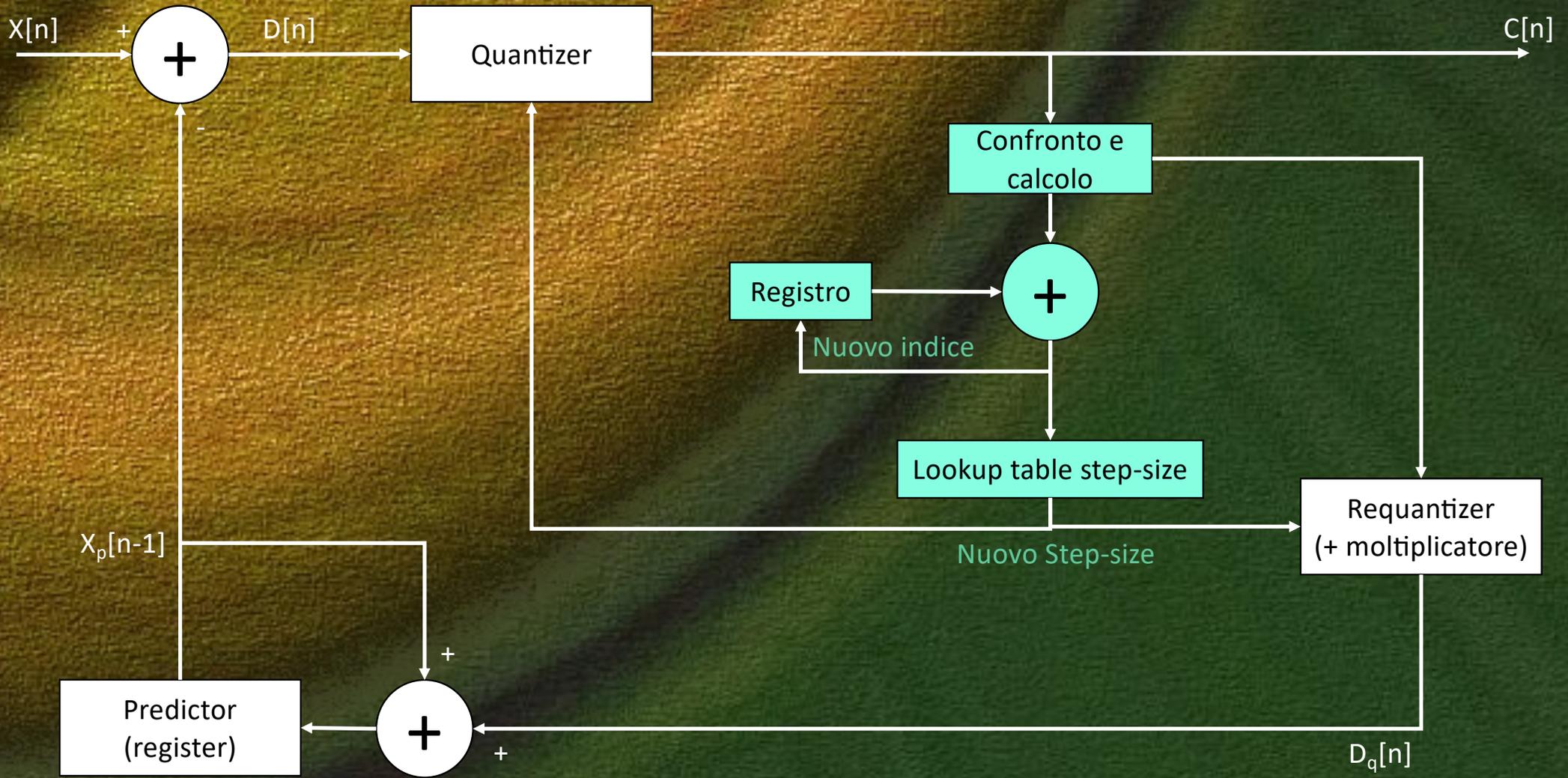
Quantizzazione	Output del Quantizer	Moltiplicatore	Variazione indice
$diff erenza < 1/4$ passo	000	0,00	-1
$1/4$ passo < $diff erenza < 1/2$ passo	001	0,25	-1
$1/2$ passo < $diff erenza < 3/4$ passo	010	0,50	-1
$3/4$ passo < $diff erenza <$ passo	011	0,75	-1
passo < $diff erenza < 5/4$ passo	100	1,00	2
$5/4$ passo < $diff erenza < 3/2$ passo	101	1,25	4
$3/2$ passo < $diff erenza < 7/4$ passo	110	1,50	6
$7/4$ passo < $diff erenza$	111	1,75	8

Indice	Passo								
0	7	18	41	36	230	54	1282	72	7132
1	8	19	45	37	253	55	1411	73	7845
2	9	20	50	38	279	56	1552	74	8630
3	10	21	55	39	307	57	1707	75	9493
4	11	22	60	40	337	58	1878	76	10442
5	12	23	66	41	371	59	2066	77	11487
6	13	24	73	42	408	60	2272	78	12635
7	14	25	80	43	449	61	2499	79	13899
8	16	26	88	44	494	62	2749	80	15289
9	17	27	97	45	544	63	3024	81	16818
10	19	28	107	46	598	64	3327	82	18500
11	21	29	118	47	658	65	3660	83	20350
12	23	30	130	48	724	66	4026	84	22358
13	25	31	143	49	796	67	4428	85	24623
14	28	32	157	50	876	68	4871	86	27086
15	31	33	173	51	963	69	5358	87	29794
16	34	34	190	52	1060	70	5894	88	32767
17	37	35	209	53	1166	71	6484		

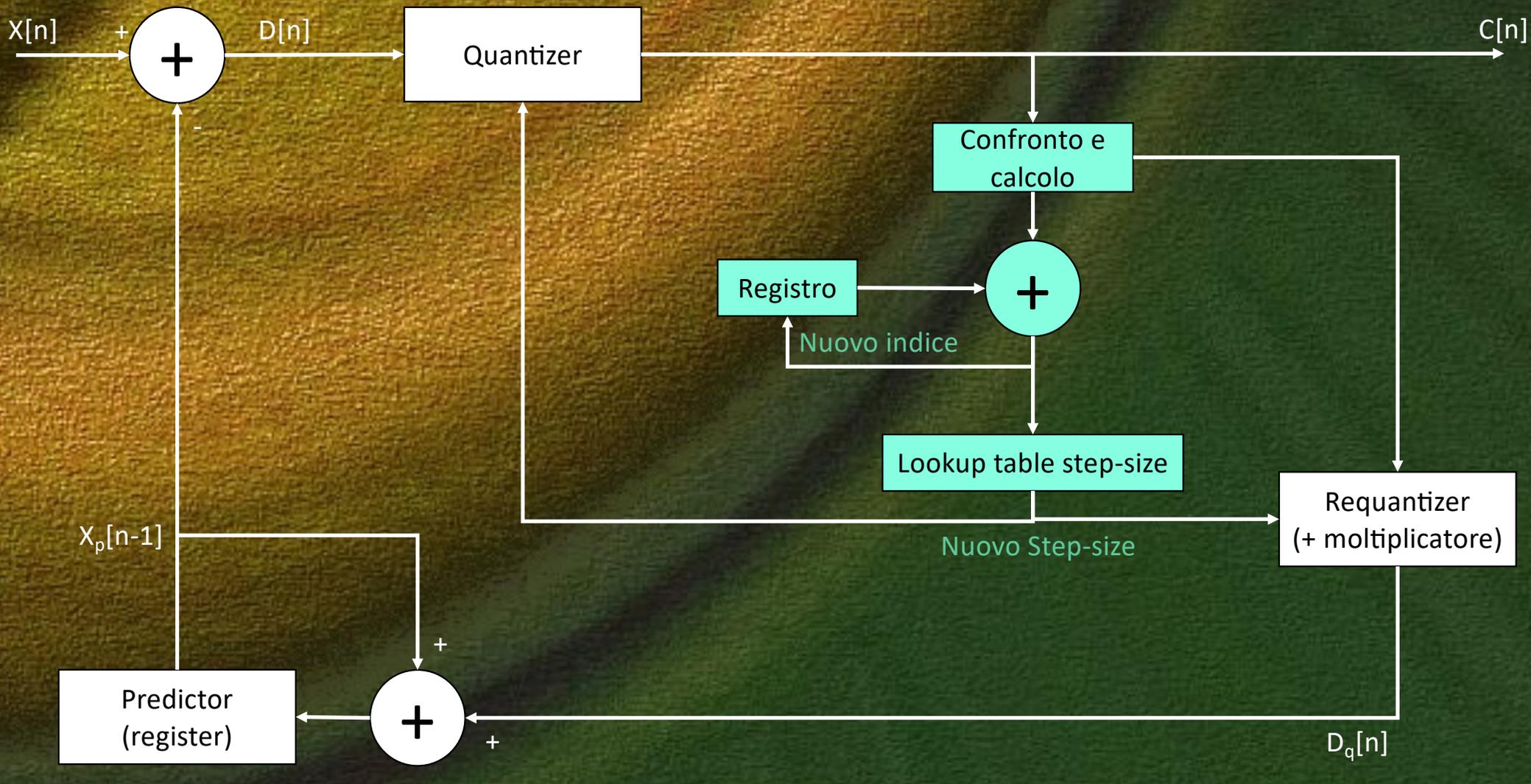
2	112	12	0/7	$3/2 \times 7 < 12 < 7/4 \times 7$	110	1,5	6	6	10,50	111
3	120	9	6/13	$1/2 \times 13 < 9 < 3/4 \times 13$	010	0,5	-1	5	6,50	118
4	127	9	5/12	$3/4 \times 12 < 9 < 12$	011	0,75	-1	4	9,00	127



3	120	9	6/13	$1/2 \times 13 < 9 < 3/4 \times 13$	010	0,5	-1	5	6,50	118
4	127	9	5/12	$3/4 \times 12 < 9 < 12$	011	0,75	-1	4	9,00	127
5	131	4	4/11	$1/4 \times 11 < 4 < 1/2 \times 11$	001	0,25	-1	3	2,75	130



13	99	-14	4/11	$5/4 \times 11 < 12 < 3/2 \times 11$	101	1,25	4	8	-13,75	99
14	54	-45	8/16	$7/4 \times 16 < 45$	111	1,75	8	16	-28,00	71
15	20	-51	16/34	$3/2 \times 34 < 51 < 7/4 \times 34$	110	1,5	6	22	-51,00	20



# Tabella dei passi di quantizzazione

Indice	Passo								
0	7	18	41	36	230	54	1282	72	7132
1	8	19	45	37	253	55	1411	73	7845
2	9	20	50	38	279	56	1552	74	8630
3	10	21	55	39	307	57	1707	75	9493
4	11	22	60	40	337	58	1878	76	10442
5	12	23	66	41	371	59	2066	77	11487
6	13	24	73	42	408	60	2272	78	12635
7	14	25	80	43	449	61	2499	79	13899
8	16	26	88	44	494	62	2749	80	15289
9	17	27	97	45	544	63	3024	81	16818
10	19	28	107	46	598	64	3327	82	18500
11	21	29	118	47	658	65	3660	83	20350
12	23	30	130	48	724	66	4026	84	22358
13	25	31	143	49	796	67	4428	85	24623
14	28	32	157	50	876	68	4871	86	27086
15	31	33	173	51	963	69	5358	87	29794
16	34	34	190	52	1060	70	5894	88	32767
17	37	35	209	53	1166	71	6484		

# Adattatore dell'ampiezza del passo

Quantizzazione	Output del Quantizer	Moltiplicatore	Variazione indice
<i>differenza &lt; 1/4 passo</i>	000	0,00	-1
<i>1/4 passo &lt; differenza &lt; 1/2 passo</i>	001	0,25	-1
<i>1/2 passo &lt; differenza &lt; 3/4 passo</i>	010	0,50	-1
<i>3/4 passo &lt; differenza &lt; passo</i>	011	0,75	-1
<i>passo &lt; differenza &lt; 5/4 passo</i>	100	1,00	2
<i>5/4 passo &lt; differenza &lt; 3/2 passo</i>	101	1,25	4
<i>3/2 passo &lt; differenza &lt; 7/4 passo</i>	110	1,50	6
<i>7/4 passo &lt; differenza</i>	111	1,75	8



# IMA ADPCM: riassunto

- Fattore di compressione di bit-per-campione:
  - 4 a 1 (da 16 a 4 bit)
  - bit-rate < 400 kbps
- Passi di quantizzazione superiori e bit-rate ridotti:
  - CCITT G.721: 16 o 32 kbps
  - CCITT G.723: 24 kbps



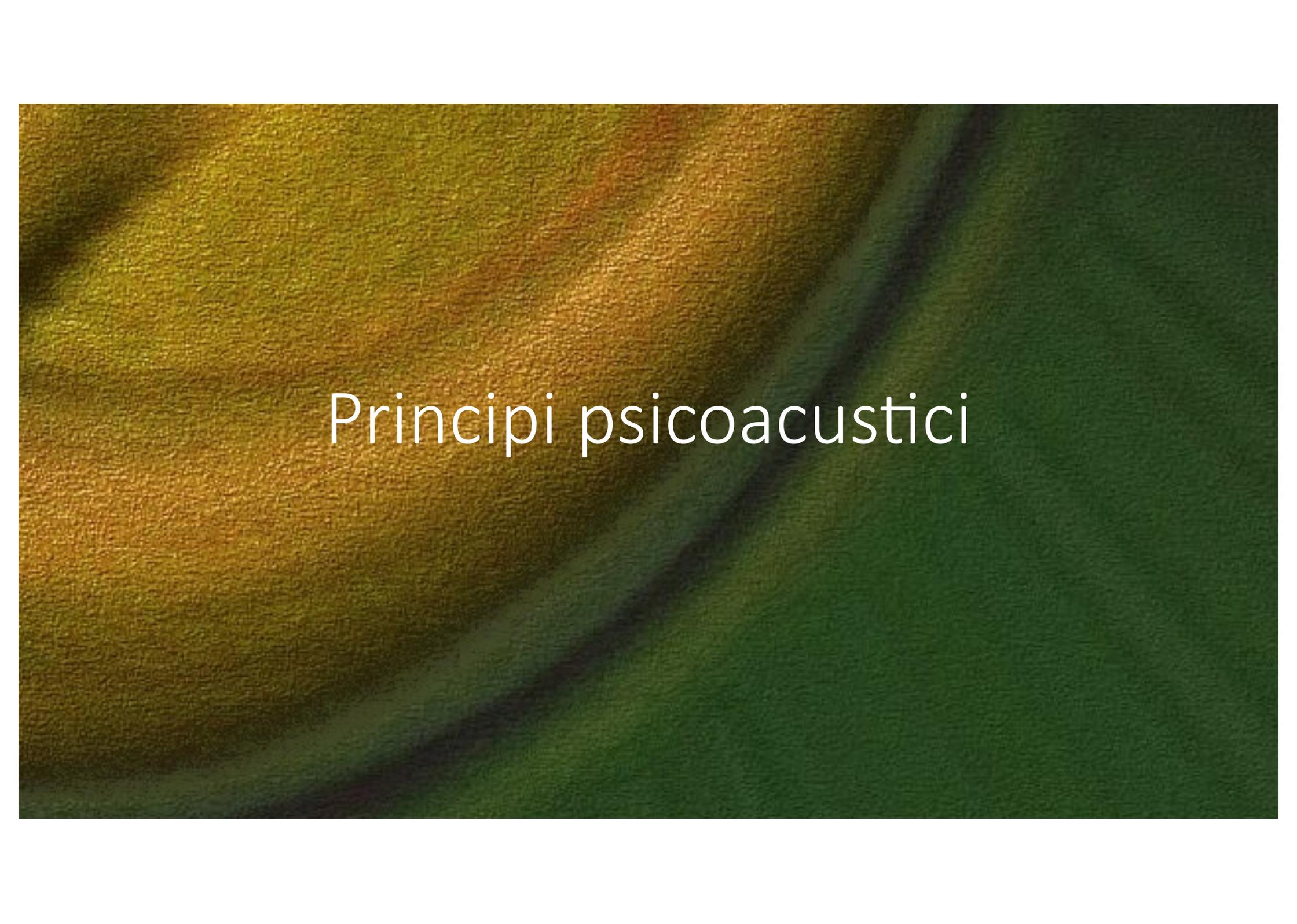
# La compressione di tipo percettivo

MP3 & Co.

Principi psicoacustici applicati

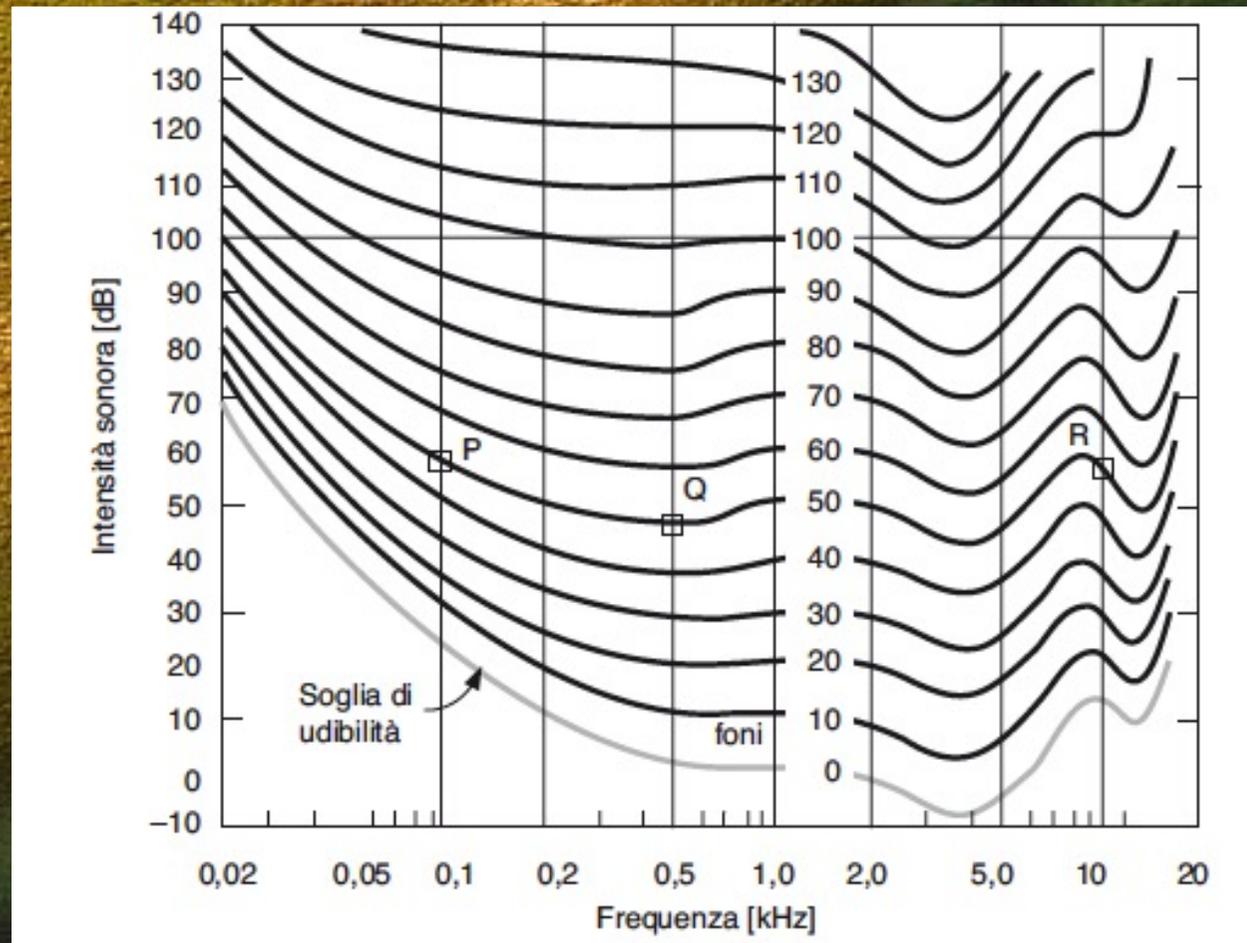
# Perché gli schemi percettivi

- Schemi semplici non producono rilevanti risparmi di spazio e tempo (peggioramento qualità)
  - Eliminare parti del segnale che il nostro apparato uditivo non percepirebbe
  - Codifiche di tipo lossy (con perdite), bit-rate molto bassi (0,5 bit/campione sul parlato)
- 
- Dibattito sulla qualità

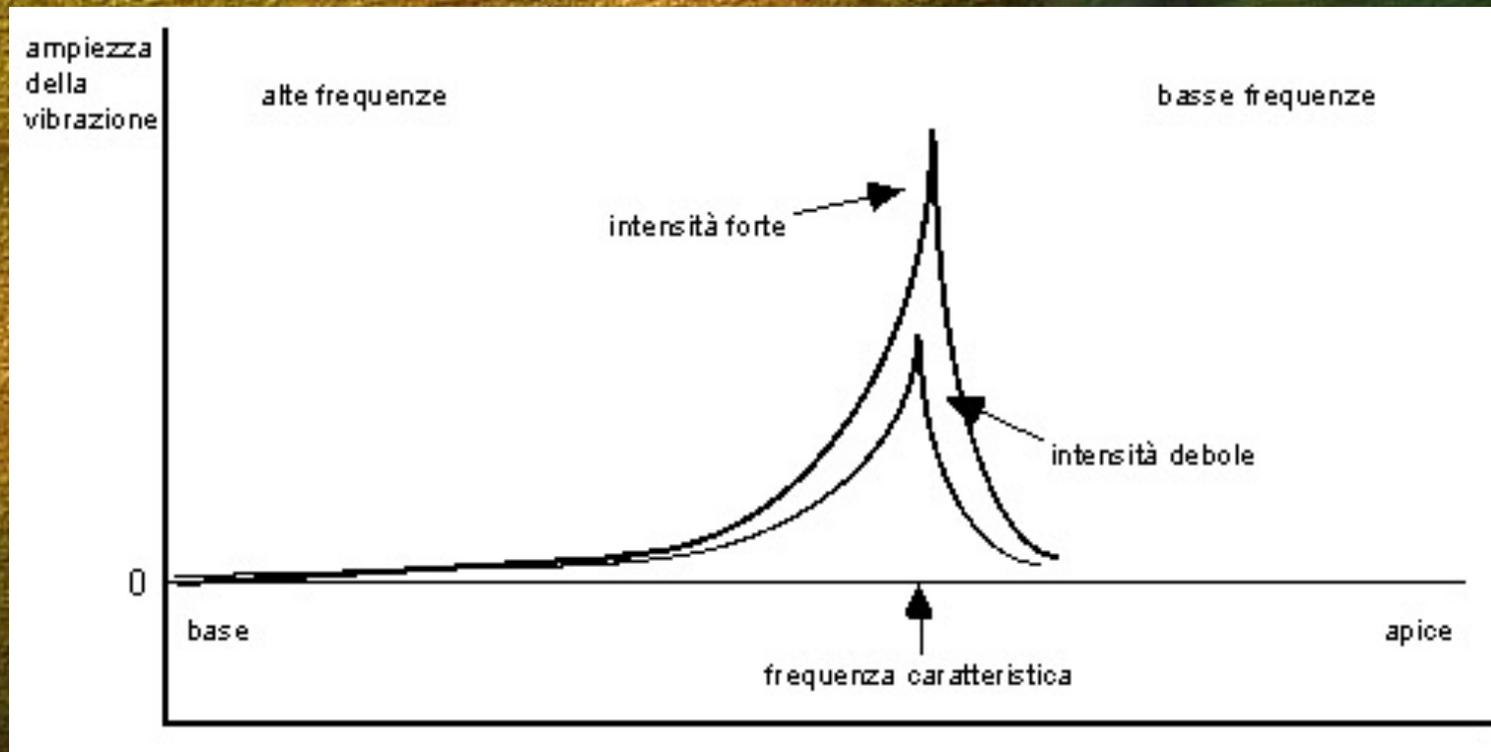


# Principi psicoacustici

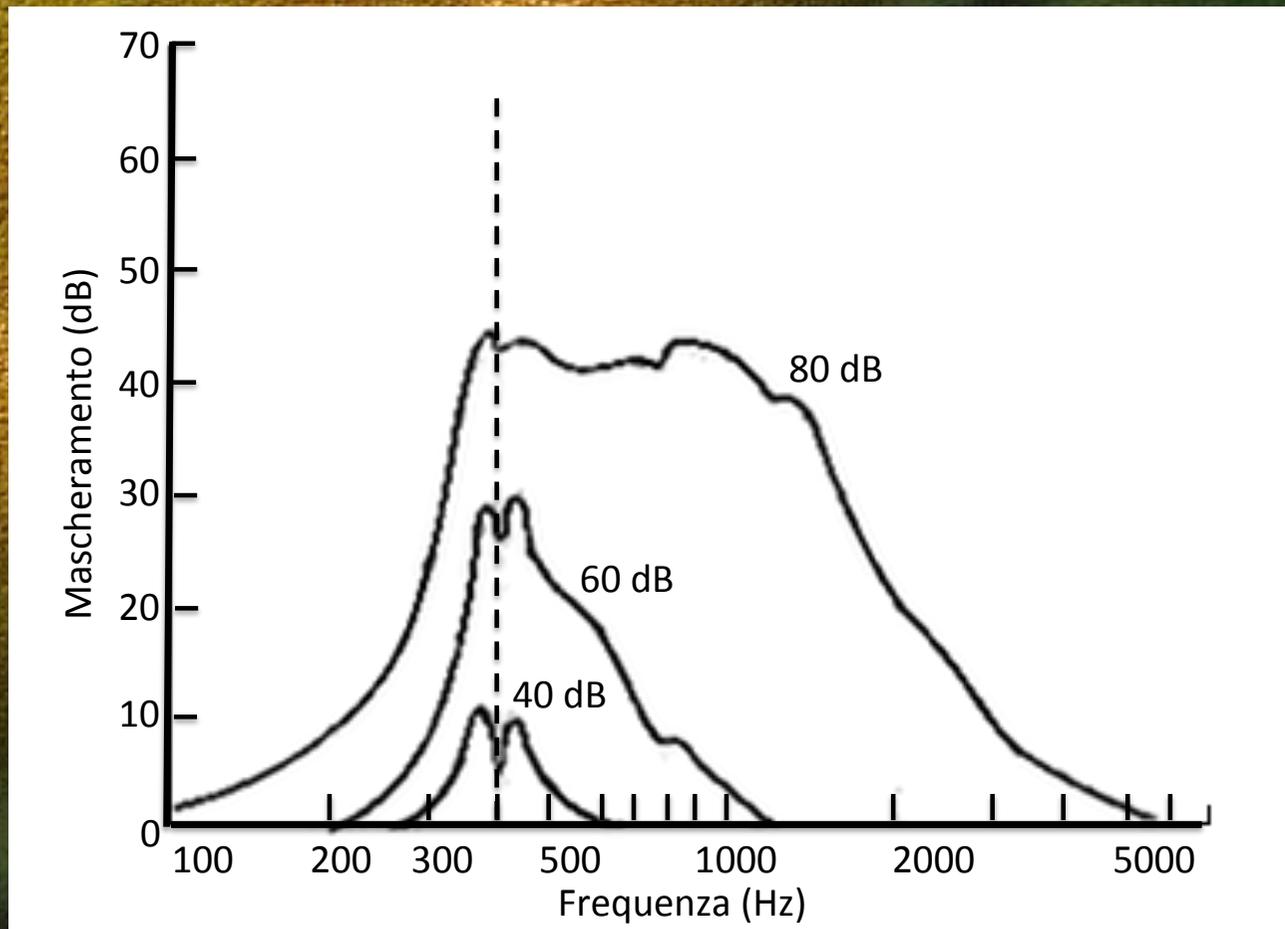
# Soglia assoluta di udibilità



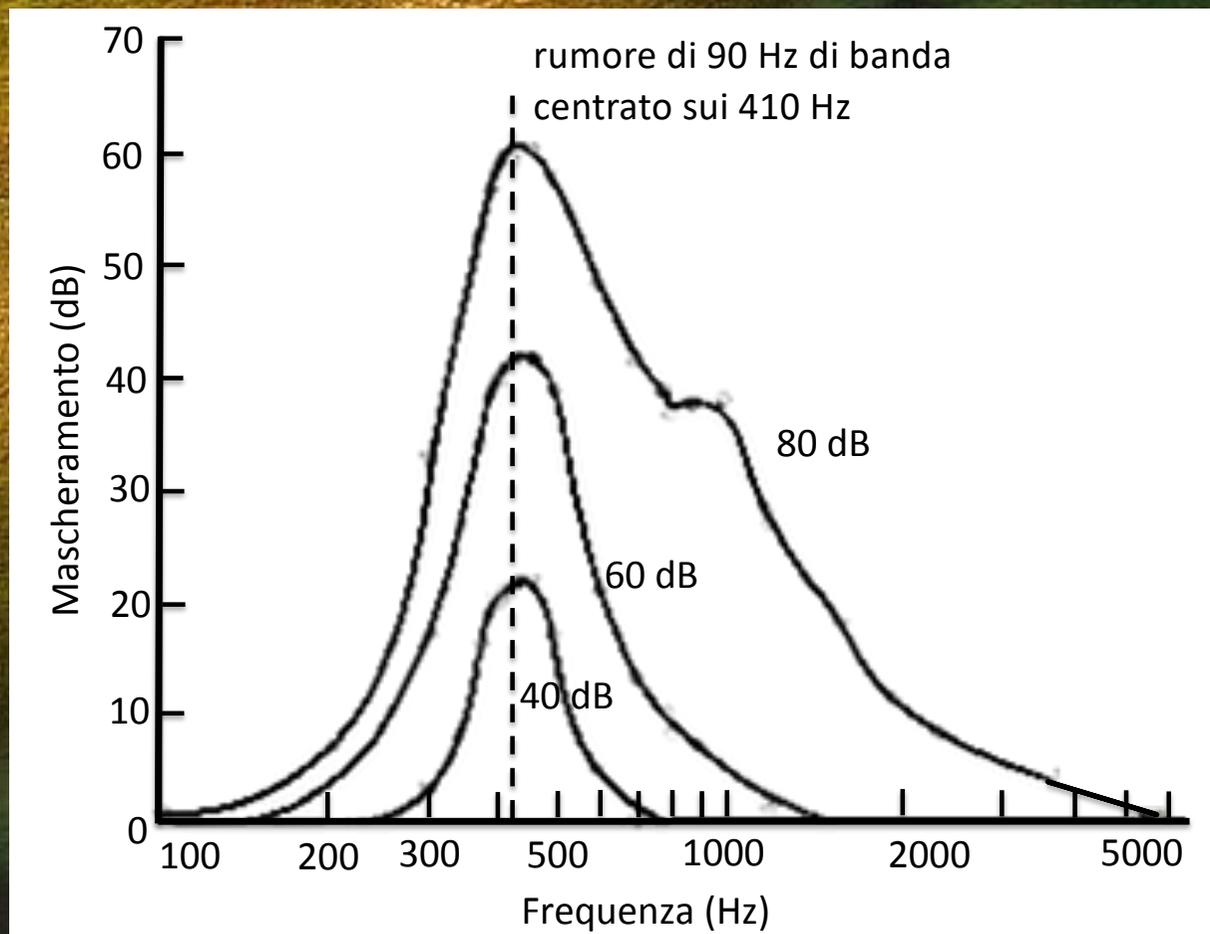
# Mascheramento



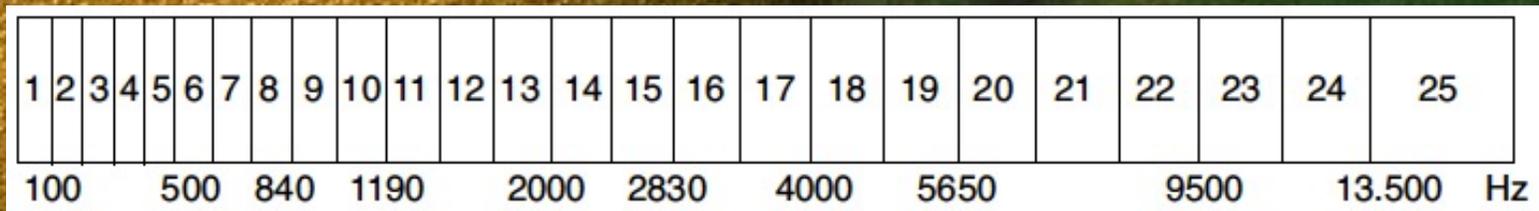
# Il mascheramento tonale



# Il mascheramento non tonale



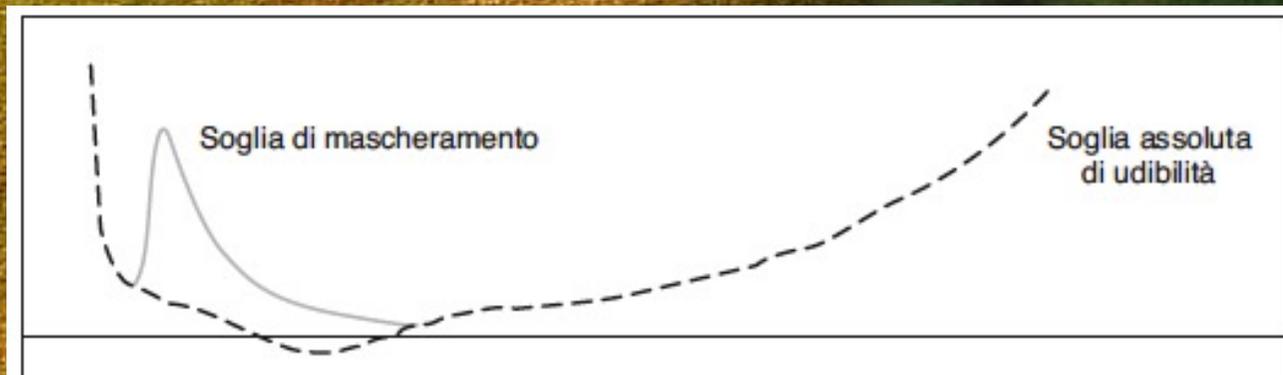
# Bande critiche



# Bande critiche di Fletcher

Numero di banda	Centro della banda	Estremi della banda e estensione
1	60	fino a 100 ( 80)
2	150	100-200 (100)
3	250	200-300 (100)
4	350	300-400 (100)
5	450	400-500 (100)
6	550	500-600 (100)
7	655	600-710 (110)
8	775	710-840 (130)
9	920	840-1000 (160)
10	1095	1000-1190 (190)
11	1300	1190-1410 (230)
12	1545	1410-1680 (270)
13	1840	1680-2000 (320)
14	2190	2000-2380 (380)
15	2605	2380-2830 (450)
16	3095	2830-3360 (530)
17	3680	3360-4000 (640)
18	4380	4000-4760 (760)
19	5205	4760-5650 (890)
20	6175	5650-6720 (1.050)
21	7360	6720-8000 (1.280)
22	8750	8000-9500 (1.500)
23	10.400	9500-11.300 (1.800)
24	12.400	11.300-13.500 (2.200)
25	16.700	da 13.500 (6.500)

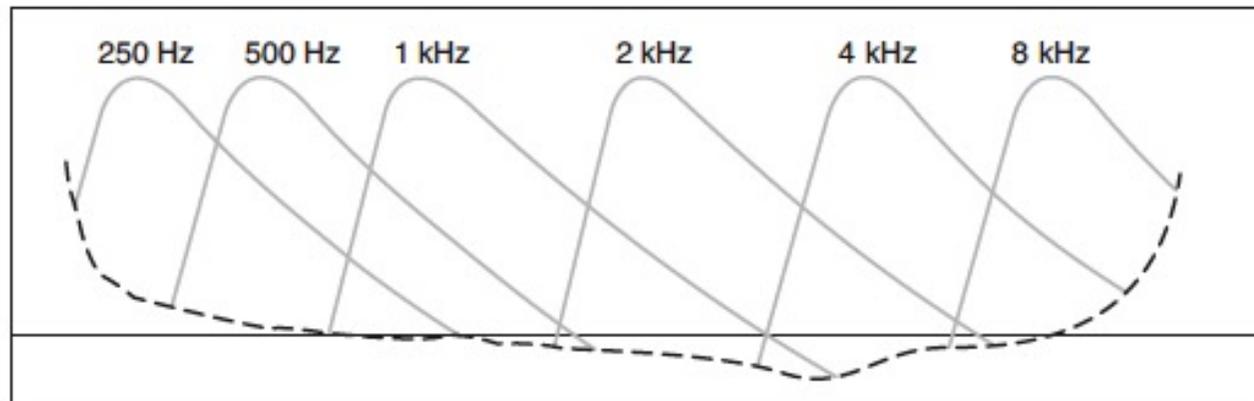
# Soglie di mascheramento



(a)

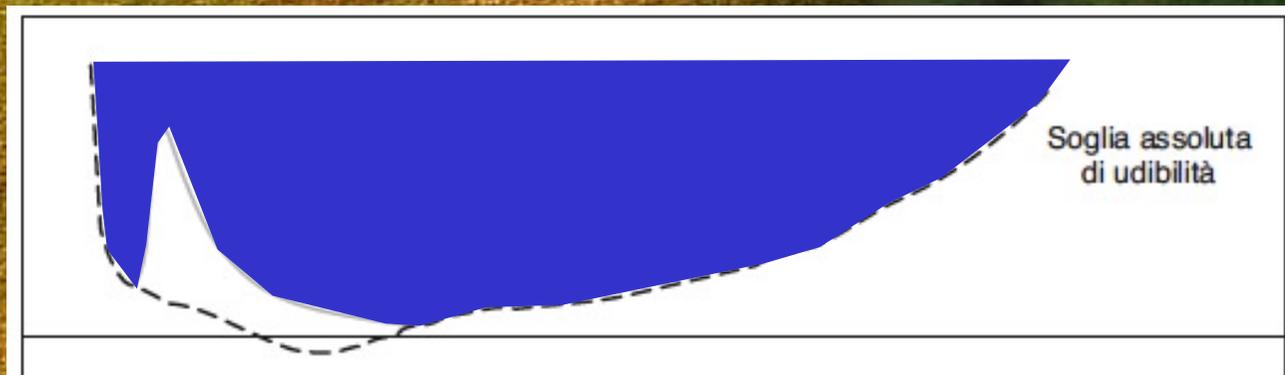
$$TH_N = E_T - 14,5 - B$$

$$TH_T = E_N - K$$



(b)

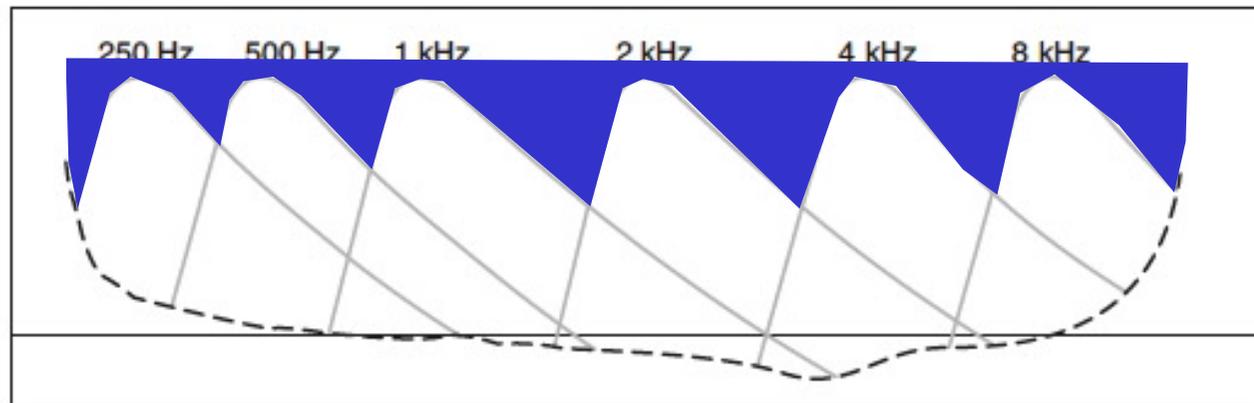
# Suono effettivamente percepito



(a)

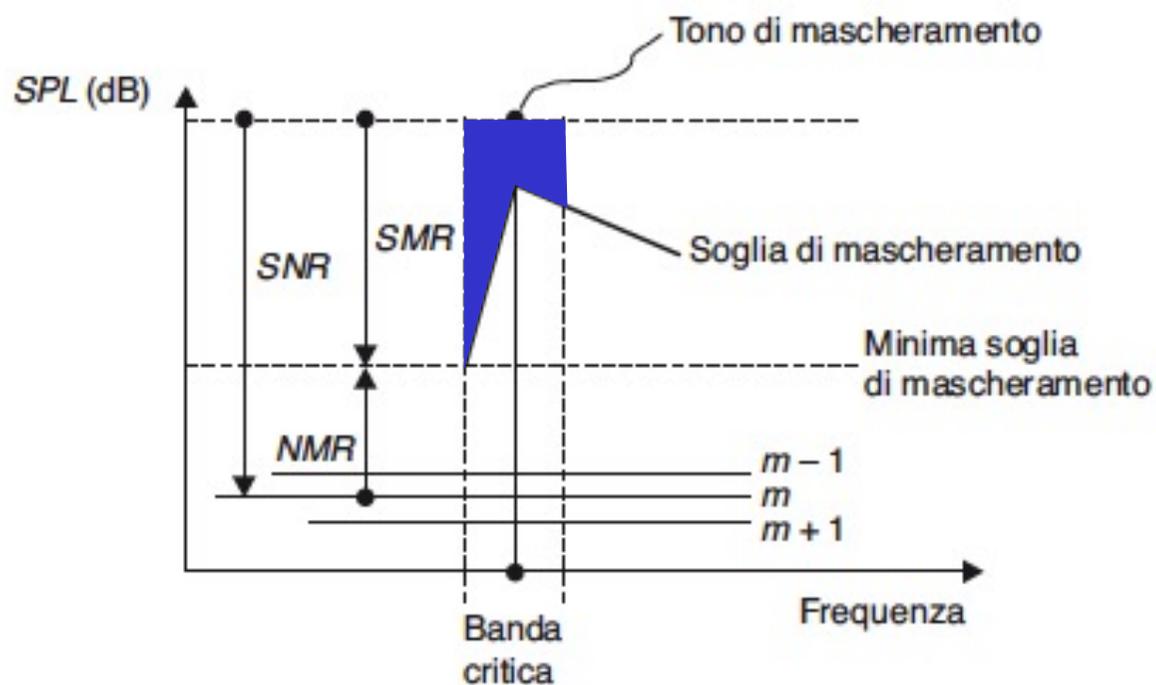
$$TH_N = E_T - 14,5 - B$$

$$TH_T = E_N - K$$



(b)

# Soglie di mascheramento e SNR



$$SNR_{log} = \log\left(\frac{\text{max\_ampiezza}}{\text{rumore\_medio}}\right) = \log(\text{max\_ampiezza}) - \log(\text{rumore\_medio})$$

# Tecniche di codifica

Codifica per blocchi (*block coding*)

Codifica del dominio trasformato (*transform coding*)

Codifica per sottobande (*sub-band coding*)

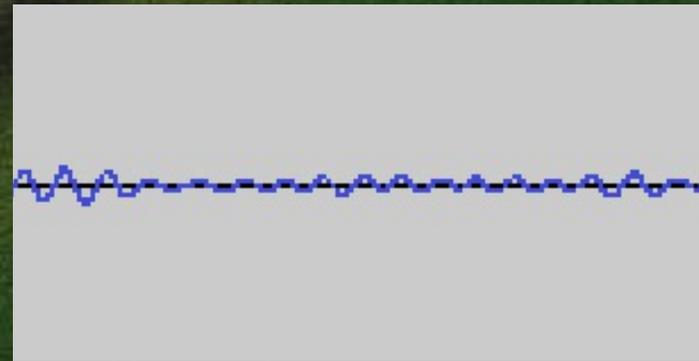
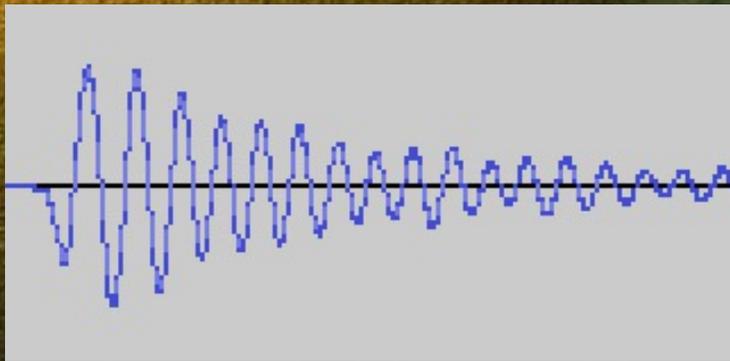
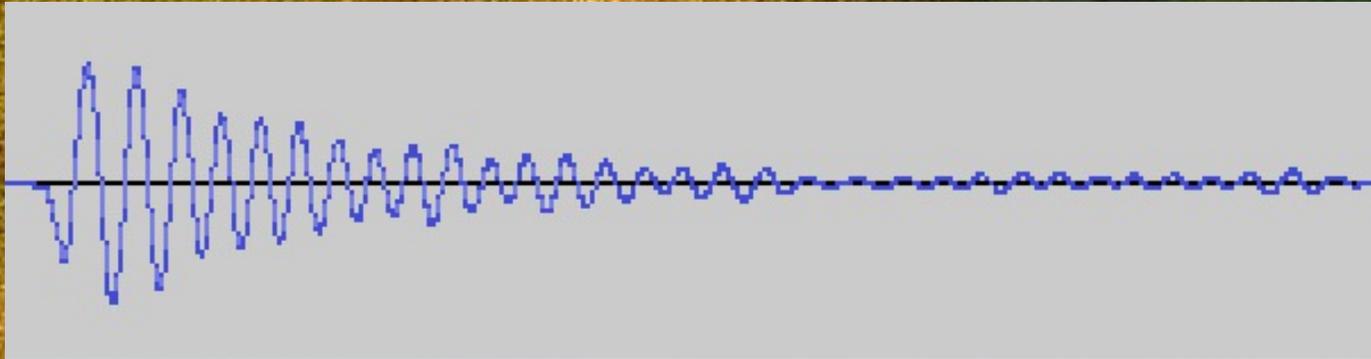
Codifica lossless (Huffman)

# Codifica per blocchi (block coding)

$$5 - 6 - 5 - 4 - \dots 2^y$$

- Codifica in virgola mobile: mantissa e esponente
  - Mantissa = valore da scalare dato l'esponente
  - Esponente come fattore di scala
- Motivazione: in un blocco, segnali audio reali non variano di ordine di grandezza  $\rightarrow$  esponente ( $y$ )
- Risparmio: si rappresenta un solo esponente
- Metodo: calcolo del numero max di bit di esponente, e si mantiene per tutto il blocco
- Problema: pre-echi

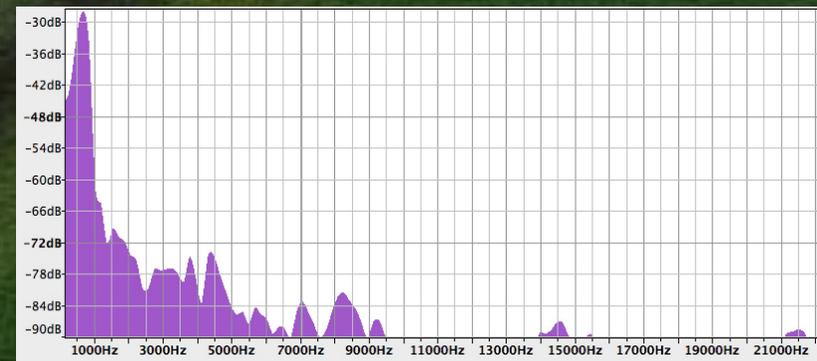
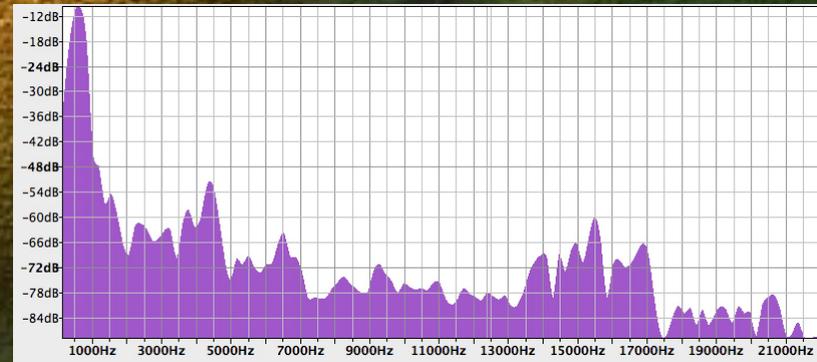
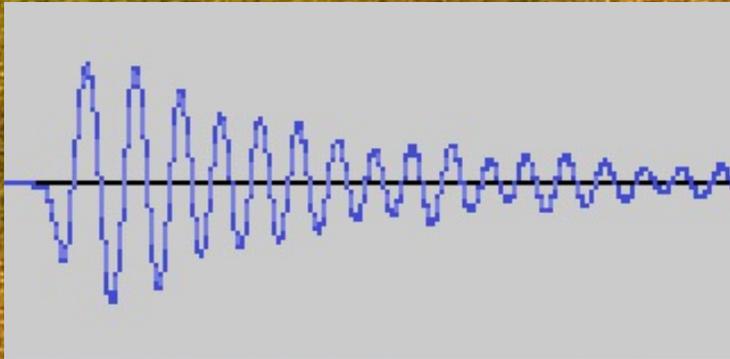
# Block coding (esempio)



# Codifica nel dominio della frequenza (transform coding)

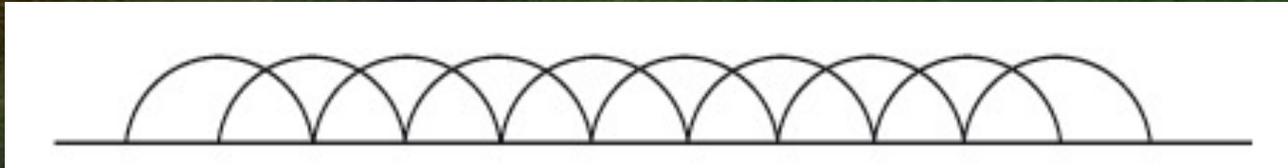
- Campioni di frequenza (e non di tempo)
- Motivazione: segnale audio cambia lentamente nel dominio della frequenza → tasso di campionamento inferiore
- Risparmio: maggiore ridondanza dei campioni → meno bit di quantizzazione
- Problema: Discriminare tra bande con diversa DR e mascheramento (SMR) → **subband coding**

# Transform coding (Esempio)



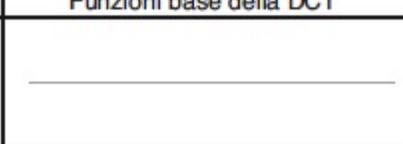
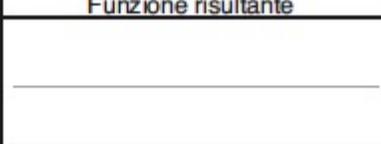
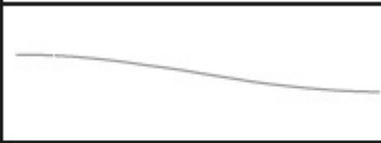
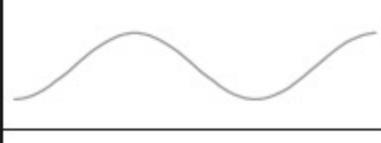
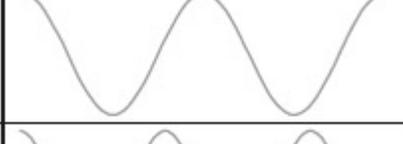
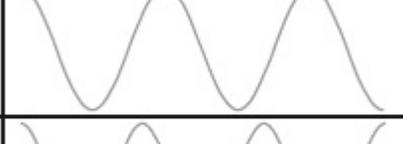
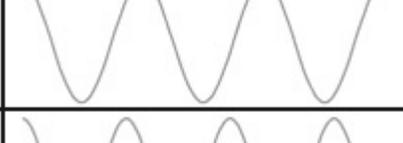
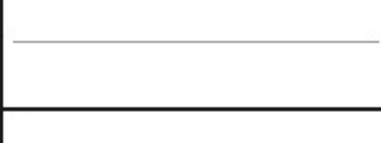
# Transform coding: problema dei transitori

- “Finestrare” il segnale in segmenti molto brevi
- Problema: divisione netta può introdurre artifici agli estremi della segmentazione (pre-echi)
- Soluzione: sovrapposizione dei segmenti; ciascun campione appartiene a due finestre, con ruoli differenti

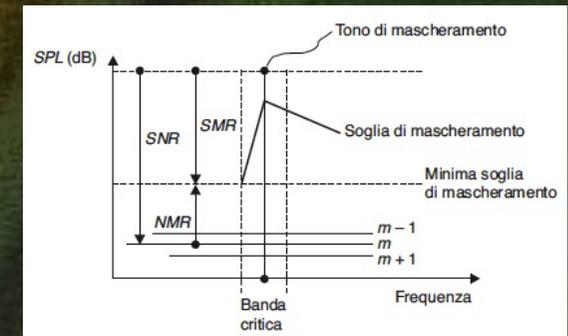


# Transform coding (uso DCT)

Discrete  
Cosine  
Transform

#	Funzioni base della DCT	Coefficiente	Funzione risultante
0		0	
1		0,2	
2		0,8	
3		-0,5	
4		0,2	
5		0	
6		0	
7		0	

# Codifica per sottobande (subband coding)



- Si divide lo spettro in bande da elaborare individualmente
- Vantaggio: le sotto-bande con DR ristretta si codificano con un minor numero di bit
- Numero di sotto-bande dipende dalle tecniche di compressione
  - massimo vantaggio dal fenomeno del mascheramento
  - nozione di banda critica, possibilmente restringendola

# Compressione lossless: Algoritmo di Huffman

## ALGORITMO DI HUFFMAN

1. Si selezionano due caratteri,  $x$  e  $y$ , con le frequenze minime
2. Si rimpiazzano  $x$  e  $y$  con un singolo carattere (fittizio)  $z$ , la cui probabilità è la somma delle probabilità di  $x$  e  $y$
3. Si torna al passo 1 con un alfabeto ridotto, in cui  $z$  sostituisce  $x$  e  $y$ , finché l'alfabeto conta un solo elemento di frequenza 100%

# Algoritmo di Huffman: esempio

~~A (60%)~~   ~~B (20%)~~   ~~C (8%)~~   ~~D (7%)~~   ~~E (5%)~~

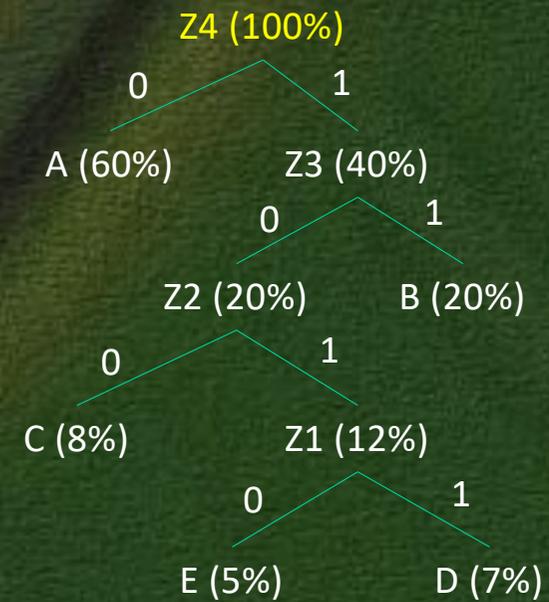
~~Z1 (12%)~~

~~Z2 (20%)~~

~~Z3 (40%)~~

Z4 (100%)

A 0  
B 11  
C 100  
D 1011  
E 1010



# Algoritmo di Huffman: esempio

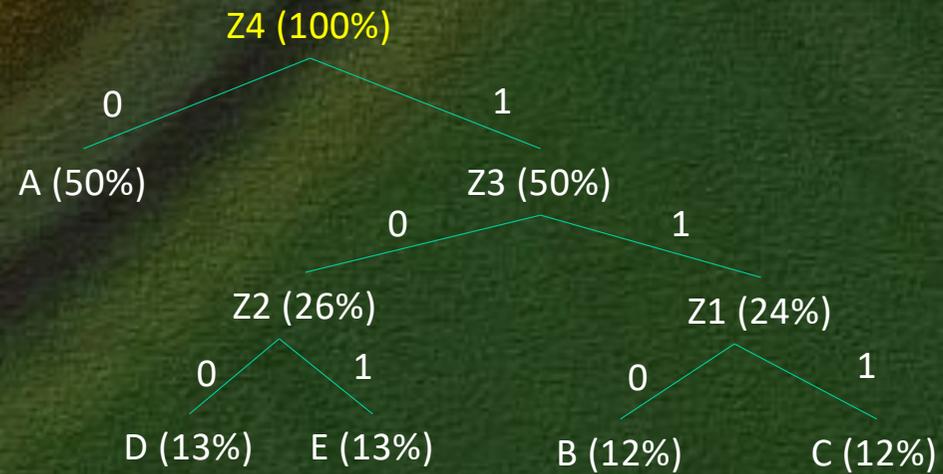
A (50%)   B (12%)   C (12%)   D (13%)   E (13%)

Alfabeto	Codifica
A	_____
B	_____
C	_____
D	_____
E	_____

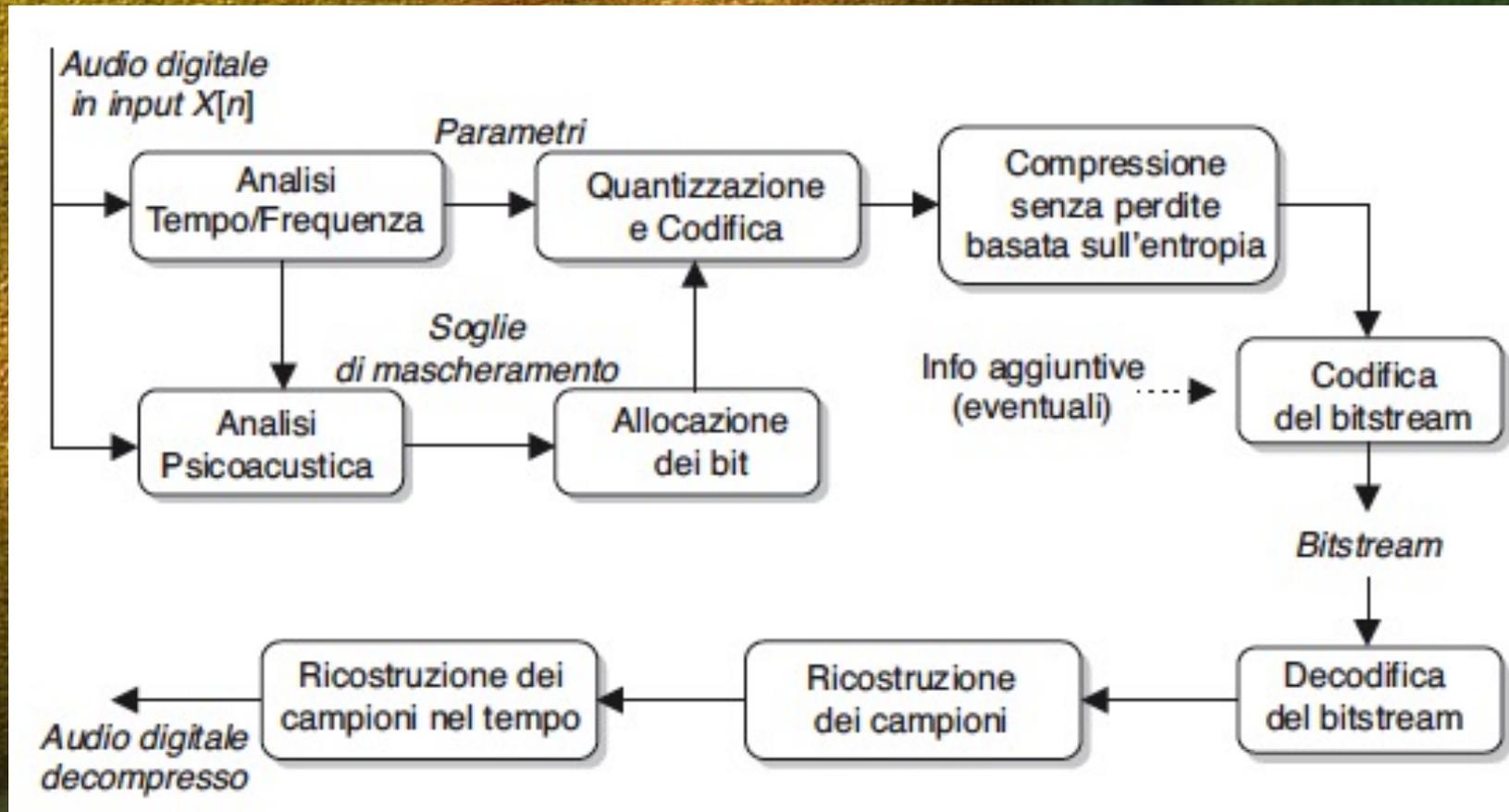
# Algoritmo di Huffman: esempio

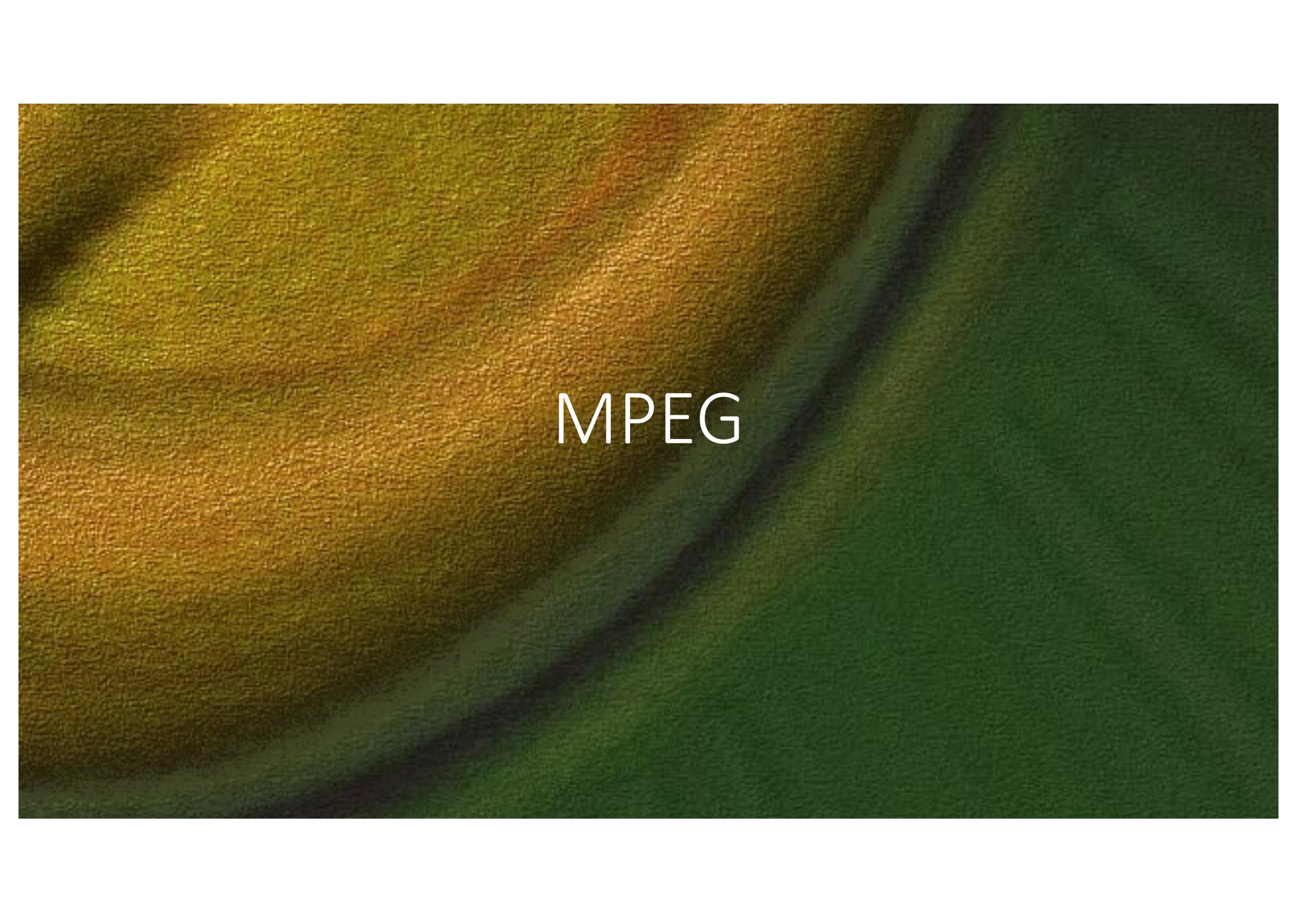
A (50%)   ~~B (12%)~~   ~~C (12%)~~   ~~D (13%)~~   ~~E (13%)~~  
                  ~~Z1 (24%)~~                   ~~Z2 (26%)~~  
  Z3 (50%)

Alfabeto	Codifica
A	0
B	110
C	111
D	100
E	101



# Schema generale di compressione percettiva





MPEG

# Motion Picture coding Experts Group

Gruppo di lavoro di ISO/IEC per lo sviluppo di standard internazionali per

- ... compressione, decompressione, elaborazione ...
- ... e rappresentazione codificata di video, audio, combinazioni A/V.

# Standard MPEG

- MPEG-1 (1992): codifica A/V a 1,5 Mbps
- MPEG-2 (1994): codifica A/V a 6 Mbps (surround + subwoofer)
- MPEG-3: per TV HD, assorbito da MPEG 2
- MPEG-4 (1999): A/V come oggetti in applicazioni multimediali
- MPEG-7 (2001): meta tag (XML), MPEG 47
- MPEG-21 (2001-): framework per ...
  - distribuzione e manipolazione di elementi digitali
  - Digital Rights Management – DRM
- MPEG-D (2007-): codifica dell'audio
  - MPEG Surround
  - Spatial Audio Object Coding
  - Unified speech and audio coding

<https://mpeg.chiariglione.org>

# Principi MPEG

- retro-compatibilità: decoder MPEG2 ok per MPEG1
- min elementi normativi per uniformità rappresentazione e efficienza decodifica
  - formato dell'audio compresso
  - struttura dell'algoritmo di decodifica
- effettivo algoritmo di compressione MPEG audio lasciato alla creatività degli sviluppatori
- specifiche includono *encoder*, ricavato dai test

# Diffusione MPEG

- Elettronica di consumo (con riproduttori portatili MP3 e MP4 – AAC)
- Telecomunicazioni (trasmissione audio su Internet)
- Multimedialità (inclusione di audio in prodotti multimediali off-line e on-line)

# MPEG 1

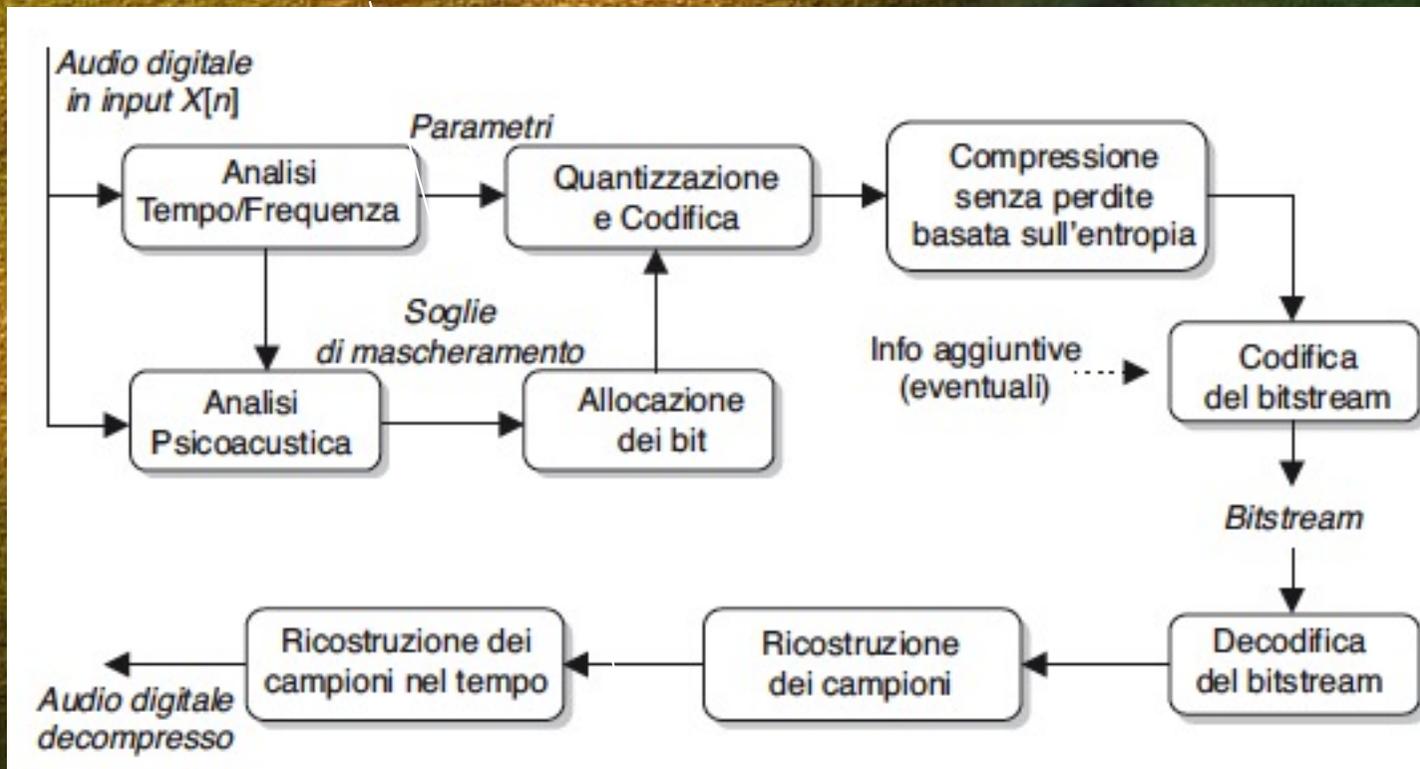
- 1987 progetto Eureka (DAB), incl. Fraunhofer IIS e Uni Erlangen
- algoritmo molto potente ISO-MPEG Audio
- struttura di base comune su più layer: sottobande + modello psicoacustico
- tre livelli (layer) di compressione:
  - Layer I: bit-rate > 128 kbps, poca compressione
  - Layer II: bit-rate = 128 kbps
  - Layer III: bit-rate = 64 kbps, molta compressione

# Schema generale MPEG

- segnale audio convertito nelle componenti spettrali
- banco di filtri (complessità diversa nei tre layer)

componenti spettrali  
quantizzate e codificate

rumore di  
quantizzazione  
sotto soglia  
di mascheramento  
fattore di scala  
per blocco  
Compansion  
Codici per subband



Compressione  
lossless

scompatta  
il bitstream

Da blocco di campioni di frequenza (subband)  
A blocco di campioni audio

banco di filtri (modalità di sintesi) dalle sotto-bande

# Modello psicoacustico e MPEG

- Presente solo nell'encoder, decoder semplice → Asimmetria MPEG, favorisce la riproduzione
- Nessun vincolo sul modello psicoacustico (qualità differisce)
- Implementazioni più accurate, FFT distinta
  - componenti spettrali tonali e non tonali
  - calcolo delle soglie di mascheramento globale

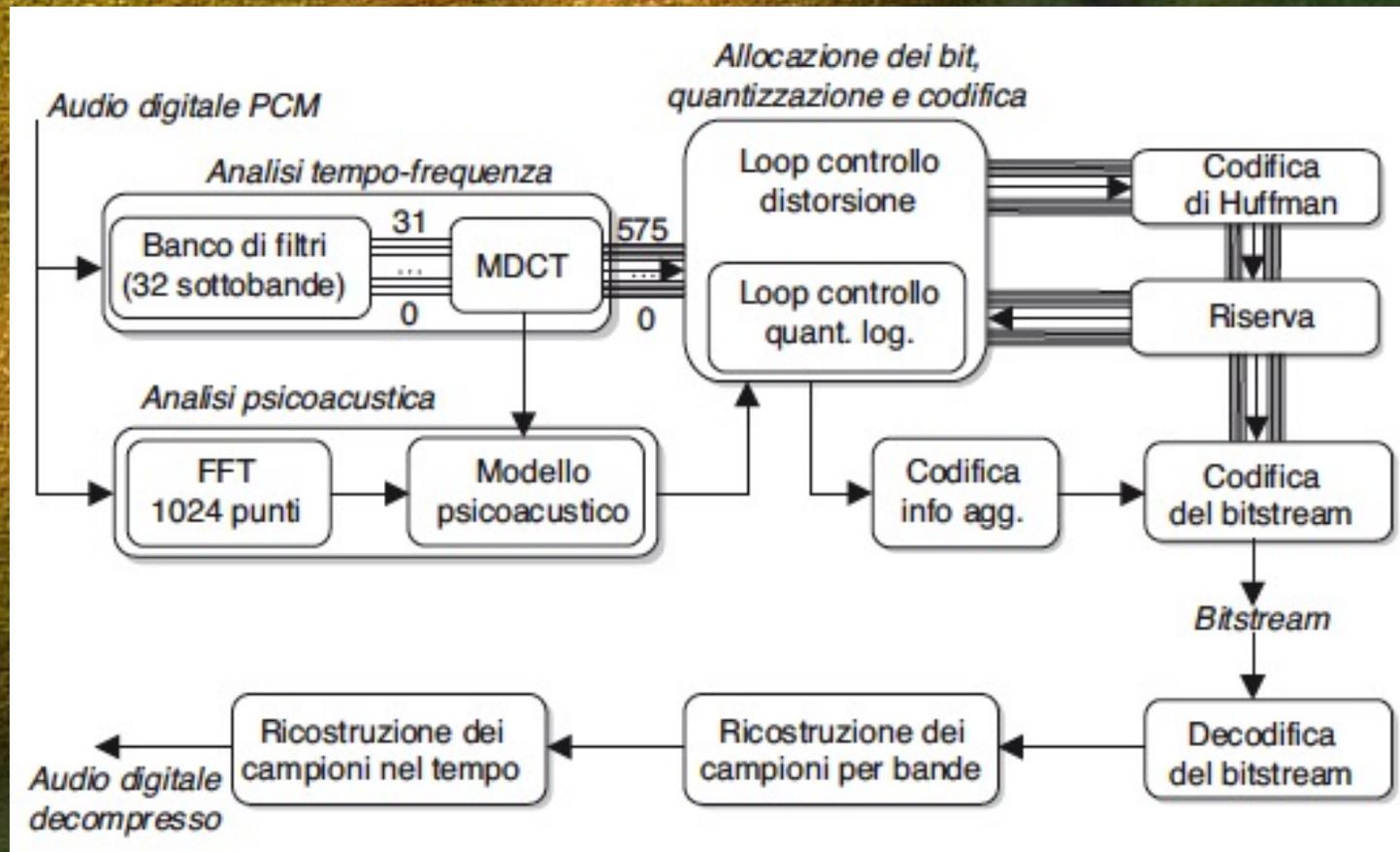
# Modalità di compressione MPEG 1

- Più frequenze di campionamento (32/44,1/48)
- Bitstream compresso supporta mono, dual mono, stereo, joint stereo
- Bit rate da 32 a 224 kbps (compressione da 2,7 a 24 volte) – tassi fissi e variabili
- Supporta controllo e correzione errori e informazioni supplementari

# Codifica joint stereo

- Risparmio quando canali L e R non molto diversi
- Metodo M=S (mid/side) stereo: L+R / L-R (lossless)
- Metodo intensity stereo (low quality):
  - percezione stereo per frequenze > 2 kHz basata su inviluppo temporale del segnale (non direzione)
  - L+R per sotto-bande > 2 kHz
  - differenze tra L e R solo sui fattori di scala

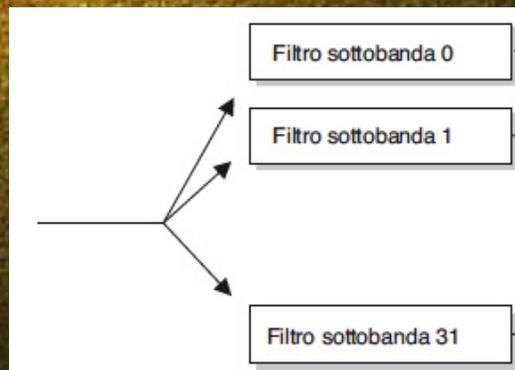
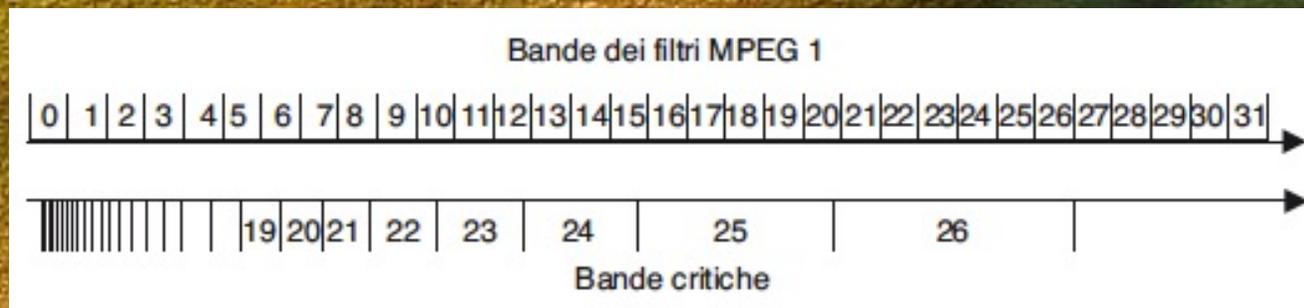
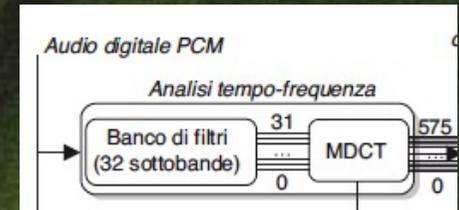
# Schema MPEG – Layer III





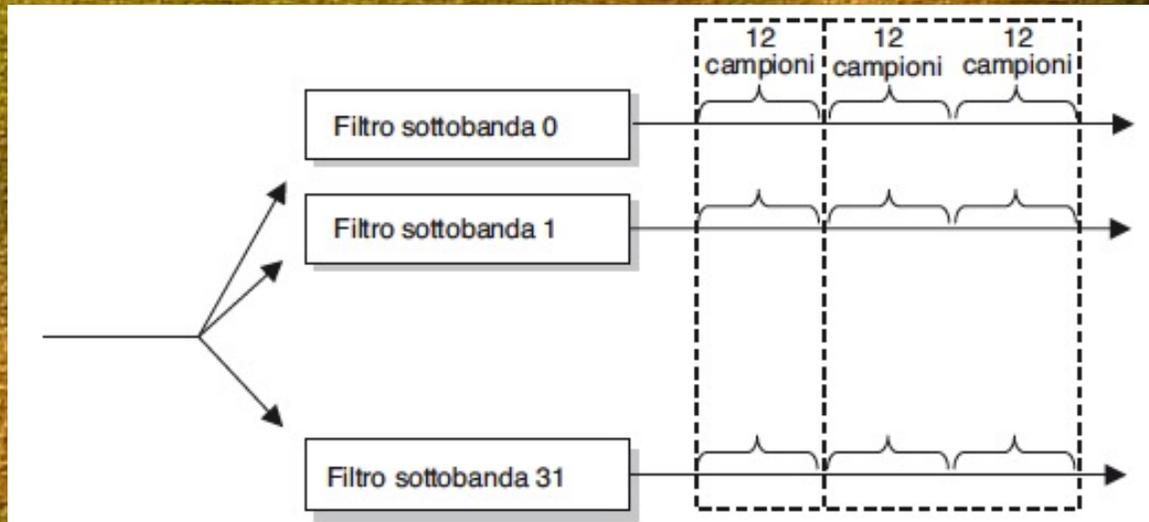
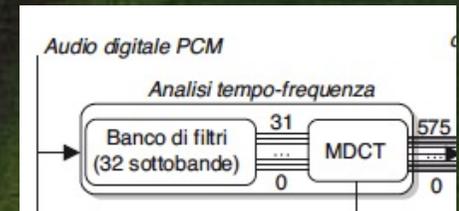
Elementi dello schema

# Codifica subband: Filtro polifase



32 sottobande ugualmente spaziate

# Block coding



Blocco temporale

a 48 kHz, 8 msec

384 campioni

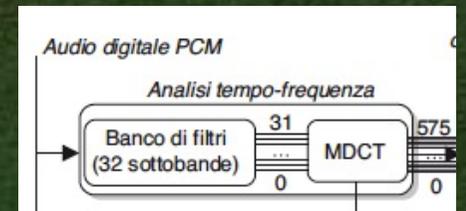
a 44,1 kHz, 8,7 msec

Blocchi (finestre, *frame*) di  $(12 \times 32 =) 384 \times 3 = 1152$  campioni (36 per sotto-banda)

pre-mascheramento temporale, parola binaria ancora lunga

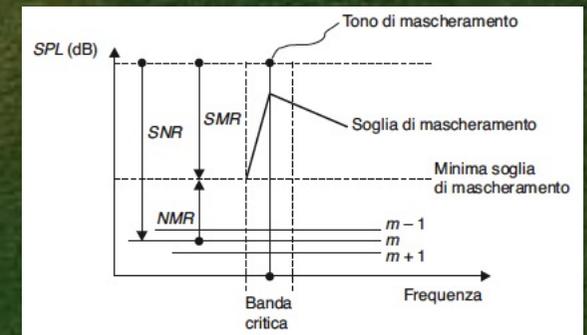
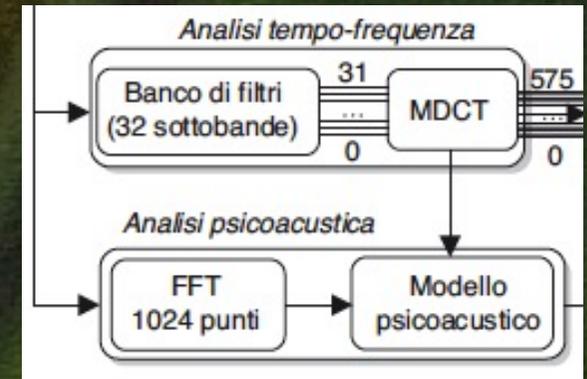
# Layer III (MP3): Subband + transform coding

- In cascata ai filtri a 32 bande, modulo *MDCT*
  - Più elevata risoluzione frequenziale
  - Basse frequenze: partizione simile alle bande critiche
  - ...
  - risoluzione totale fino a 576 sottobande (32x18)
- Si differenzia il trattamento del segnale stazionario / transitori, si limitano i pre-echi



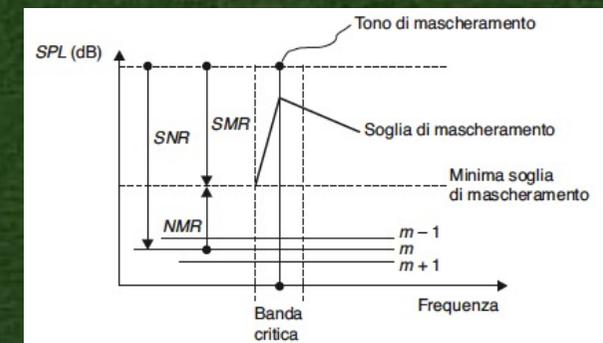
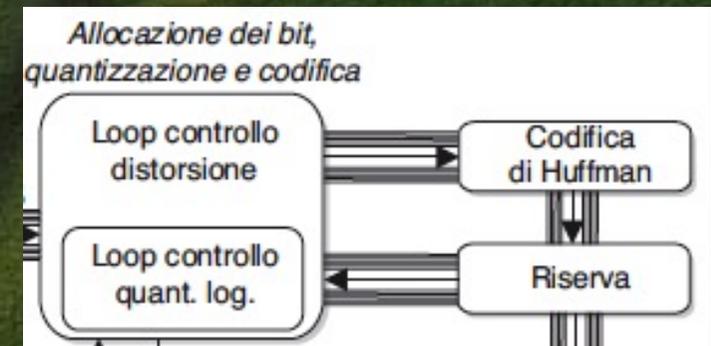
# Modello psico-acustico

- FFT da 512 a 1024 punti
- Separazione del segnale in tonale e non tonale
  - “perceptual subband coding”
  - “perceptual noise shaping”
- Calcolo soglie mascheramento globale
- Calcolo rapporto SMR (Signal to Masker Ratio) per quantificare i dati da comprimere



# Allocazione bit, quantizzazione, codifica

- Più cicli di quantizzazione e codifica
  - alloca i bit necessari per sottobanda (calcolo SMR)
  - quantizzazione non uniforme (III), compansion
  - sotto-bande raggruppate per fattore di scala
- Fattori
  - modello psicoacustico
  - bitrate in uscita
- Codici di Huffman per le 576
- Bitstream (III) soddisfa bit-rate variabili per sotto-banda
  - finestre di 1152 campioni, diversa lunghezza
  - “riserva” di bit

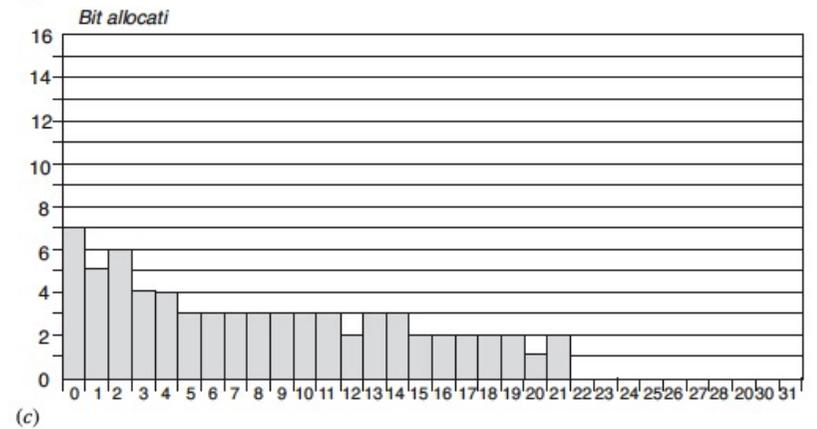
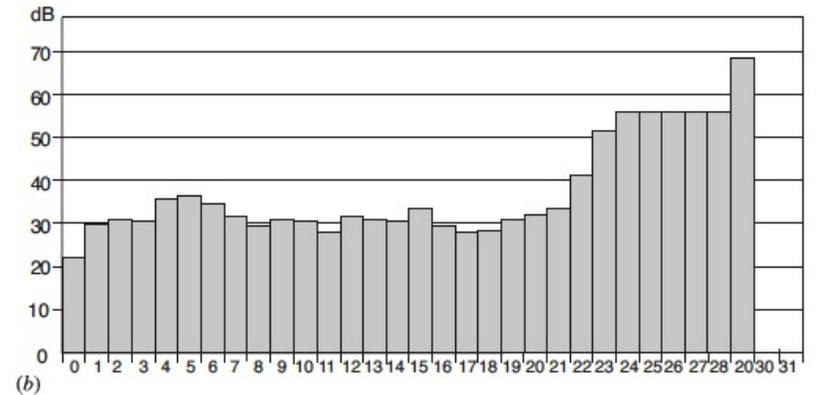
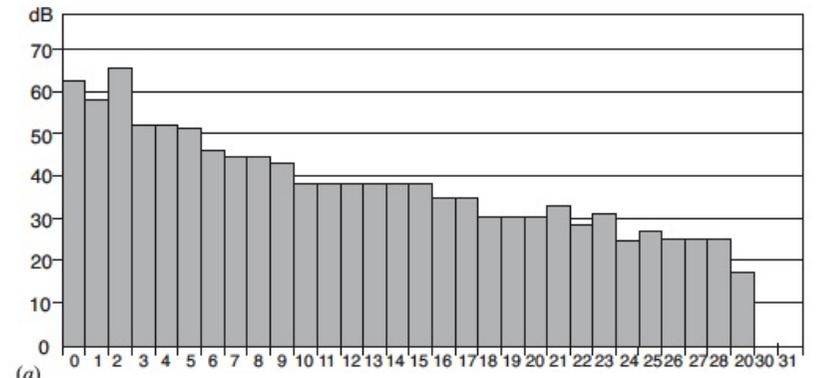


# Esempio di allocazione

Spettro di ampiezza

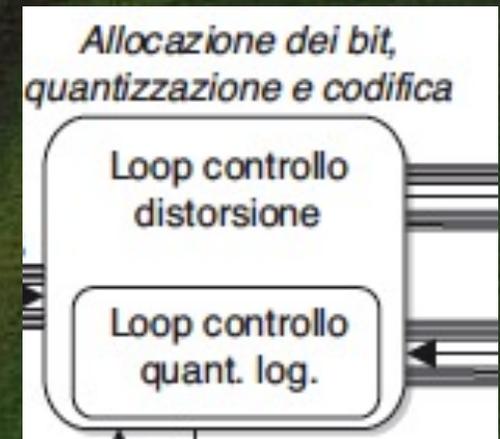
Soglie di mascheramento

Allocazione dei bit



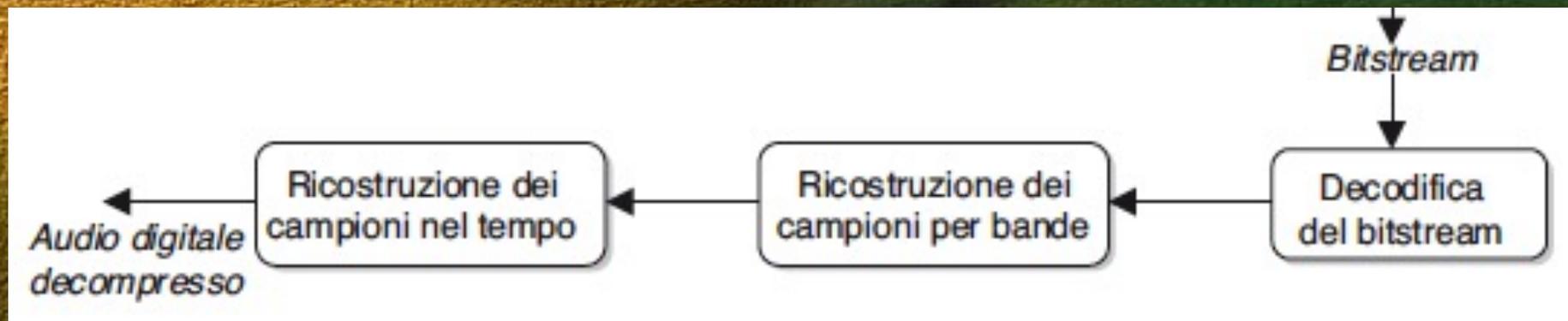
# Loop e terminazione

- Due loop
  - Loop interno sui limiti del bitrate (RATE LOOP)
  - Loop esterno sul controllo del rumore per banda
- 3 casi di terminazione
  - In nessuna banda il rumore supera la soglia del mascheramento (no distorsione)
  - Iterare ancora comporta per qualche banda il superamento dei valori permessi
  - Prossima iterazione aumenta i bit per tutte le bande
- Gli encoder che lavorano in real time hanno anche limiti di tempo



# Decodifica

- Sintetizza un segnale a partire dalle componenti spettrali codificate
- Non si ha più lo stesso segnale!!!
- Qualità dipende dal bit-rate



# Performance di MP3

- qualità telefonica: 96:1 (2.5 kHz / mono / 8 kbps)
- meglio di AM radio: 24:1 (7.5 kHz / mono / 32 kbps)
- simile a FM radio: 26..24:1 (11 kHz / stereo / 56...64 kbps)
- quasi-CD: 16:1 (15 kHz / stereo / 96 kbps)
- CD: 14..12:1 (>15 kHz / stereo / 112..128kbps) prende approx. 1Mb/minute di spazio hard-disk
  
- Oltre: 8...4:1 per la musica acustica

# L'encoder migliore

- La risposta dipende dalle esigenze
  - encoder veloci/lenti (> velocità, < fedeltà audio)
  - confrontare mp3 ottenuti da encoder diversi a parità di bit-rate
- Consiglio pratico (ovvio): creare MP3 con basso bitrate con encoder 'lenti'

# Frontend MP3

- Interfaccia ai codec
- Parametro di qualità o bit-rate
- Alcuni frontend implementano funzioni per “normalizzare” il volume, o realizzano ID TAGS
- IDTAG è informazione (testuale, in genere) nel file di layer III (autore, titolo, etc ...)



Formato MPEG

# Formato di file MPEG

- Funzionale per trasmissione di musica sulla rete
- Basato su blocchi indipendenti l'uno dall'altro
  - 1152 campioni divisi per sotto-bande nel Layer III
  - Ciascun blocco con intestazione e info aggiuntive
- Non include un'intestazione globale del file
  - Riproduzione anche parziale (inizio qualsiasi frame)
  - Non completamente vero per Layer III (riserva di bit)

# Frame MPEG e intestazione

- lunghezza variabile
- intestazione di quattro byte (32 bit)

**AAAAAAAA AAABBCCD EEEFFGH IIJJKLMM**

# Intestazione frame MPEG

AAAAAAAA AAABBCCD EEEFFFGH IIJJKLMM

- 11 bit A sempre a 1
  - campo della sincronizzazione del frame (*frame synch*).
  - inizio frame = byte 255 + byte con primi tre bit a 1
- 2 bit B: versione di MPEG del frame (MPEG-1, 2, ...)
- 2 bit C: livello (Layer I, II o III)
- 1 bit D: flag per frame protetto
  - meccanismo CRC
  - insieme ai dati, 16 bit di controllo

# Intestazione frame MPEG

AAAAAAAA AAABBCCD **EEEE**FFGH I I J J K L M M

Bitrate in kbps

Bit E	V1/LI	V1/LII	V1/LIII	V2/LI	V2/LII,LIII
0000	free	free	free	free	free
0001	32	32	32	32	8
0010	64	48	40	48	16
0011	96	56	48	56	24
0100	128	64	56	64	32
0101	160	80	64	80	40
0110	192	96	80	96	48
0111	224	112	96	112	56
1000	256	128	112	128	64
1001	288	160	128	144	80
1010	320	192	160	160	96
1011	352	224	192	176	112
1100	384	256	224	192	128
1101	416	320	256	224	144
1110	448	384	320	256	160
1111	N.V.	N.V.	N.V.	N.V.	N.V.

# Intestazione frame MPEG

AAAAAAAA AAABBCCD EEEE**FFGH** IIJJKLMM

- 2 bit F: frequenza di campionamento
  - 00 = 44100 in MPEG-1 e 22050 in MPEG-2,
  - 01 = 48000 e 24000,
  - 10 = 32000 e 16000,
  - 11 configurazione riservata.
- 1 bit G: bit di padding (o riempitivo) --- extra byte o meno
- 1 bit H: uso privato; attivazione di eventi specifici per particolari applicazioni

# Intestazione frame MPEG

AAAAAAAA AAABBCCD EEEFFFGH **IIJJKLMM**

- 2 bit I: informazione di canale
  - 00 *stereo*, 01 *joint stereo*, 10 *dual mono*, 11 *mono*
- 2 bit J (caso I = 01, cioè *joint stereo*): info di join
  - 00 (I e II intensity stereo a bande 4-31; III stereo)
  - 01 (I e II intensity stereo a bande 8-31; III intensity stereo)
  - 10 (I e II intensity stereo a bande 12-31; III M/S stereo)
  - 11 (I e II intensity stereo a bande 16-31; III intensity + M/ S)
- 1 bit K: frame coperto da copyright
- 1 bit L: frame in medium originale o copia
- 2 bit M: enfasi eventualmente applicata (compansion Dolby)

# Tag ID3

- Inclusi nei file MPEG
- Tipologie: Titolo, Esecutore, Album, Anno di edizione, Genere musicale, Commenti liberi
- Meta-dati
  - Estratti e mostrati sull'interfaccia di riproduzione
  - Inseriti al tempo della codifica o più avanti
  - Processo di scrittura gestito dagli encoder

# Tag ID3: generi musicali

0 'Blues'	20 'Alternative'	40 'AlternRock'	60 'Top 40'
1 'Classic Rock'	21 'Ska'	41 'Bass'	61 'Christian Rap'
2 'Country'	22 'Death Metal'	42 'Soul'	62 'Pop/Funk'
3 'Dance'	23 'Pranks'	43 'Punk'	63 'Jungle'
4 'Disco'	24 'Soundtrack'	44 'Space'	64 'Native American'
5 'Funk'	25 'Euro-Techno'	45 'Meditative'	65 'Cabaret'
6 'Grunge'	26 'Ambient'	46 'Instrumental Pop'	66 'New Wave'
7 'Hip-Hop'	27 'Trip-Hop'	47 'Instrumental Rock'	67 'Psychedelic'
8 'Jazz'	28 'Vocal'	48 'Ethnic'	68 'Rave'
9 'Metal'	29 'Jazz+Funk'	49 'Gothic'	69 'Showtunes'
10 'New Age'	30 'Fusion'	50 'Darkwave'	70 'Trailer'
11 'Oldies'	31 'Trance'	51 'Techno-Industrial'	71 'Lo-Fi'
12 'Other'	32 'Classical'	52 'Electronic'	72 'Tribal'
13 'Pop'	33 'Instrumental'	53 'Pop-Folk'	73 'Acid Punk'
14 'R&B'	34 'Acid'	54 'Eurodance'	74 'Acid Jazz'
15 'Rap'	35 'House'	55 'Dream'	75 'Polka'
16 'Reggae'	36 'Game'	56 'Southern Rock'	76 'Retro'
17 'Rock'	37 'Sound Clip'	57 'Comedy'	77 'Musical'
18 'Techno'	38 'Gospel'	58 'Cult'	78 'Rock & Roll'
19 'Industrial'	39 'Noise'	59 'Gangsta'	79 'Hard Rock'

# Tag ID3: v1 e v2

- ID3v1

- lunghezza fissa di 128 byte
- localizzato alla fine di un file MPEG

- ID3v2

- non ha limitazioni (immagini, lyrics, note di performance, preset di equalizzazione, ...)
- localizzato all'inizio del file (supporta streaming)
- una struttura aperta, può essere ampliata

# Dopo MP3

MPEG-AAC, Dolby AC-3, WMA,  
Ogg Vorbis, FLAC

# MPEG-AAC (Advanced Audio Coding)

- Già con MPEG-2, soprattutto in MPEG-4
- Formati
  - .aac (formato “raw” per streaming AAC)
  - .mp4 (file MPEG-4, con video, testo o immagini)
  - .m4a (versione solo audio di mp4)
  - .m4v (versione video di mp4)
  - .3gp (mp4 ottimizzato per dispositivi mobili)
- Il generazione di codifiche di tipo percettivo
- Per segnali stereo e multicanale

# MPEG-AAC

- Due modalità
  - ampliate modalità operative di MPEG-1 con back-compatibility
  - nuovo sistema, rinunciando alla back-compatibility
- supporta fino a 48 canali audio VS. 2 di MP3
- supporto immediato di segnali mono, stereo e 5.1

# MPEG AAC

- Base: moduli di codifica che migliorano MP3 (Es.: MDCT a 1024 bande invece di 576)
- bitstream in differenti profili:
  - Principale, poco usato, usa tutti i moduli
  - Bassa complessità (AAC-LC), compressioni elevate, usa alcuni moduli, riducendo le risorse impegnate
  - Basso ritardo (AAC-LD), per riproduzione in real-time
  - Tasso di campionamento scalabile (SSR o HE-AAC), quattro bande, bitstream indipendente, noise shaping
- Anche codifiche lossless

# MPEG AAC

- Mercato consumer da MP3 a AAC
  - ampio consenso sulle prestazioni migliori
  - materia di dibattito il grado di miglioramento
- Esempi:
  - 128 kbps AAC comparabile a 192 kbps MP3
  - 320 kbps AAC per codifica multicanale 5.1
- Codifica standardizzata dei metadati → AAC compatibile con informazioni aggiuntive di altri media e formati

# MPEG e diritti a pagamento

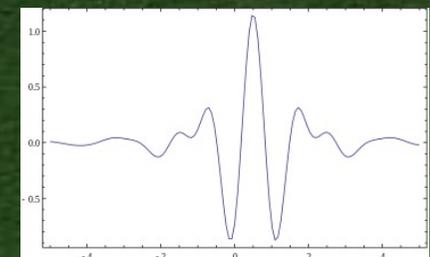
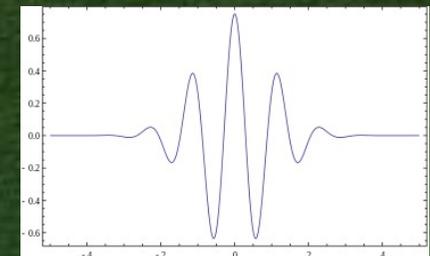
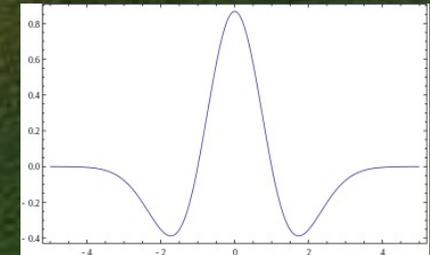
- Standard MPEG suddiviso in
  - Parte normativa: specifica la struttura del bitstream e del decoder e gli strumenti usati dall'encoder
  - Parte informativa: descrive la struttura dell'encoder (nello standard è un esempio)
- Uso libero della parte normativa: ok decoder che usano il formato del bitstream standard
- A pagamento: encoder per fini commerciali
- Brevetti MP3 appannaggio di Thomson e Fraunhofer

# Window Media Audio (WMA)

- Proprietario Microsoft (suite Windows Media, tra cui ASF per web radio)
- Prestazioni migliori di MP3 a parità di bit-rate
  - Buone su audio musicale ma non su parlato
  - Non molto diffuso
- Deficienze di WMA
  - immagine stereo degradata
  - formazione di pre-echi per transitori (come Layer I e II)
- Vantaggio: integrazione con suite Windows Media di Microsoft (ASF adotta in parte WMA)

# OGG Vorbis

- Sistema open-source, gestito Xiph.org
- Idea: creazione di standard di qualità pubblico
- Modalità operative comparabili a MPEG-2
- Due versioni con due differenti trasformate:
  - MDCT
  - banco di filtri basato su wavelet
- Tipica compressione di tipo percettivo
- Possibili più bit-rate



# Free Lossless Audio Codec (FLAC), di Xiph.org

- Fattore di compressione 50% (ZIP 10-12%)
  - Efficace per archiviazione dati e riproduzione di qualità
  - Orientato all'audio, competitivo sulla voce
  - Predizione lineare (ADPCM) e RLE per silenzio
  - Decodifica veloce, indipendente da compressione
- Usato per riproduzione a qualità CD
  - Ricevitori/riproduttori di musica incorporano FLAC (formato di qualità come WAV e AIFF)
  - Archiviazione (da parte dei proprietari dei CD)
  - Supporto per etichettatura e ricerca titoli

# Progetto open--source FFmpeg

- Suite di strumenti per codifica, decodifica, transcodifica, streaming, filtraggio, multiplexing e riproduzione di dati multimediali
- Idea: implementare qualsiasi codec progettato
  - da comitato di standard (come MPEG)
  - da company (come Microsoft)
- Librerie usate da app generiche e tool FFmpeg
- Centinaia di codec (alcuni protetti da licenze che vietano reverse engineering – Attenzione!)



Grazie dell'attenzione