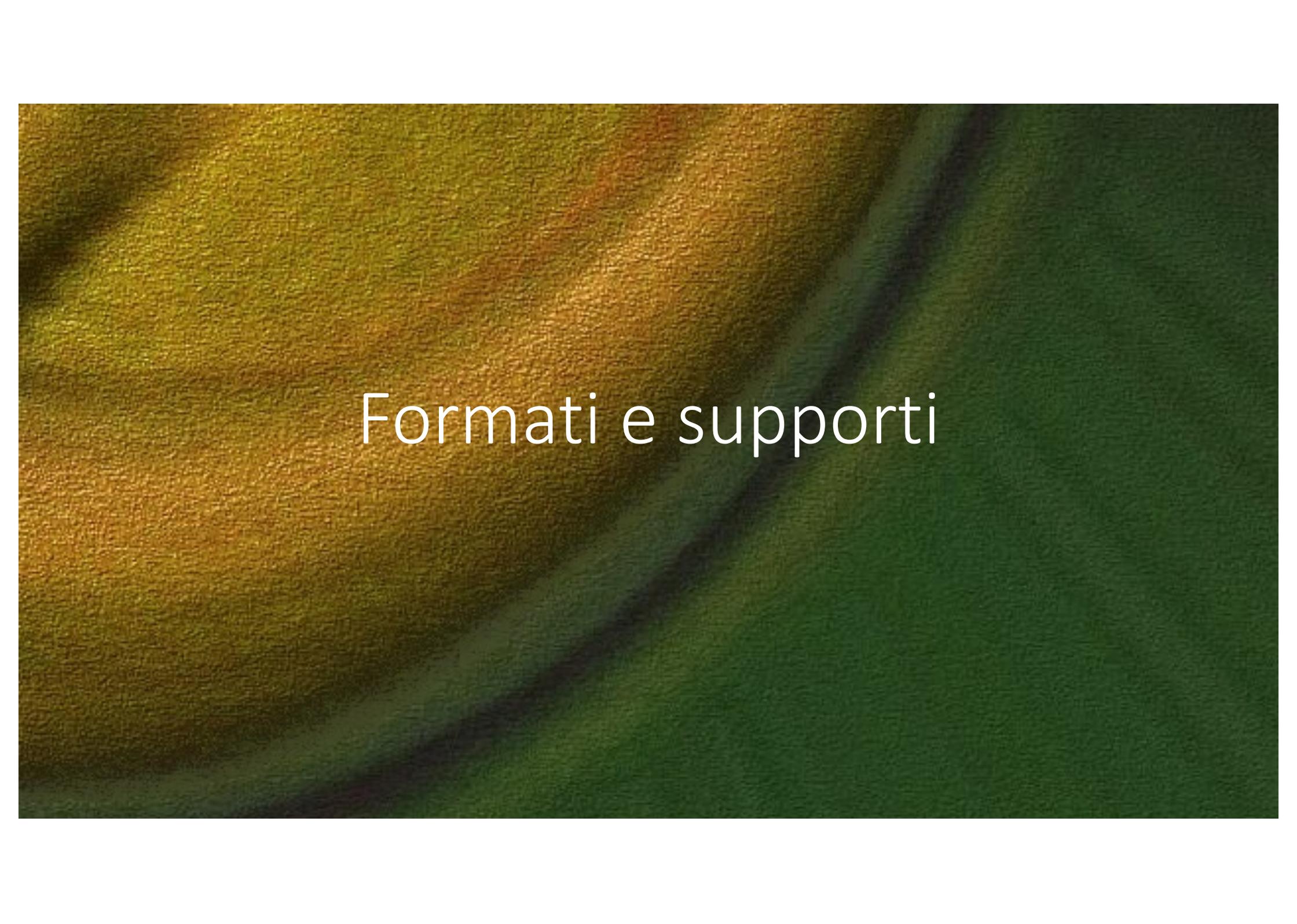




# Tecnologie digitali per il suono e l'immagine 2021/22

Vincenzo Lombardo  
Corso di Laurea in DAMS  
Università di Torino

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



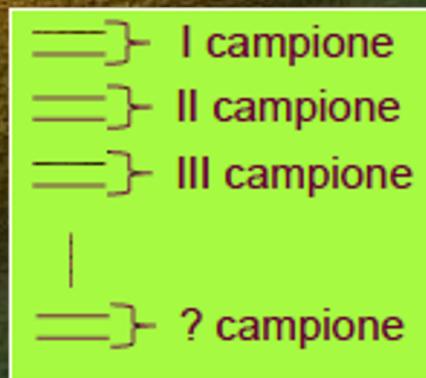
# Formati e supporti

# Archiviazione dei dati audio

- Fondamentali parametri di
  - campionamento
  - quantizzazione
  - codifica
- Ulteriori convenzioni utili nella descrizione
  - modalità di memorizzazione dei canali multipli
  - struttura interna dei campioni

# Interleaving dei campioni multicanale

- frame di campioni da tutti i canali
- N canali: frame di N campioni



Es. stereo: canale sx precede canale dx

# Frame multicanale

- Vantaggi
  - facile sincronizzazione dei canali
  - memorizzazione e trasmissione efficiente
- Svantaggi
  - spreco di spazio (N canali necessari?)
  - operazioni sui singoli canali

# Struttura interna dei campioni

Big Endian  
(Motorola)

Little Endian  
(Intel)

Big Endian

00000000  
00001000



Little Endian

00001000  
00000000

Esempio: 00000000 00001000

File format design per specifico chip

Es. AIFF per Motorola 680x0 (Apple Macintosh): Most Significant Byte (MSB) first , Big Endian

Es. Microsoft WAVE format per Intel 80x86: LSB first (Little Endian) (decimale '21' come '12')

# Formati

- Storicamente, ogni tipo di macchina usava un suo formato per i dati audio
- Alcuni formati applicabili in modo più generale
- Conversioni tra quasi tutti i formati (talvolta si perde informazione)

## Due tipi di formati

- autodescriventi: parametri e forma di codifica espliciti in una intestazione (*header*)
- senza intestazione (*headerless* - anche detti “*raw*”): parametri e codifica fissati a priori

# Formati con intestazione

- Definiscono una famiglia di codifiche di dati
  - <parametro, valore> definisce la variante usata
  - Es.: parametri del device per il campionamento
  - Es.: descrizione leggibile dall'utente (+ copyright)
- Header = “parola chiave” + dati di codifica
  - Es.: signed / unsigned
  - Es.: bytes / short integers

# Formati di file audio

Estensione	Origine	Parametri
.au, .snd	NeXT, Sun	tasso di campionamento, #canali, codifica, stringa di informazioni
.aif(f), AIFF	Apple, SGI	tasso di campionamento, #canali, quantizz., molte informazioni
.aif(f), AIFC	Apple, SGI	(AIFF con compressione)
.wav, WAVE	Microsoft	tasso di campionamento, #canali, quantizz., molte informazioni
.voc	Creative	tasso di campionamento (8 bits/1 ch; silence deletion)
.iff, IFF/8SVX	Amiga	tasso di campionamento, #canali, informazioni strumenti (8 bit)
.mp3, .mp4	MPEG	tasso di campionamento, #canali, qualità dei campioni
.sf	IRCAM	tasso di campionamento, #canali, codifica, informazioni
.ra	Real Networks	tasso di campionamento, #canali, qualità dei campioni

# Formati senza intestazione

Definiscono una singola codifica dei dati

- non ammettono variazioni dei parametri
- a volte più sr, ma può essere un problema tirarla fuori

Estensione (o nome) / Origine / Parametri

- .snd, .fssd / Mac, PC / var. sr, 1 canale, 8 bit unsigned
- .ul / telefonia US / 8 k, 1 canale, 8 bit "u-law"
- .snd / Amiga / var. sr, 1 canale, 8 bits signed



Formato Wave di Microsoft

# Formato WAVE (.wav)

- molto versatile: supporta più livelli di quantizzazione, tassi di campionamento, canali
- standard in programmi professionali di elaborazione audio
- caratteristiche discendono direttamente dal processore Intel (LITTLE ENDIAN)
- basato sul formato RIFF
- memorizzazione dei dati in sezioni logiche all'interno del file, dette *chunk*.



# Formato RIFF

Byte offset	Nome	Lunghezza (in byte)	Descrizione
00	rID	4	contiene i caratteri "RIFF"
04	rLEN	4	lunghezza rimanente del file
08	rDATA	RLEN	i dati

```
RIFF$@@NUL WAVEfmt DLENUL NUL NUL SOHNUL STXNUL d" NUL NUL OLE#STXNUL EOTNUL DLENUL dataNUL@@NUL NUL NUL NUL NUL NUL NUL NUL
NUL " VTNUL " VTNUL " FFNUL NUL NUL FFNUL NUL NUL
NUL NUL NUL
NUL NUL NUL SONUL NUL NUL SONUL NUL NUL ST NUL NUL NUL ST NUL NUL NUL ST NUL NUL NUL STNUL " STNUL " STNUL " STNUL " STNUL
```

# Formato WAV: definizione del chunk rDATA

rDATA Byte offset	Nome	Lunghezza (in byte)	Descrizione
00	wID	4	contiene i caratteri "WAVE"
04	Format chunk	24	chunk del formato dei dati nel data chunk
28	Data chunk	?	chunk dei dati

```
RIFF$@@NUL WAVEfmt DLE NUL NUL NUL SOH NUL STX NUL 0" NUL NUL DLE=STX NUL EOT NUL DLE NUL data NUL @NUL NUL NUL NUL NUL NUL N
NUL " VT NUL " VT NUL " FF NUL NUL NUL FF NUL NUL NUL
NUL NUL NUL
NUL NUL NUL SO NUL NUL NUL SO NUL NUL NUL ST NUL NUL NUL ST NUL NUL NUL ST NUL NUL NUL ST NUL " ST NUL " ST NUL " ST NUL " ST NUL
```

# Formato WAVE: format chunk

WAVE Byte offset	Nome	Lunghezza (in byte)	Descrizione
00	fID	4	Contiene i caratteri «fmt »
04	fLEN	4	Lunghezza di questo chunk
08	wFormatTag	2	Codifica del segnale (1 = PCM)
10	#Canali	2	1 = Mono, 2 = Stereo
12	#Campioni/sec	4	Tasso di riproduzione
16	#AvgByte/sec	4	Media dei byte al secondo
20	#Block Align	2	Numero di allineamento
22	wBitsPerSample	2	Numero di bt per campione (8, 16, ...)
24	FormatSpecific	?	Campi aggiuntivi (non esistono per wFormatTAG = 1)

```

RI FF $ @ 0 N U L N A V E f m t D L E N U L N U L N U L S O H N U L S T X N U L d " N U L N U L D L E = S T X N U L E O T N U L D L E N U L d a t a N U L @ @ N U L N U L N U L N U L N U L N U L N U L
N U L * * V T N U L * * V T N U L * * F F N U L N U L N U L F F N U L N U L N U L
N U L N U L N U L
N U L N U L N U L S O N U L N U L N U L S O N U L N U L N U L S T N U L N U L N U L S T N U L N U L N U L S T N U L N U L N U L S T N U L * * S T N U L * * S T N U L * * S T N U L * * S T N U L
    
```

# Formato WAVE: data chunk

rDATA	Byte offset	Nome	Lunghezza (in byte)	Descrizione
00		dID	4	Contiene i caratteri "data"
04		dLEN	4	Lunghezza del campo dDATA
08		dDATA	dLEN	Dati audio

```

RIFF$@@NUL WAVEfmt 0LE NUL NUL NUL SOH NUL STX NUL d" NUL NUL 0LE# STX NUL EOT NUL 0LE NUL data NUL @@ NUL NUL NUL NUL NUL NUL N
NUL " VT NUL " VT NUL " FF NUL NUL NUL FF NUL NUL NUL
NUL NUL NUL
NUL NUL NUL SO NUL NUL NUL SO NUL NUL NUL ST NUL NUL NUL ST NUL NUL NUL ST NUL NUL NUL ST NUL " ST NUL " ST NUL " ST NUL " ST NUL

```

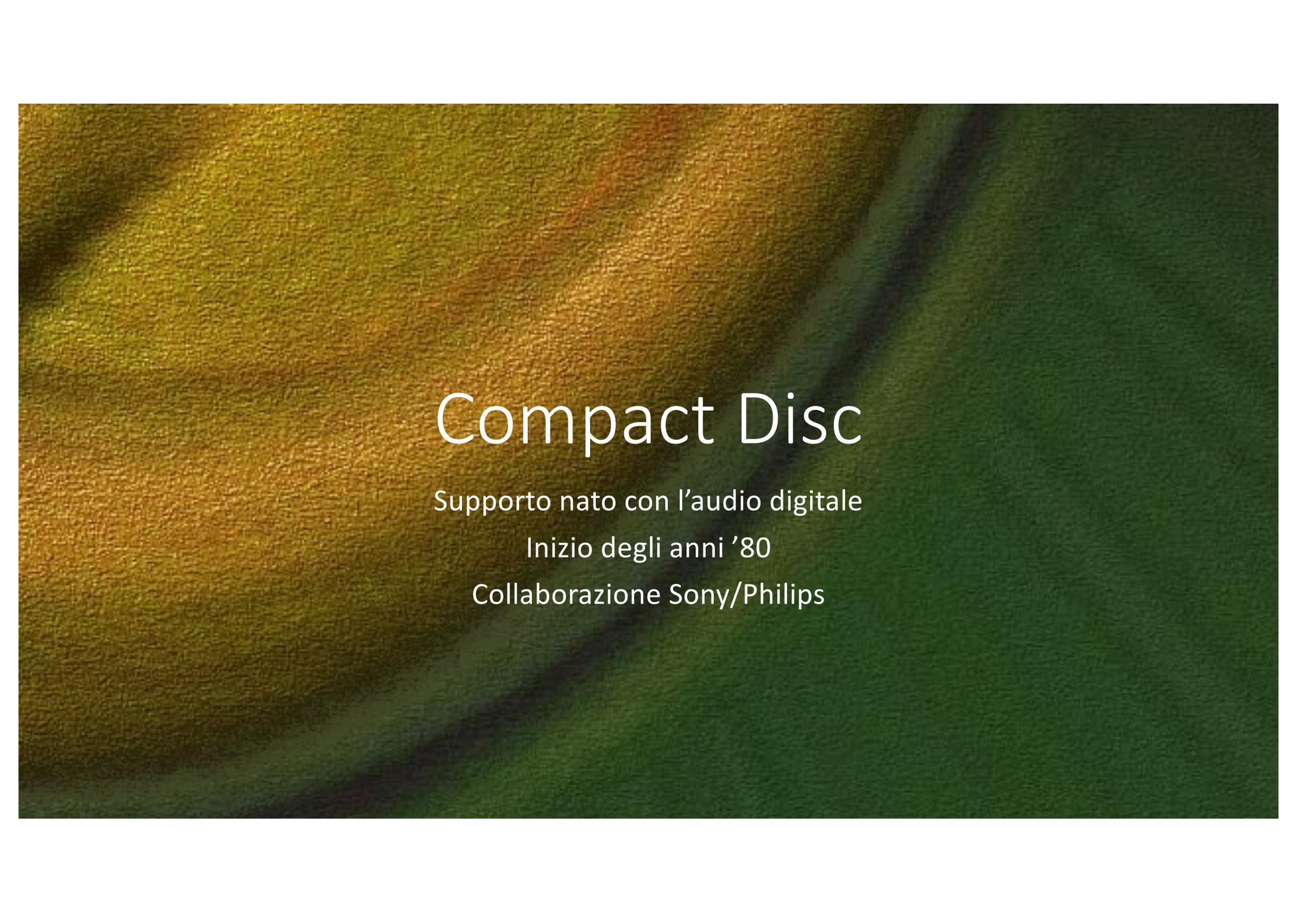




# Supporti per l'audio digitale

# Supporti

- Caratteristiche fisiche (magnetici, ottici, elettronici) e logiche (struttura dei dati)
- caratteristiche rilevanti per l'audio digitale:
  - capacità di contenere i dati
  - velocità di scrittura/lettura per registrazione/riproduzione
- Richieste per 1 minuto, qualità CD
  - Capacità: oltre 10 MB
  - Velocità di trasferimento dati: 176 KB/sec

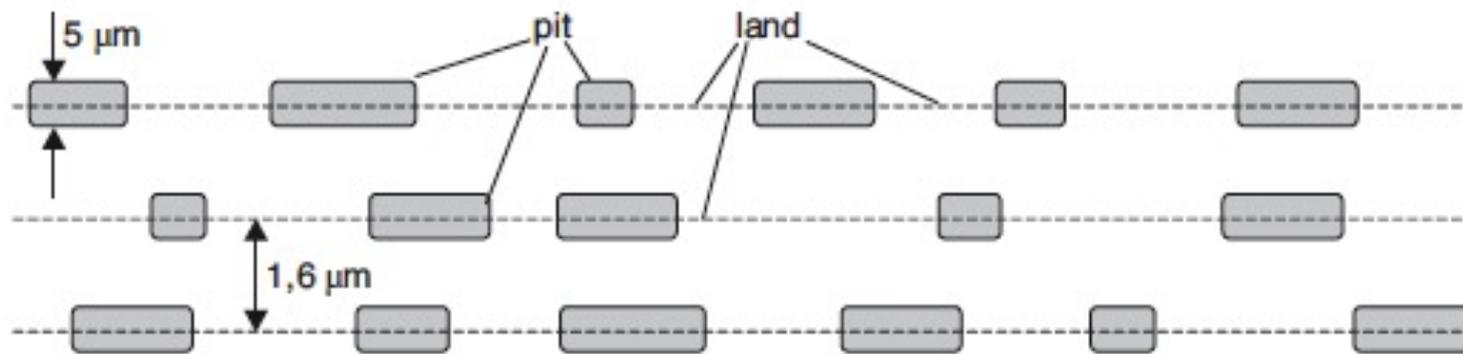
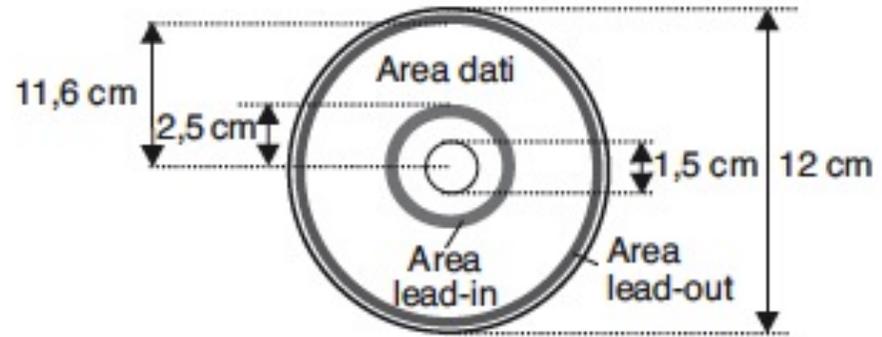


# Compact Disc

Supporto nato con l'audio digitale

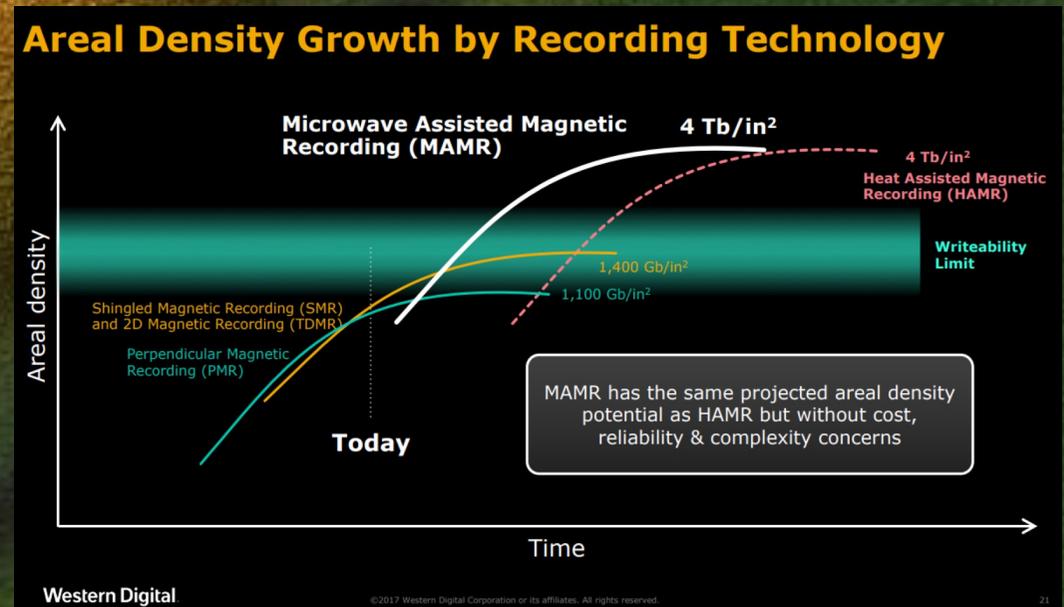
Inizio degli anni '80

Collaborazione Sony/Philips



# Densità delle informazioni *tpi* e *Gbit/in<sup>2</sup>*

- CD: 16000 tpi (vinile: 200-250 e HD 400 tpi)
- CD: 0,90 *Gbit/in<sup>2</sup>*
- DVD: 2,2 *Gbit/in<sup>2</sup>*
- HD DVD: 7,5 *Gbit/in<sup>2</sup>*
- Blu-ray: 12,5 *Gbit/in<sup>2</sup>* (a strato singolo)
- HD Store perpendicolare: centinaia di *Gbit/in<sup>2</sup>*, limite teorico 1 Terabit
- ...



Courtesy of HEPiX TechWatch WG

# Rimescolamento dei dati

A U D I O N E L M U L T I M E D I A L E

# # # # # # E L M U L T I M E D I A L E

E M T L D L L I U M E D I U A O I E N A

# # # # # # L I U M E D I U A O I E N A

A U D I O N E # # U L # I M E # I A # #

# Codifica EFM

## *Eight-to-Fourteen Modulation*

- Minimizzare transizioni 0-1 e 1-0 in lettura
  - Transizione bordi dei pit = 1; riflessione costante = 0
  - Evitare pit di piccole dimensioni (transizioni veloci)
- Permesse solo sequenze di bit che presentano un numero di 0 consecutivi (da 2 a 10)
- Codici di 8 bit diventano di 14 bit
- Esempio
  - 8 bit 11011111 in 14 bit 00100000010001
  - Tabelle per 256 codici EFM
  - pit che coprono da 3 a 11 bit (2-10 0 consecutivi)

# Cross Interleave Reed-Solomon Code

2 campioni

$L_1 = 0111\ 0000\ 1010\ 1000$   
 $R_1 = 1100\ 0111\ 1010\ 1000$

Frame di 12 campioni

$L_1\ R_1\ L_2\ R_2\ L_3\ R_3\ L_4\ R_4\ L_5\ R_5\ L_6\ R_6$

24 byte  
192 bit

$L_{1a}\ L_{1b}\ R_{1a}\ R_{1b}\ L_{2a}\ L_{2b}\ R_{2a}\ R_{2b}\ L_{3a}\ L_{3b}\ R_{3a}\ R_{3b}$   
 $L_{4a}\ L_{4b}\ R_{4a}\ R_{4b}\ L_{5a}\ L_{5b}\ R_{5a}\ R_{5b}\ L_{6a}\ L_{6b}\ R_{6a}\ R_{6b}$

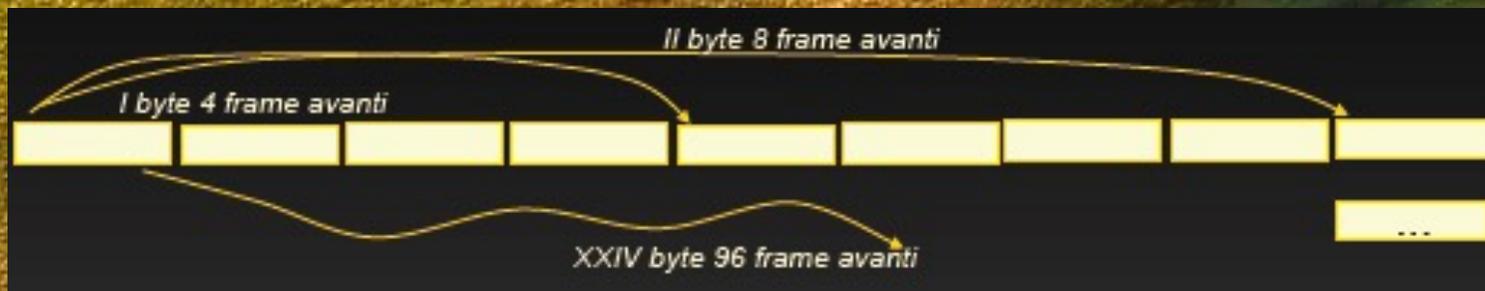
Byte pari ritardati di 2 frame +  
Rimescolamento

$R_{6b}\ L_{5a}\ L_{4a}\ L_{3a}\ L_{5b}\ R_{5a}\ R_{1b}\ L_{4b}\ R_{2a}\ R_{6a}\ L_{6b}\ R_{4a}$   
 $R_{1a}\ R_{2b}\ R_{3b}\ R_{5b}\ L_{1b}\ L_{2a}\ L_{2b}\ R_{4b}\ L_{6a}\ R_{3a}\ L_{3b}\ L_{1a}$

+ 4 byte di parità ... 28 byte  
224 bit

$R_{6b}\ L_{5a}\ L_{4a}\ L_{3a}\ L_{5b}\ R_{5a}\ R_{1b}\ L_{4b}\ R_{2a}\ R_{6a}\ L_{6b}\ R_{4a}$   
 $R_{1a}\ R_{2b}\ R_{3b}\ R_{5b}\ L_{1b}\ L_{2a}\ L_{2b}\ R_{4b}\ L_{6a}\ R_{3a}\ L_{3b}\ L_{1a}$   
 $Q_1\ Q_2\ Q_3\ Q_4$

# CIRC (2)



Nuova codifica di parità, totale 32 byte  
256 bit

$R_{6b} L_{5a} L_{4a} L_{3a} L_{5b} R_{5a} R_{1b} L_{4b} R_{2a} R_{6a} L_{6b} R_{4a}$   
 $R_{1a} R_{2b} R_{3b} R_{5b} L_{1b} L_{2a} L_{2b} R_{4b} L_{6a} R_{3a} L_{3b} L_{1a}$   
 $Q_1 Q_2 Q_3 Q_4$   
 $P_1 P_2 P_3 P_4$

Ulteriore spostamento dei byte  
dispari di un singolo blocco  
+ 8 bit (1 byte) di info ...  
... totale 33 byte / 264 bit per frame!!!

$I R_{6b} L_{5a} L_{4a} L_{3a} L_{5b} R_{5a} R_{1b} L_{4b} R_{2a} R_{6a} L_{6b} R_{4a}$   
 $R_{1a} R_{2b} R_{3b} R_{5b} L_{1b} L_{2a} L_{2b} R_{4b} L_{6a} R_{3a} L_{3b} L_{1a}$   
 $Q_1 Q_2 Q_3 Q_4$   
 $P_1 P_2 P_3 P_4$

# Informazioni e totale

33 byte x 8 = 264 bit



33 byte x 14 = 462 bit

*I R<sub>6b</sub> L<sub>5a</sub> L<sub>4a</sub> L<sub>3a</sub> L<sub>5b</sub> R<sub>5a</sub> R<sub>1b</sub> L<sub>4b</sub> R<sub>2a</sub> R<sub>6a</sub> L<sub>6b</sub> R<sub>4a</sub>  
R<sub>1a</sub> R<sub>2b</sub> R<sub>3b</sub> R<sub>5b</sub> L<sub>1b</sub> L<sub>2a</sub> L<sub>2b</sub> R<sub>4b</sub> L<sub>6a</sub> R<sub>3a</sub> L<sub>3b</sub> L<sub>1a</sub>  
Q<sub>1</sub> Q<sub>2</sub> Q<sub>3</sub> Q<sub>4</sub>  
P<sub>1</sub> P<sub>2</sub> P<sub>3</sub> P<sub>4</sub>*

*I(14)*

*R<sub>6b</sub>(14) L<sub>5a</sub>(14) L<sub>4a</sub>(14) L<sub>3a</sub>(14) L<sub>5b</sub>(14) R<sub>5a</sub>(14) R<sub>1b</sub>(14) L<sub>4b</sub>(14)  
R<sub>2a</sub>(14) R<sub>6a</sub>(14) L<sub>6b</sub>(14) R<sub>4a</sub>(14)*

*R<sub>1a</sub>(14) R<sub>2b</sub>(14) R<sub>3b</sub>(14) R<sub>5b</sub>(14) L<sub>1b</sub>(14) L<sub>2a</sub>(14) L<sub>2b</sub>(14) R<sub>4b</sub>(14) L<sub>6a</sub>(14)  
R<sub>3a</sub>(14) L<sub>3b</sub>(14) L<sub>1a</sub>(14)*

*Q<sub>1</sub>(14) Q<sub>2</sub>(14) Q<sub>3</sub>(14) Q<sub>4</sub>(14)*

*P<sub>1</sub>(14) P<sub>2</sub>(14) P<sub>3</sub>(14) P<sub>4</sub>(14)*

# Bit di synch

*L<sub>1a</sub> L<sub>1b</sub> R<sub>1a</sub> R<sub>1b</sub> L<sub>2a</sub> L<sub>2b</sub> R<sub>2a</sub> R<sub>2b</sub> L<sub>3a</sub> L<sub>3b</sub> R<sub>3a</sub> R<sub>3b</sub>  
L<sub>4a</sub> L<sub>4b</sub> R<sub>4a</sub> R<sub>4b</sub> L<sub>5a</sub> L<sub>5b</sub> R<sub>5a</sub> R<sub>5b</sub> L<sub>6a</sub> L<sub>6b</sub> R<sub>6a</sub> R<sub>6b</sub>*

462 +

24 bit

... + 24 bit di synch = 486 bit x frame

100000000001000000000010

*I(14)*

*R<sub>6b</sub>(14) L<sub>5a</sub>(14) L<sub>4a</sub>(14) L<sub>3a</sub>(14) L<sub>5b</sub>(14) R<sub>5a</sub>(14) R<sub>1b</sub>(14) L<sub>4b</sub>(14) R<sub>2a</sub>(14)  
R<sub>6a</sub>(14) L<sub>6b</sub>(14) R<sub>4a</sub>(14)*

*R<sub>1a</sub>(14) R<sub>2b</sub>(14) R<sub>3b</sub>(14) R<sub>5b</sub>(14) L<sub>1b</sub>(14) L<sub>2a</sub>(14) L<sub>2b</sub>(14) R<sub>4b</sub>(14) L<sub>6a</sub>(14)  
R<sub>3a</sub>(14) L<sub>3b</sub>(14) L<sub>1a</sub>(14)*

*Q<sub>1</sub>(14) Q<sub>2</sub>(14) Q<sub>3</sub>(14) Q<sub>4</sub>(14)*

*P<sub>1</sub>(14) P<sub>2</sub>(14) P<sub>3</sub>(14) P<sub>4</sub>(14)*

# 3 bit di intercapedine per codici EFM

192 bit in partenza

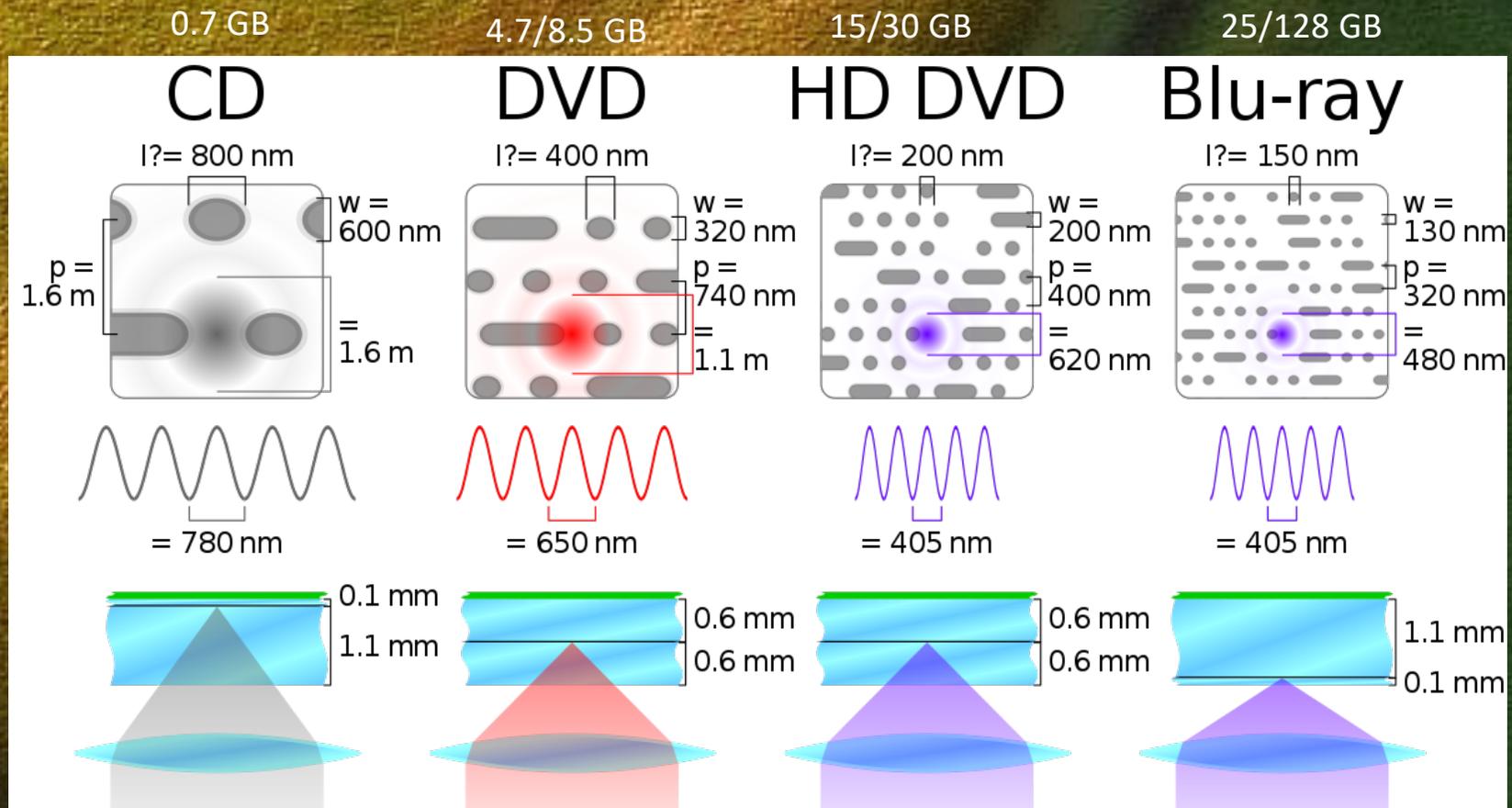
33 byte ... 32 intercapedini + 2 agli estremi ... totale 34

34 intercapedini x 3 bit = 102 bit Totale 486 + 102 = 588 bit per frame

Più del triplo!!!

Tipo di informazione	Bit × unità	Numero di unità	Totale in bit
Campioni audio	28 (14 × 2)	12	336
Sotto-codice di info agg.	14	1	14
Byte di parità	14	8	112
Bit di intercapedine	3	34	102
Bit di sincronizzazione	24	1	24
<b>Totale generale</b>			<b>588</b>

# Confronto su optical storage



[fonte: Wikipedia]



Grazie dell'attenzione