



**UNIVERSITÀ DEGLI STUDI DI ROMA  
TOR VERGATA**

**PROJECT WORK**

**TECNICHE DI IMPLEMENTAZIONE BASATE SULLA PARALLELIZZAZIONE PARZIALE  
PER LA DECODIFICA DEI CODICI LOW DENSITY PARITY CHECK E  
REALIZZAZIONE DI TABELLE DI SOFT-DEMODULAZIONE PER I DIVERSI CASI  
DI MODULAZIONE IN UN RICEVITORE DVBS 2.**

**SUPERVISORI**

**Francesco Richichi**

**Giuseppe Chiassarini**

**CANDIDATO**

**Nico Saraceno**

# DVB-S2 Generalità

Nel 2003, a dieci anni dalla nascita del DVB, è stato standardizzato il sistema di seconda generazione per la trasmissione via satellite, DVB-S2, erede del sistema di diffusione della televisione digitale da satellite DVB-S, attualmente usata dalla maggior parte degli operatori satellitari nel mondo

Il sistema DVB-S2 è stato progettato per varie applicazioni :

- servizi diffusivi di TV a definizione standard (**SDTV**) e ad alta definizione (**HDTV**);
- applicazioni **interattive** per l'utenza domestica e professionale, compreso l'accesso a Internet;
- servizi professionali di contribuzione **TV e SNG** (*Satellite News Gathering*);
- distribuzione di dati e di siti Internet (*Internet trunking*);

# Sottosistemi

- Il sistema DVB-S2 è composto da una sequenza di blocchi che svolgono determinate funzioni:
  - Mode adaptation
  - Stream adaptation
  - **Forward Error Correction (FEC) Encoding**
  - **Mapping**
  - Physical layer framing
  - Base-Band Filtering and Quadrature Modulation

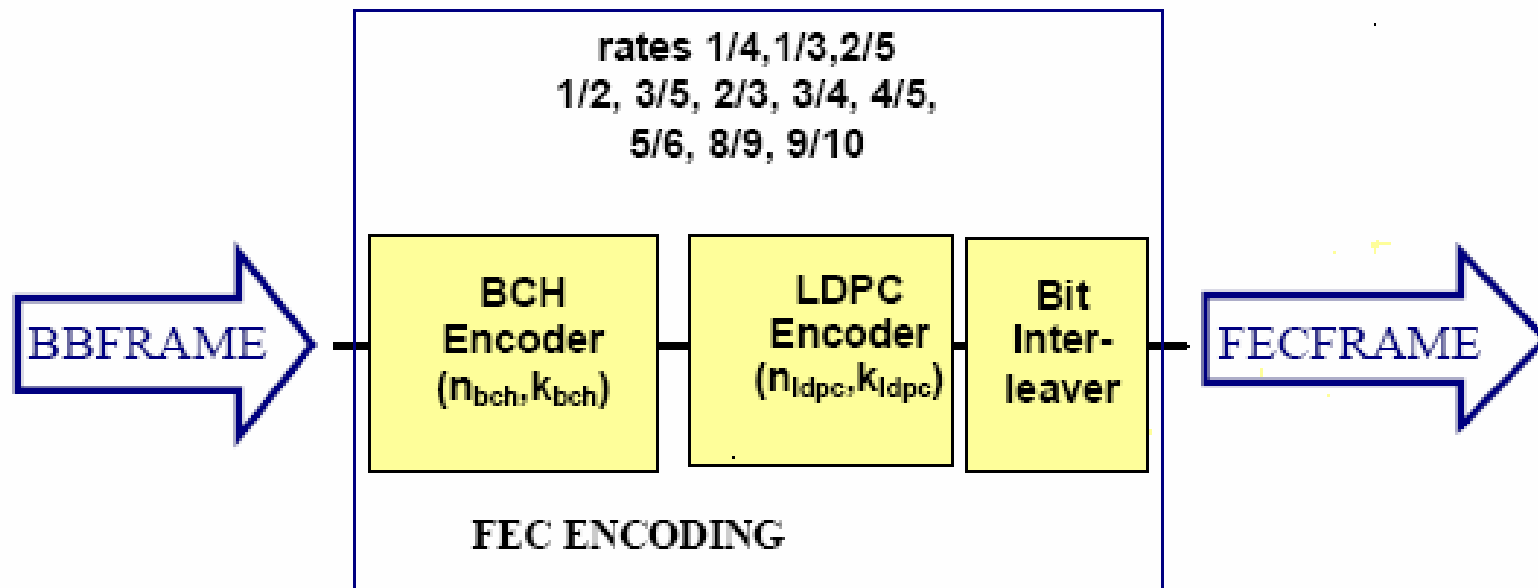
# Struttura della TRAMA

La trama del segnale è suddivisa in due livelli:

- il primo, a livello fisico (**PL**, *Physical Layer*), che trasporta pochi bit di segnalazione molto protetti;
- il secondo, a livello di banda base (**BB**, *BaseBand*), che trasporta molti bit di segnalazione, per consentire la massima flessibilità di adattamento del segnale di ingresso.

# Forward Error Correction

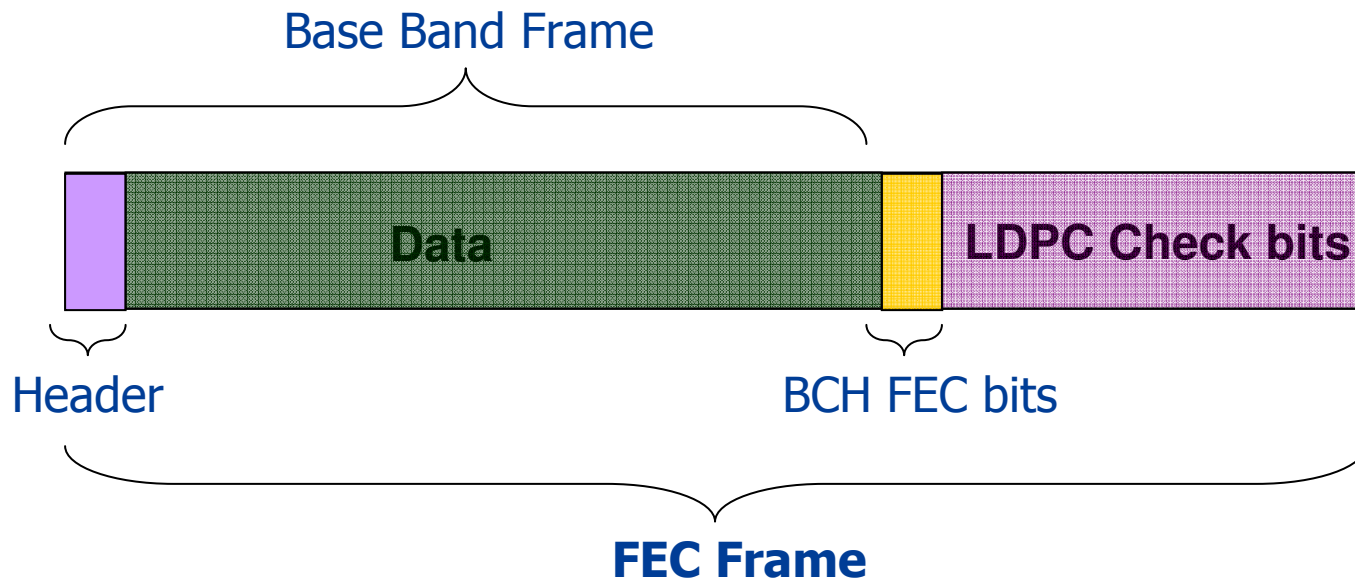
- Il sistema FEC permette di **correggere gli errori al ricevitore**, e quindi di recuperare un segnale perfetto anche con rapporti segnale rumore molto bassi



# FEC Frame

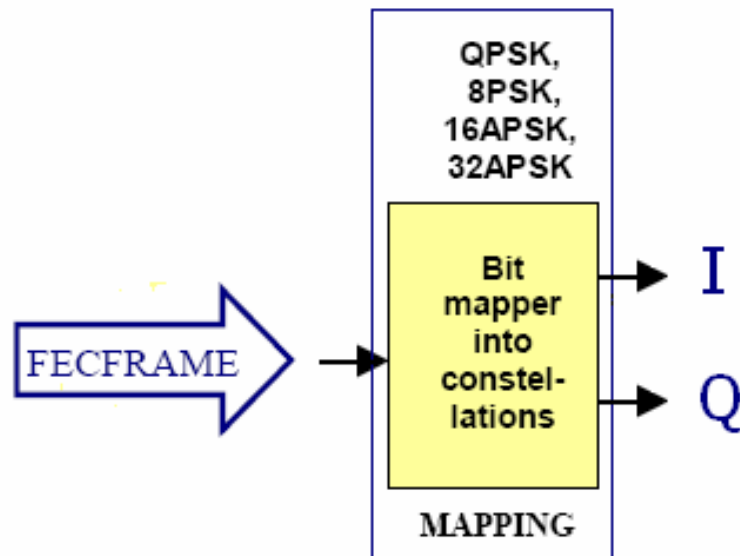
Esistono 2 versioni di trama:

- **Long** Frame (64800 bits) – Migliori prestazioni vs maggiore ritardo globale della catena trasmissiva (buono per applicazioni come diffusione programmi)
- **Short** Frame (16200 bits) – Può essere immediatamente messo in onda dalla stazione trasmittente (buono per applicazioni interattive)



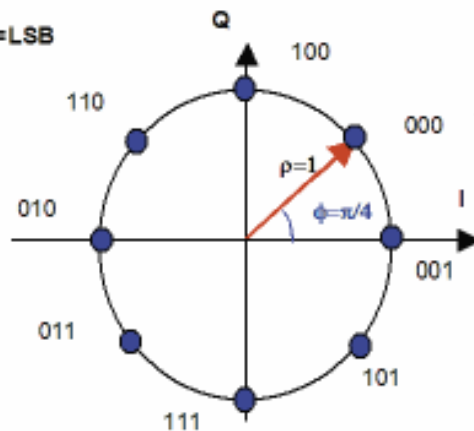
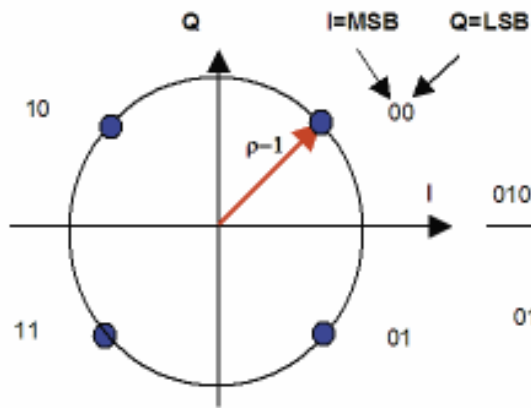
# Mapping

- Ogni **FECFRAME** ( flusso di bit codificato dal codice LDPC/BCH) può essere mappato in 4 differenti costellazioni: QPSK / 8-PSK / 16-APSK / 32-APSK

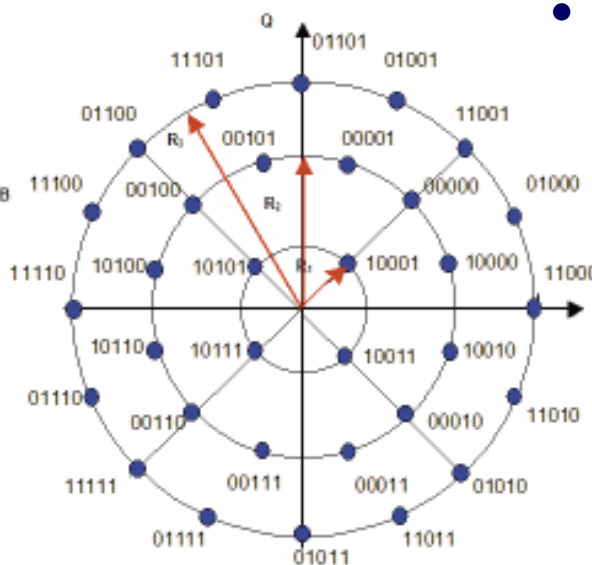
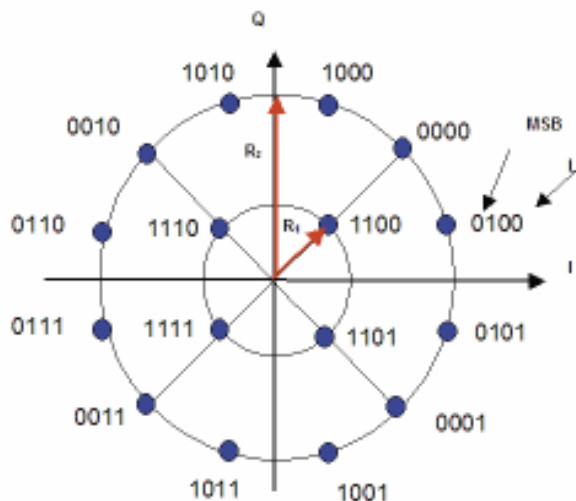


La sequenza d'uscita sarà una **XFECFRAME** (FECFRAME complessa): composta da un numero di simboli pari a 64 800 (per il long frame, 16200 nel caso short frame) diviso il numero di bit per simbolo della costellazione.

# Costellazioni DVB-S2



- Le costellazioni **QPSK** e **8PSK** sono usate per applicazioni broadcast



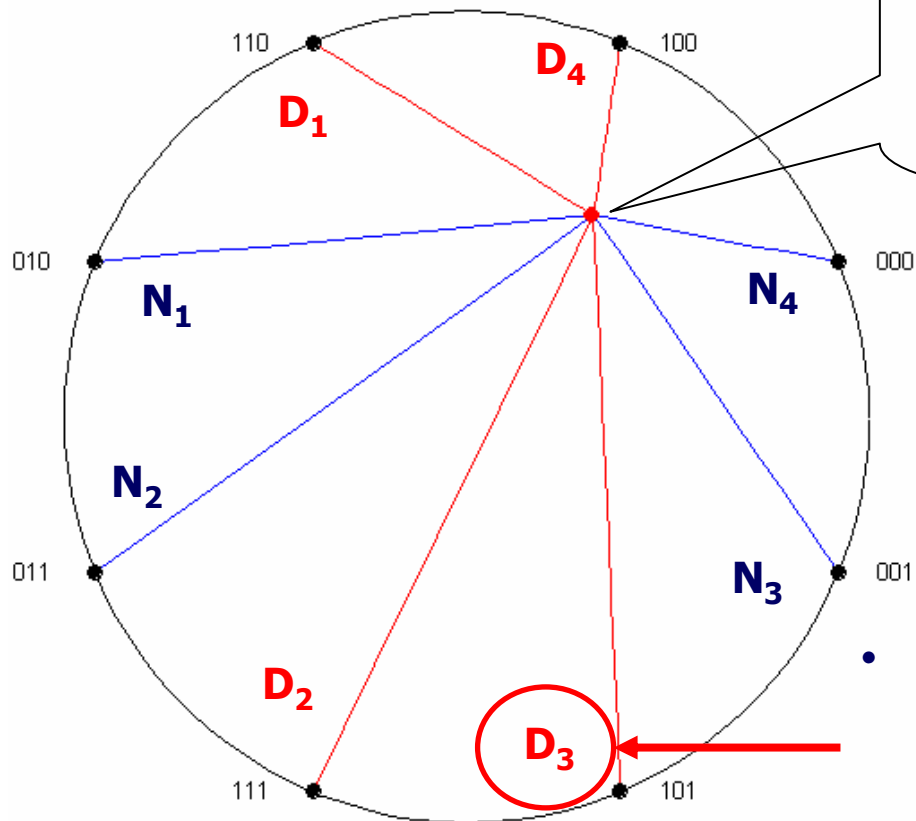
- Le costellazioni **16-APSK** e **32-APSK** sono usate per applicazioni professionali

# Log Likelihood Ratio

- È la strategia di rilevazione **soft decision** che fornisce la “misura” di quanto il campione del segnale ricevuto è vicino al simbolo della costellazione.
- Il ricevitore con ingressi ed uscite *soft* calcola i logaritmi di verosimiglianza indicati con l'acronimo **LLR** (*Log-Likelihood Ratio*) dei bit in ingresso al codificatore.
- I logaritmi di verosimiglianza assumono valori positivi o negativi quando il simbolo da decidere è rispettivamente uno 0 o un 1, ed il modulo di un LLR è proporzionale all'**affidabilità** della decisione.
- Il criterio deriva dalla minimizzazione della probabilità di errore sul bit  $b_i$  di una sequenza  $\mathbf{b}$  trasmessa, condizionata al campione  $\mathbf{Y}$  del segnale ricevuto.
- Nel caso di canale **AWGN** il logaritmo di verosimiglianza si ottiene andando a misurare la distanza di  $\mathbf{Y}$  rispetto ai simboli  $\mathbf{X}(\mathbf{b})$  della costellazione o valori del segnale trasmesso libero da rumore.

# In Ricezione

## Esempio 8PSK:



Per un singolo campione  $Y$  del segnale ricevuto si generano tanti valori di *soft decision* quanti sono i bit per simbolo della costellazione

- Nell'esempio è rappresentato il solo caso del bit più significativo

$$L_{b_1/y} = \log \frac{\sum_j N_j}{\sum_j D_j}$$

- Ciascun termine dipende dalla distanza euclidea di  $Y$  da un simbolo della costellazione e dalla frequenza del codice LDPC mediante l'efficienza spettrale.

# Considerazioni

- La funzione densità di probabilità di  $\mathbf{Y}$  condizionata al simbolo  $\mathbf{X}(\mathbf{b})$  trasmesso è una Gaussiana con media pari al valore di  $\mathbf{X}(\mathbf{b})$

$$L_{b_1/y} = \log \frac{\sum_j N_j}{\sum_j D_j} \quad N_j = f(y | \mathbf{x}(\mathbf{b})) = \frac{1}{\sqrt{2\pi\sigma}} \exp \left\{ -\frac{E_s}{N_0} \left[ (y_I - x_I(\mathbf{b}))^2 + (y_Q - x_Q(\mathbf{b}))^2 \right] \right\}$$

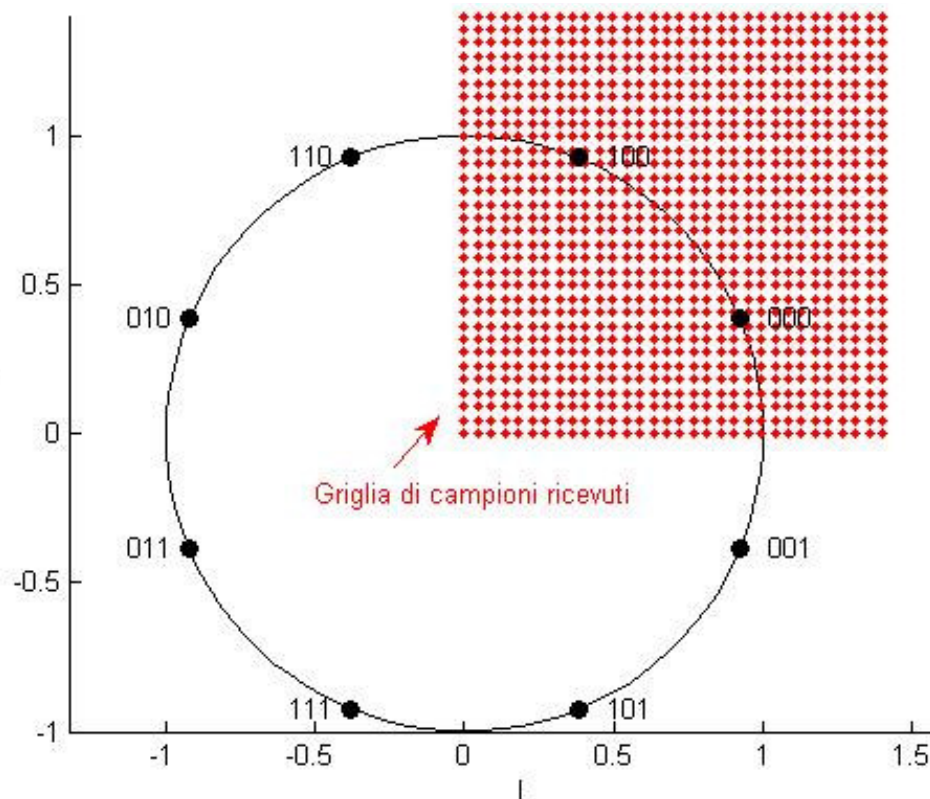
Dipende dalla frequenza del codice LDPC mediante l'efficienza spettrale



Per la stessa costellazione le tabelle variano con il code rate

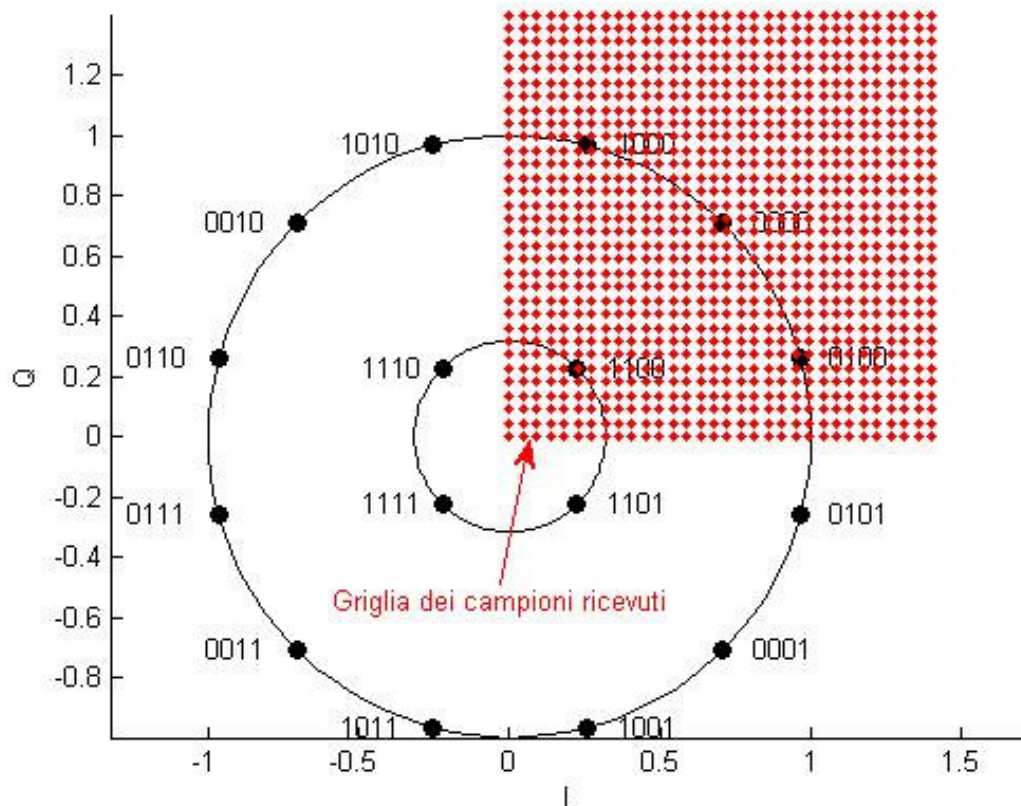
# 8-PSK

Da un campione a una griglia di campioni



- Come scelta di sistema, il ricevitore usa una tabella con i valori precalcolati (**LUT**) dei soft bit associati a ciascun campione del segnale ricevuto
- È necessario ricoprire una zona significativa della costellazione. In conseguenza al **labelling di Gray** è possibile sfruttare la simmetria dei bit di simbolo e ricoprire solo il primo quadrante della 8 PSK.

# 16-APSK



La 16-APSK è costituita da 2 costellazioni, una QPSK interna e una 12-PSK esterna. Il rapporto tra il raggio esterno e quello interno varia con il **code rate** ed è dato da:

$$\gamma = \frac{R_{esterno}}{R_{interno}}$$

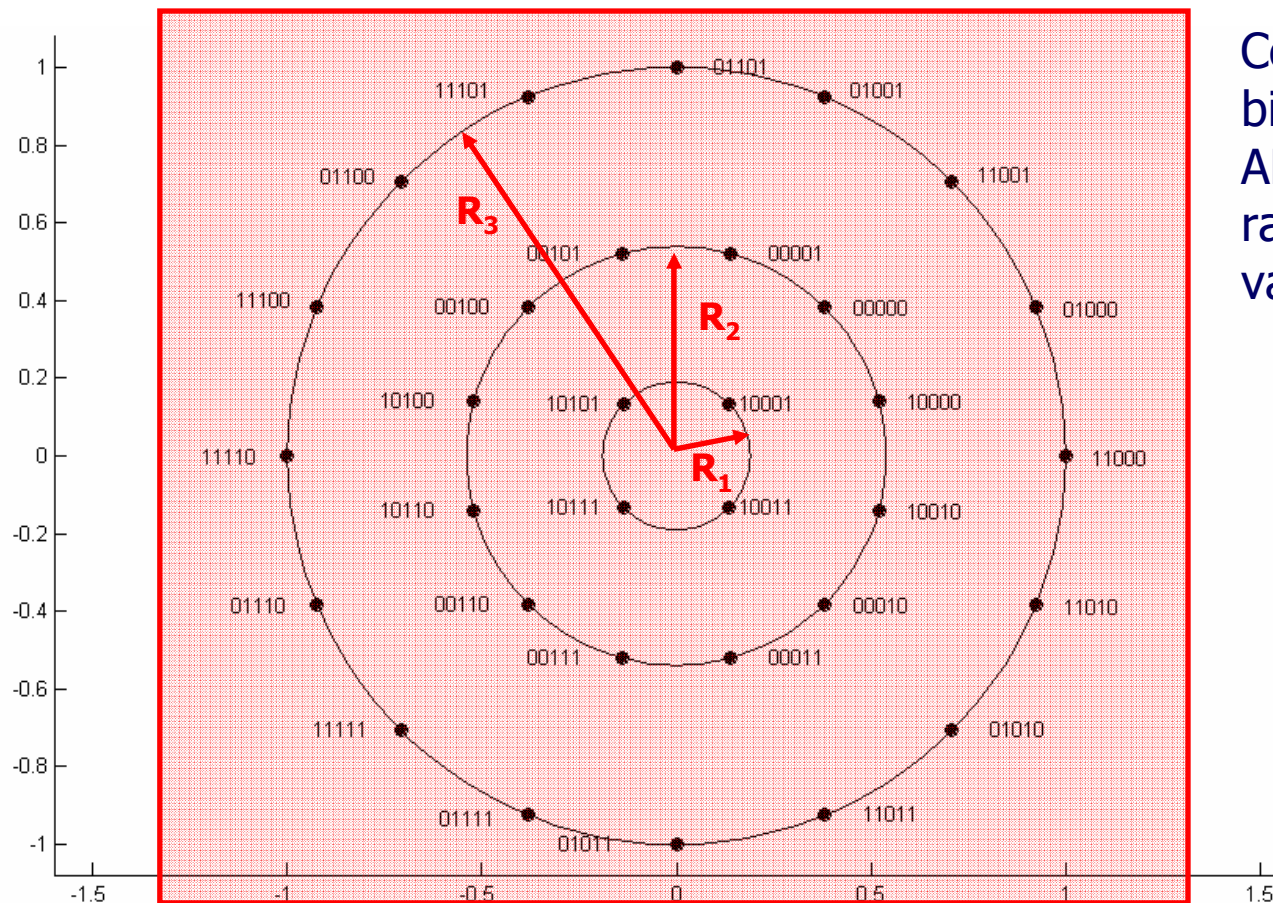
- Per ogni campione ricevuto ho 4 valori di soft-bit
- Per ogni griglia ho 4 matrici
- Sfruttando la simmetria dei bit di simbolo posso coprire solo il primo quadrante

# 32-APSK

Come per la 16-APSK, bisogna generare la 32-APSK tenendo conto dei rapporti tra i raggi variabili con il **code-rate**:

$$\gamma_1 = \frac{R_2}{R_1}, \gamma_2 = \frac{R_3}{R_1}$$

Zona di copertura per il calcolo dei soft-bit è tutta la costellazione 32-APSK

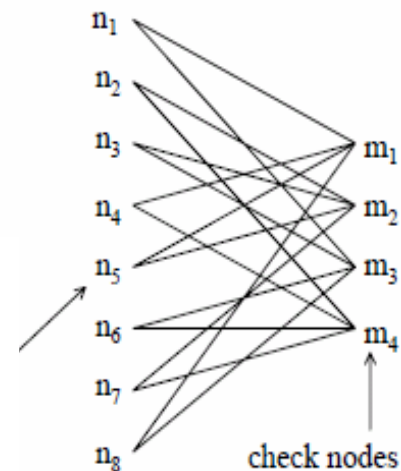


# Low Density Parity Check

- I codici LDPC sono codici **lineari a blocchi** la cui caratteristica è quella di avere una matrice di controllo di parità **H** che sia **sparsa**, ovvero che sia composta da un numero di "1" sensibilmente minore del numero di "0".
- La descrizione del codice e dell'algoritmo di decodifica utilizzato per i codici LDPC è facilitata dall'introduzione di un grafo, detto **Tanner graph**, che viene definito a partire dalle equazioni di parità che si ricavano dalla matrice di controllo di parità **H**.
- In un decodificatore soft, piuttosto che operare con una hard decision, si propagano le probabilità attraverso il Tanner graph, così da accumulare l'evidenza che i checks forniscono riguardo i valori dei bits

$$H = \begin{array}{cccccccc|c} n_1 & n_2 & n_3 & n_4 & n_5 & n_6 & n_7 & n_8 & \\ \hline 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & m_1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & m_2 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & m_3 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & m_4 \end{array}$$

Bit nodes  
Rappresentano  
i bit trasmessi  
di una  
codeword

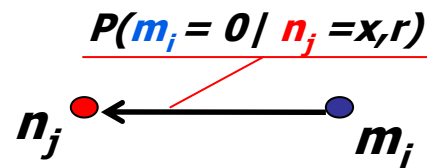


Check nodes  
 $m_i$  sono le  
equazioni di  
parità che  
riguardano i  
nodi  $n_i$

# Decodifica

La **decodifica** può essere suddivisa in 3 step di cui gli ultimi 2 sono iterativi:

- 1. INIZIALIZZAZIONE:** All'informazione inviata dai bit nodes ai check nodes siano *inizialmente* assegnati i valori stimati delle probabilità a priori dei simboli ricevuti.
- 2. CHECK NODE UPDATE:** Ogni nodo di parità  $m_i$  calcola ed invia a ciascun nodo di simbolo  $n_j$  cui è collegato una stima della probabilità che l'i-esima equazione di controllo di parità sia soddisfatta se il j-esimo nodo di simbolo è nello stato  $\mathbf{x} \in \{0,1\}$ , tenendo conto dell'informazione fornitagli da tutti gli altri nodi di simbolo collegati ad esso.



Dove  $\mathbf{r}$  è il vettore ricevuto

- 3. BIT NODE UPDATE:** Ogni nodo di simbolo  $n_j$  calcola ed invia ad ogni nodo di controllo di parità  $m_i$ , cui è collegato, la probabilità che il nodo di controllo di parità i-esimo sia nella stato  $\mathbf{x}$ .

$$P(n_j = x | r, \text{tutti i check-nodes, escluso } m_i, \text{ sono soddisfatti dal bit } n_j)$$



# Parallelizzazione Parallela

- Nell'algoritmo di decodifica i bit nodes e i check nodes aggiornano (leggere/processare/scrivere) i valori delle loro connessioni adiacenti in maniera iterativa.
- Nell'assunzione di nessuna restrizione alla matrice di controllo di parità  $H$ , le connessioni sono **distribuite a caso** nella memoria. In tale scenario, accedere ai contenuti di  $n$  connessioni implica  $n$  cicli di clock.
- Nell'algoritmo della **parallelizzazione parziale**, un certo numero di nodi ( $M$ ) viene processato in parallelo, ciò equivale ad accedere al blocco di memoria che corrisponde alle **connessioni adiacenti** dei nodi "in process".
- Un piccolo numero di **nodi fisici** sono riutilizzati per servire tutti i **nodi logici** in modo da scavalcare la limitazione della complessità introdotta da una architettura totalmente parallela per blocchi di grandi dimensioni.

# Matrice H

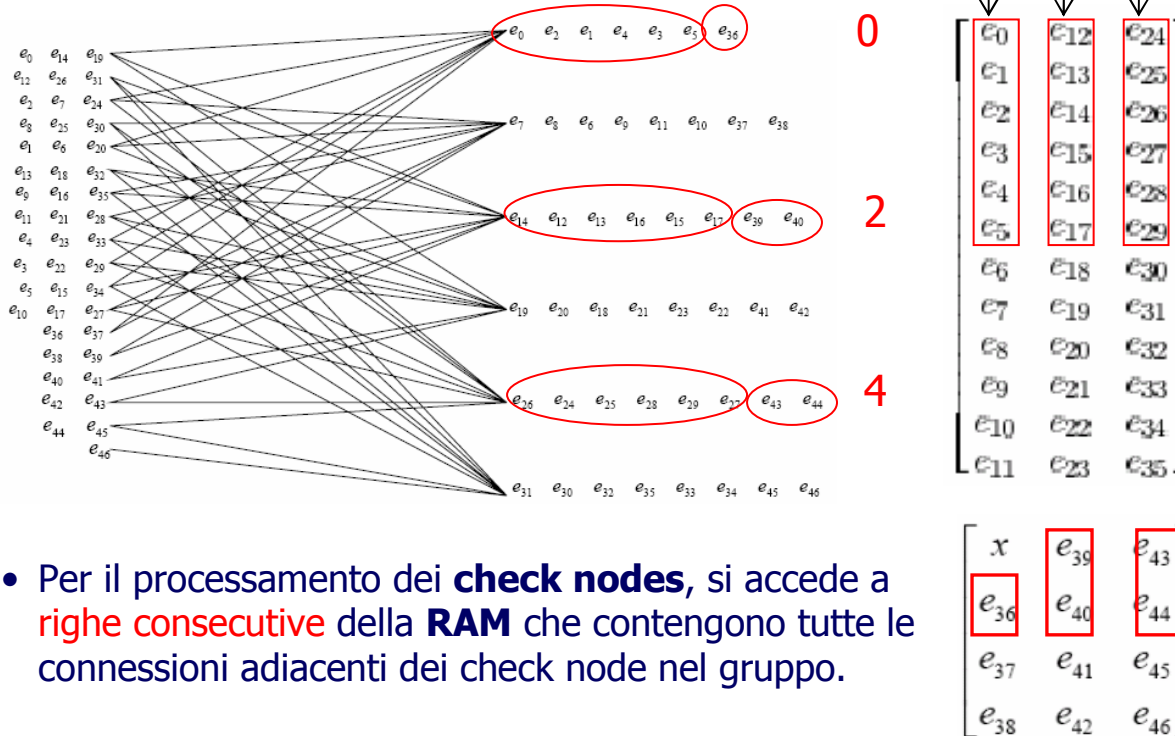
- Il metodo restringe la matrice di controllo di parità ad avere la seguente forma:

$$H = \begin{array}{c}
 \begin{array}{cc}
 \text{Dimensione (N-K)xK} & \text{Dimensione (N-K)x(N-K)} \\
 \left[ \begin{array}{cccccccccccc}
 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0
 \end{array} \right] & \left[ \begin{array}{cccccc}
 1 & 0 & 0 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 1 & 1
 \end{array} \right] \\
 \underbrace{\hspace{10em}}_A & \underbrace{\hspace{5em}}_B
 \end{array}
 \end{array}
 \quad \begin{array}{l}
 \text{B MATRICE} \\
 \text{TRIANGOLARE}
 \end{array}$$

- Numerazione delle connessioni** (ovvero degli "1" in H): Ogni connessione del *Tanner graph* (o grafo bipartito) è associata a un indice.
- I valori delle connessioni sono inserite nella parte superiore (**TOP**) o inferiore (**BOTTOM**) della **RAM delle connessioni** a seconda che le connessioni appartengono rispettivamente alla sottomatrice A o B.

# Posizionamento connessioni

- La **RAM delle connessioni** è divisa in 2: **Top RAM** e **Bottom RAM**
- Posizionamento** delle connessioni adiacenti di un check node a gruppi di **M** nella RAM:  $\{0, 2, 4\}$ ,  $\{1, 3, 5\}$ .



## TOP RAM

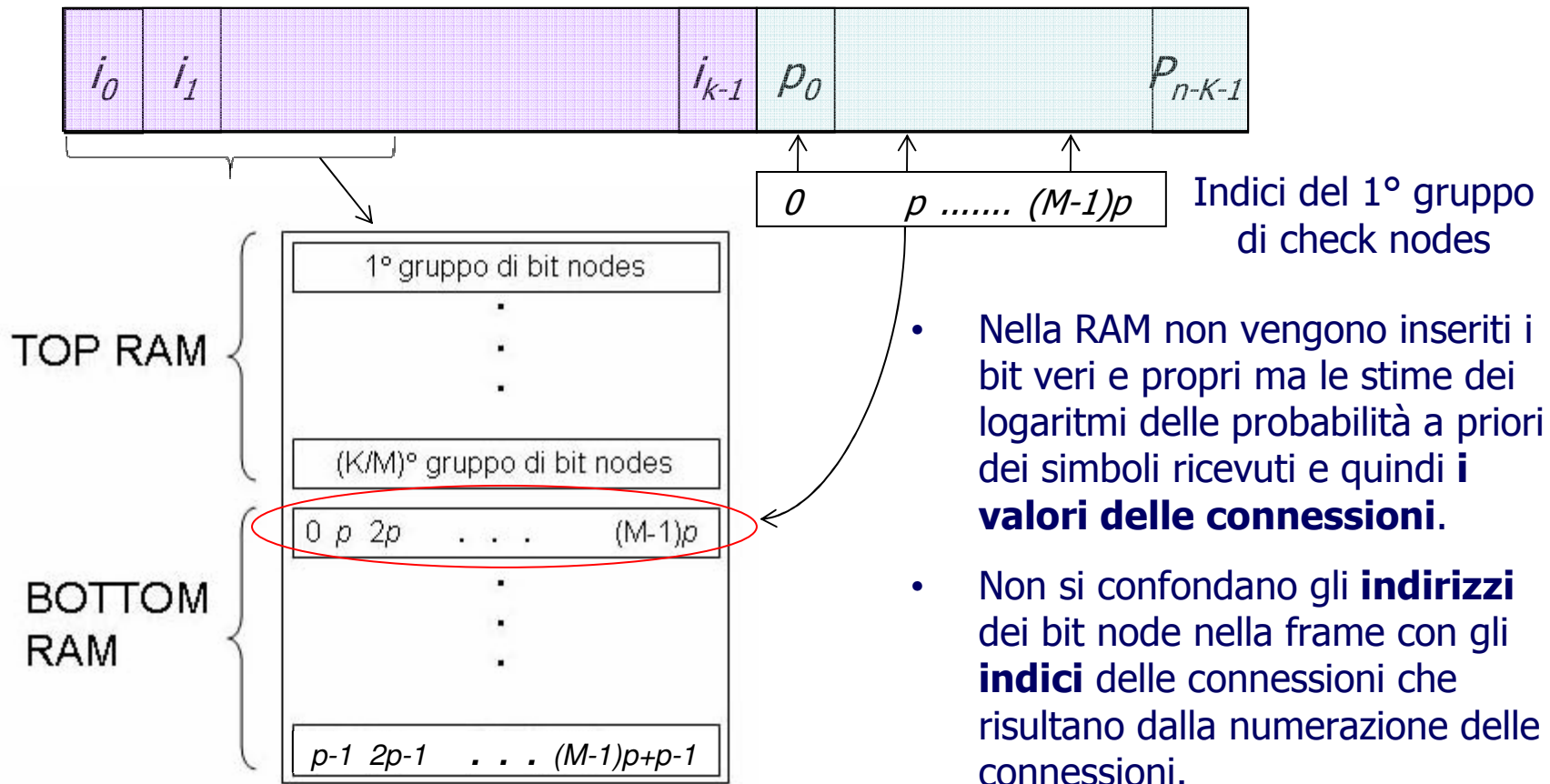
Ogni connessione della top RAM corrisponde a "1" nella **sottomatrice A**.

## BOTTOM RAM

Ogni connessione della Bottom RAM corrisponde a "1" nella **sottomatrice B**.

# Inizializzazione della RAM

## FEC Frame



# Vettori di indici

- **Check nodes update**

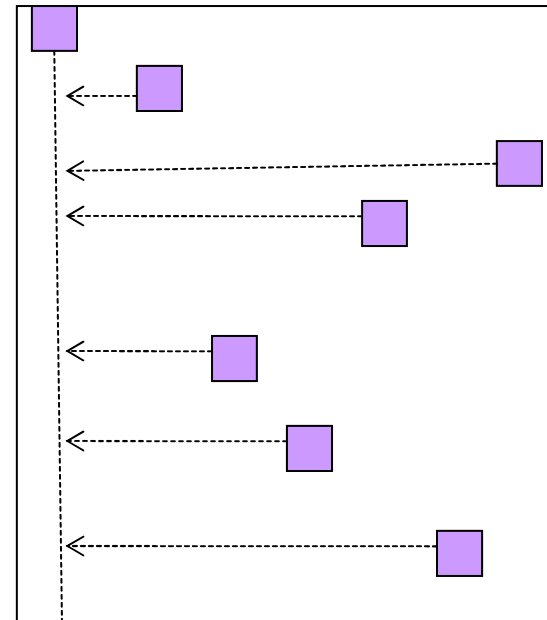
Le connessioni adiacenti dei check node non sono allineate in colonna, perciò bisogna implementare una coppia di indici che riordina le connessioni per poter eseguire il check node update:

- Il vettore di indici di riga che indirizza la RAM
- Il vettore di valori di offset che indica di quanto scivolare circolarmente la riga indirizzata (switch matrix).

In figura sono rappresentate le connessioni adiacenti di un check node, ma il processo è eseguito in **parallelo** per tutti i check nodes.

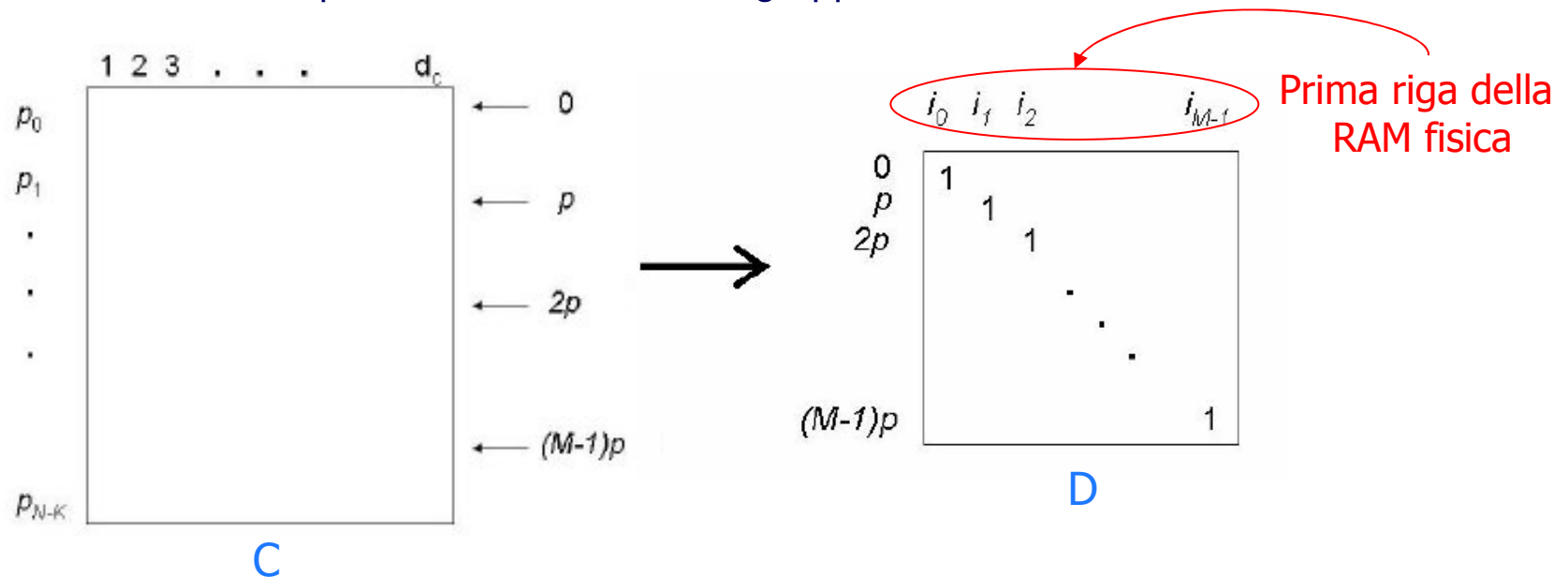
- **Bit nodes update**

Per la disposizione ordinata degli elementi della FEC frame, le connessioni adiacenti dei bit nodes sono in righe consecutive della RAM. È necessario solo il vettore di valori di offset per scivolare la switch matrix.



# Implementazione

- Dall'operazione di codifica si ricava la matrice **C** i cui elementi sono gli indirizzi dei bit nodes associati agli indirizzi dei check nodes opportunamente ordinati.
- Partendo dall'ultima colonna (massimo grado dei check node), si estrae una sottomatrice **D** i cui indici di riga sono gli indirizzi dei check node di un gruppo connessi ai corrispondenti bit nodes di un gruppo.



- L'indirizzo del check node connesso al 1° bit node di un gruppo individua il valore di offset dello switch matrix nelle fasi di check node e bit node update.

# Conclusioni

L'esperienza di stage si è concentrata sullo sviluppo dei seguenti aspetti di un ricevitore **DVB-S2**:

- Calcolo dei coefficienti di Soft-Demodulazione per i soft-bit in ingresso al decodificatore LDPC.
- Calcolo dei vettori di indicizzazione per l'implementazione parallela del decodificatore LDPC.