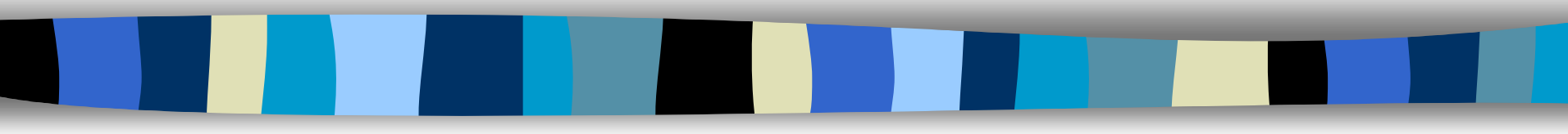
S-72.245 Transmission Methods in Telecommunication System (4cr)

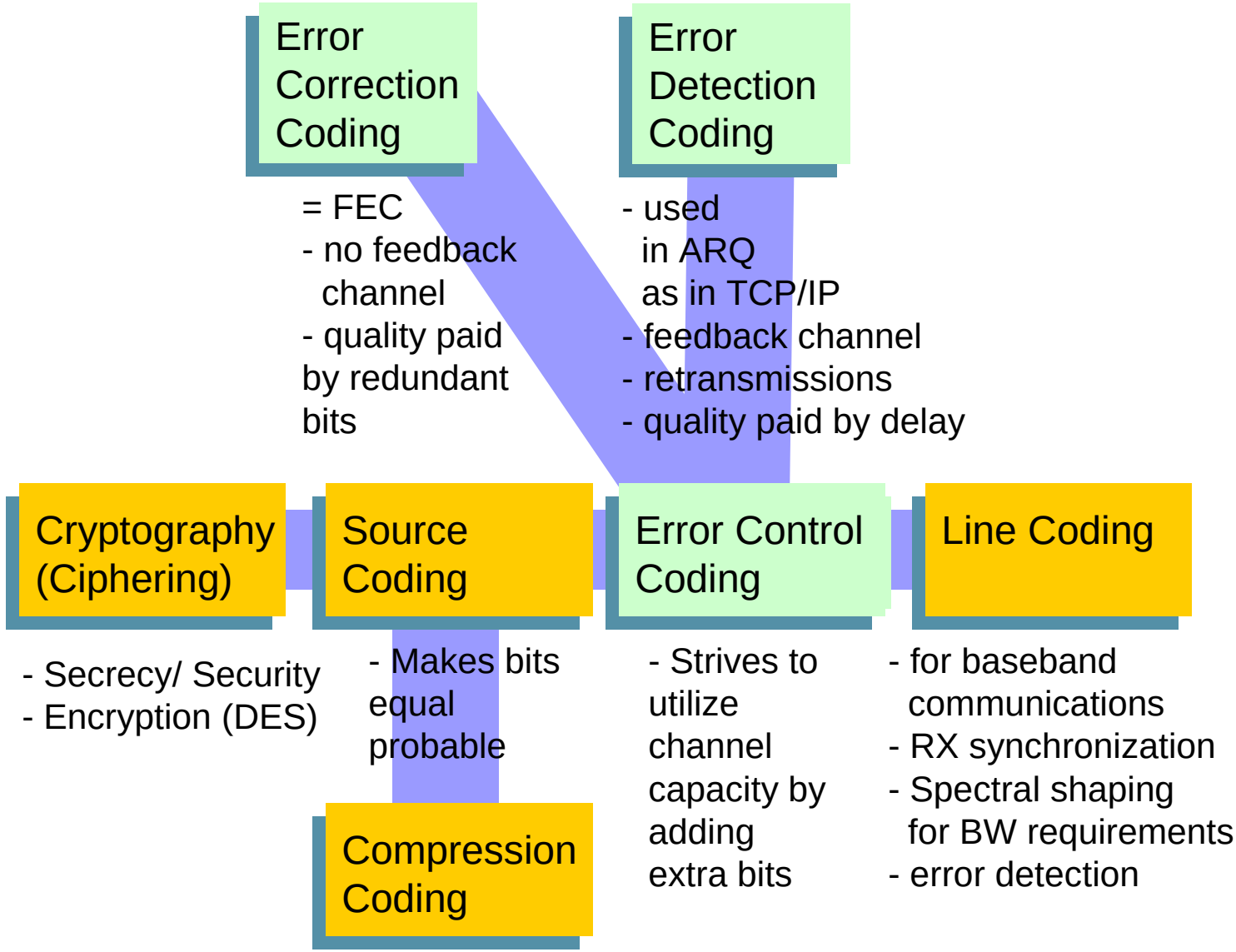


Error Control Coding

Topics Today

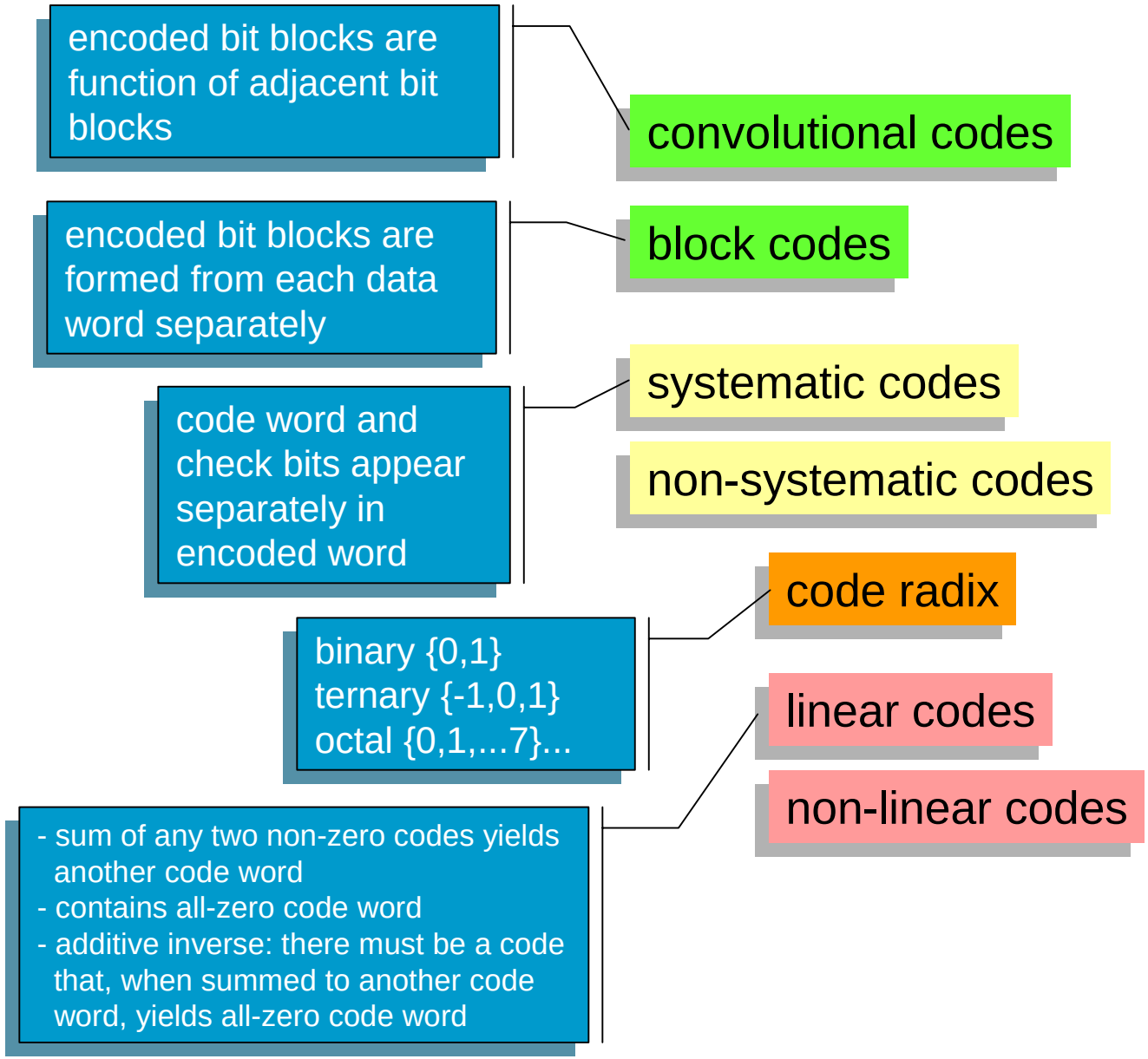
- Codes
  - repetition
  - parity check
  - block codes
- Code vectors
  - Hamming distance
  - error correct and detection capability
- Soft and hard decoding
- Syndrome decoding

Taxonomy of Coding



FEC: Forward Error Correction
ARQ: Automatic Repeat Request
DES: Data Encryption Standard

Forward Error Correction Coding (FEC)



Repetition Coding

- In repetition coding bits are simply repeated several times
- Can be used for error correction or detection
- Assume binomial error distribution for *n* bits:

$$P(i,n) = \binom{n}{i}\alpha^i(1-\alpha)^{n-i} \approx \binom{n}{i}\alpha^i, \alpha << 1$$

- Example: In 3-bit repetition coding encoded word is formed by a simple rule:

$$1 \rightarrow 111 \quad 0 \rightarrow 000$$

- Code is decoded by majority voting, e.g. for instance

$$001 \rightarrow 0, \quad 101 \rightarrow 1$$

- Error in decoding is introduced if all the bits are inverted (a code word is swapped into another code word) or two bits are inverted (code word damaged resulting its location on a wrong side of decoding decision region)

$$P_{we} = P(3,3) + P(2,3) \approx 3\alpha^2 - 2\alpha^3$$

Parity-check Coding

- Note that repetition coding can greatly improve transmission reliability because

$$P_{we} = 3\alpha^2 - 2\alpha^3 << P_e = \alpha, \alpha << 1$$

- However, due to repetition transmission rate is reduced. Here the code rate was 1/3 (that is the ration of the bits to be coded to the encoded bits)
- In parity-check encoding a check bit is formed that indicates number of "1":s in the word to be coded
- Even number of "1":s mean that the the encoded word has always even parity
- Example: Encoding 2-bit words by even parity is realized by

$$00 \rightarrow 000,\ 01 \rightarrow 011$$

$$10 \rightarrow 101,\ 11 \rightarrow 110$$

Note that for error detection encoded word parity is checked (how?)

Parity-check Error Probability

- Note that the error is not detected if even number of errors have happened
- Assume *n-1* bit word parity encoding (the encoded word has thus *n* bits) and consider the probability to have 2 bit errors only (this would mean that the error would not be revealed eg error would be produced to decoder output)

$$\alpha << 1 \Rightarrow P_{we} \approx P(2,n) \approx n(n-1)\alpha^2/2$$

- Without error correction, an *n-1*-bit word will have a decoding error with probability

$$P_{uwe} = 1 - \underbrace{P(0,n-1)}_{\text{prob. to have no errors}} = 1 - \frac{(n-1)!}{(n-1)!}\alpha^0(1-\alpha)^{n-1} = 1-(1-\alpha)^{n-1}$$

$$\approx \alpha(n-1)$$

$1-(1-\alpha)^3$ expand $\rightarrow 3\cdot\alpha - 3\cdot\alpha^2 + \alpha^3$

$1-(1-\alpha)^7$ expand $\rightarrow 7\cdot\alpha - 21\cdot\alpha^2 + 35\cdot\alpha^3 - 35\cdot\alpha^4 + 21\cdot\alpha^5 - 7\cdot\alpha^6 + \alpha^7$

Parity-check Error Probability (cont.)

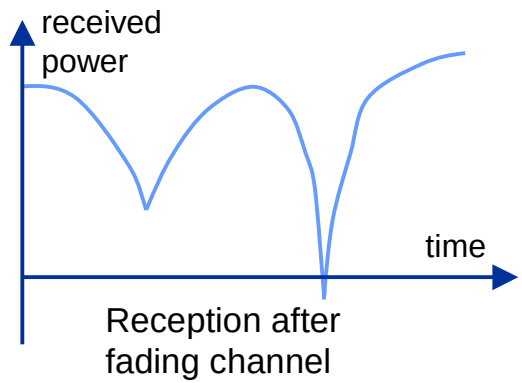
- Hence we note that parity checking is a very efficient method of error detection: Example:

$$n = 10,\ \alpha = 10^{-3},\ p_{uwe} \approx 10^{-2},\ p_{we} \approx 5 \times 10^{-5}$$

- At the same time information rate was reduced by 9/10 only
- Most telecommunication channels are memoryless AWGN channels or fading multipath channels
- In **memoryless channels** bits are considered to be *independent*
- In **fading multipath channels** transfer function changes as a function of time
  - Interleaving can be used to make bits more independent This can be realized by
    - block interleaving
    - convolutional interleaving
- Problem of interleaving is delay that is 50% smaller for convolutional interleavers. Also their memory requirement is 50% smaller.

Block Interleaving

- In fading channels received data can experience burst errors that destroy large number of consecutive bits. This is harmful for channel coding
- Interleaving distributes **burst errors** along data stream
- A problem of interleaving is introduced extra delay
- Example below shows block interleaving:



Received interleaved data: **1 0 0 0 1 1 1 0 1 0 1 1 1 0 0 0 1 1 0 0 1**

Block deinterleaving :

1 0 0 0 1 1 1
0 1 0 1 1 1 0
0 0 1 1 0 0 1

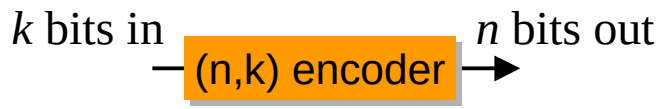
Recovered data: **1 0 0 0 1 0 0 0 1 0 1 1 1 1 0 1 1 0 1 0 1**

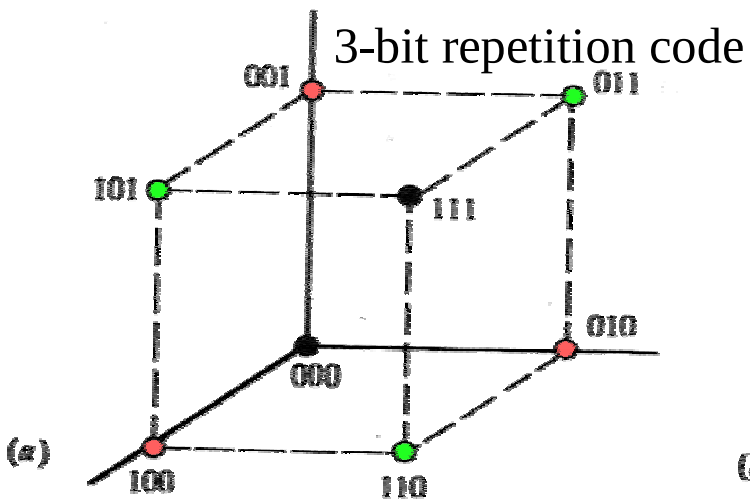
Representing Codes by Vectors

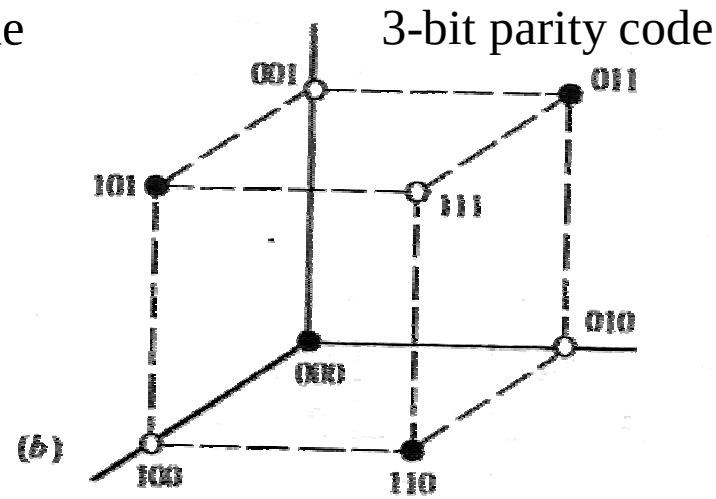
- Hamming distance *d(X,Y)* is the number of bits that are different in code words

$$X = (1\,0\,1), Y = (1\,1\,0) \Rightarrow d(X,Y) = 2$$

- Code strength is measured by minimum Hamming distance:
  - Codes are more powerful when their minimum Hamming distance d_{min} (over all codes in the code family) is as large as possible
- Codes can be mapped into *n*-dimensional grid:







Hamming Distance and Code Error Detection and Correction Capability

- Channel noise can produce non-valid code words that are detected at the receiver
- Number of non-valid code words depends on d_{min}, and consequently the number of errors that can be detected at the reception is $l = d_{\min} - 1$

  If l +1 errors produced (for instance by noise/interference), received code word is transformed into another code word
- The number of errors that can be corrected is

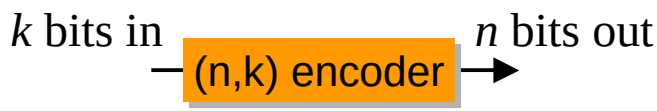
  $t = \lfloor l/2 \rfloor$ $\lfloor \ \rfloor$ (denotes the integer part)

  If more bit errors than t is produced, maximum likelihood detection can not decide correctly in which decision region received code belongs

Code Efficiency and the Largest Minimum Distance

- The **largest possible minimum distance** is achieved by repetition codes that is

$$d_{\min}\Big|_{\max} = n - k + 1$$



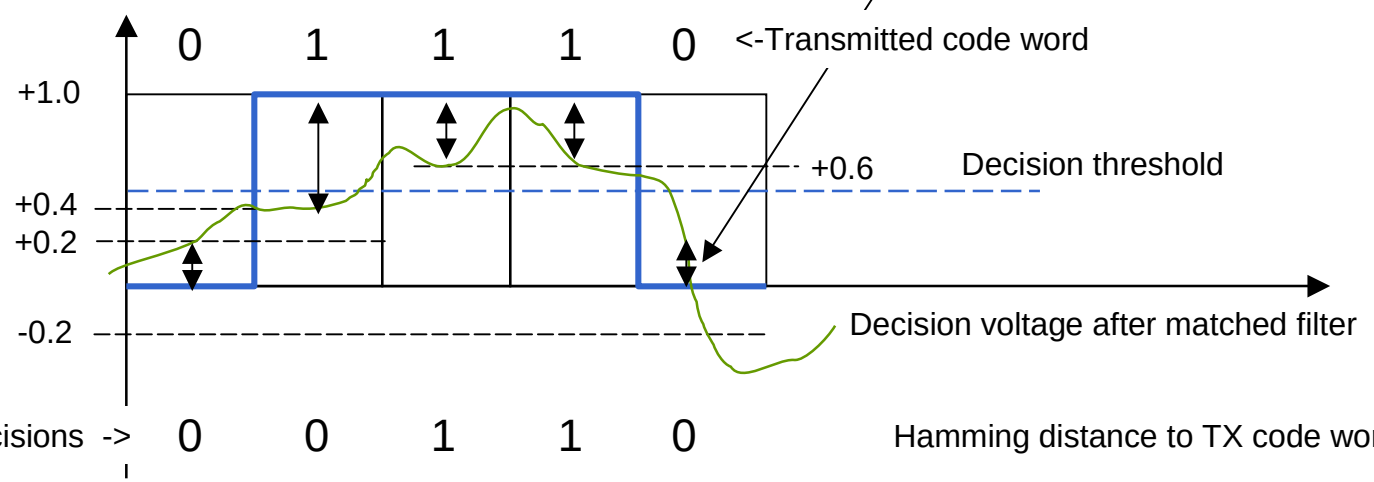
where *n* and *k* are the number of bits in the encoded word and in the word to be encoded respectively. Code rate is a measure of **code efficiency** and it is defined by

$$R_C = k/n \leq 1$$

- Note that $q=n-k$ bits are added for error detection/correction
- We noticed that repetition codes have a very low efficiency because their rate is only 1/n. On the contrary, parity check codes have much higher efficiency of (n-1)/n and and they have still a relatively good performance.

Hard and Soft Decision Decoding

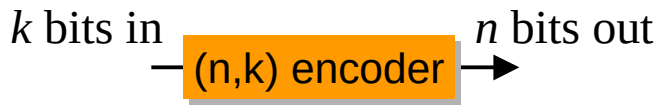
- Hard decision decoding
  - each received code word is compared to the applied codes and the code with the minimum **Hamming distance** is selected
- Soft decision decoding
  - decision reliability is estimated from demodulator's analog voltage after matched filter by **Euclidean distance**
- Finally, the code word with the smallest distance with respect of the all the possible code words is selected as the received code word



Hard and Soft Decoding (cont.)

- Hard decoding:
  - calculates Hamming distances to all allowed code words
  - selects the code with the smallest distance
- Soft decoding:
  - calculates Euclidean distance to all allowed code words
  - selects the code with the smallest distance
- Soft decoding yields about 2 dB improvement when compared to hard decisions (requires a high SNR)
- Often soft decoding realized by Viterbi decoder. In fading channels bit stream is interleaved to avoid effect of burst errors
- Computational complexity in decoding can be reduced by using convolutional codes

Block Codes



- In (*n*,*k*) block codes each sequence of *k* information bits is mapped into a sequence of *n (>k)* channel inputs in a fixed way regardless of the previous information bits (in contrast to convolutional codes)
- The formed code family should be formed such that the code minimum distance and code rate is large -> high error correction/detection capability
- A systematic block code: In the encoded word
  - the first *k* elements are the same as the message bits
  - *q=n-k* bits are the check bits
- Therefore a code vector of a block code is

$$\mathbf{X} = (m_1 \;\; m_2 \;\; \ldots m_k \;\; c_1 \;\; c_2 \;\; \ldots . \;\; c_q), \;\; q = n - k$$

or as a partitioned representation

$$\mathbf{X} = (\mathbf{M} \mid \mathbf{C})$$

Block Codes by Matrix Representation

- Given a message vector **M**, the respective linear, systematic block code **X** can be obtained by matrix multiplication by

$$\mathbf{X} = (\mathbf{M} \mid \mathbf{C}) = \mathbf{MG}$$

- **G** is the generator matrix with the general structure

$$\mathbf{G} = (\mathbf{I}_k \mid \mathbf{P})$$

where I_k is *kxk* identity matrix and **P** is a *kxq* binary submatrix

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1q} \\ p_{21} & p_{22} & \cdots & p_{2q} \\ \vdots & \vdots & & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{kq} \end{bmatrix}$$

Example: Hamming Codes

- Hamming codes are defined by

$$n = 2^q - 1, q = 2, 3, \ldots \quad k = n - q \quad d_{\min} = 3$$

- Take a **systematic** (*n,k*) Hamming code with q=*3* and $n=2^3-1=7$ and $k=n-q=7-3=4$. The generator matrix is

$$\mathbf{G} = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right] = \left[\underbrace{\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}}_{\mathbf{I}} \; \underbrace{\begin{array}{ccc} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}}_{\mathbf{P}}\right]$$

- In Hamming codes *kxq* submatrix **P** includes all the *q*-bit words that have *q*-1 or *q* of “1”:s

Check-bit Equations

- For *u* message vectors **M** (each consisting of *k* bits) the respective *n*-bit block codes **X** are determined by matrix multiplication **X=MG**

$$\mathbf{G} = (\mathbf{I}_k \mid \mathbf{P}) \begin{bmatrix} 1 & 0 & \cdots & 0 & p_{1,1} & p_{1,2} & \cdots & p_{1,q} \\ 0 & 1 & \cdots & 0 & p_{2,1} & p_{2,2} & \cdots & p_{2,q} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & p_{k,1} & p_{k,2} & \cdots & p_{k,q} \end{bmatrix}$$

$$\mathbf{X} = \mathbf{MG} = \underbrace{\begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,k} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,k} \\ \vdots & \vdots & & \vdots \\ m_{u,1} & m_{u,2} & \cdots & m_{u,k} \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,k} & c_{1,1} & c_{1,2} & \cdots & c_{1,q} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,k} & c_{2,1} & c_{2,2} & \cdots & c_{2,q} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ m_{u,1} & m_{u,2} & \cdots & m_{u,k} & c_{u,1} & c_{u,2} & \cdots & c_{u,q} \end{bmatrix}}_{\mathbf{X}=(\mathbf{M}\mid\mathbf{C})}$$

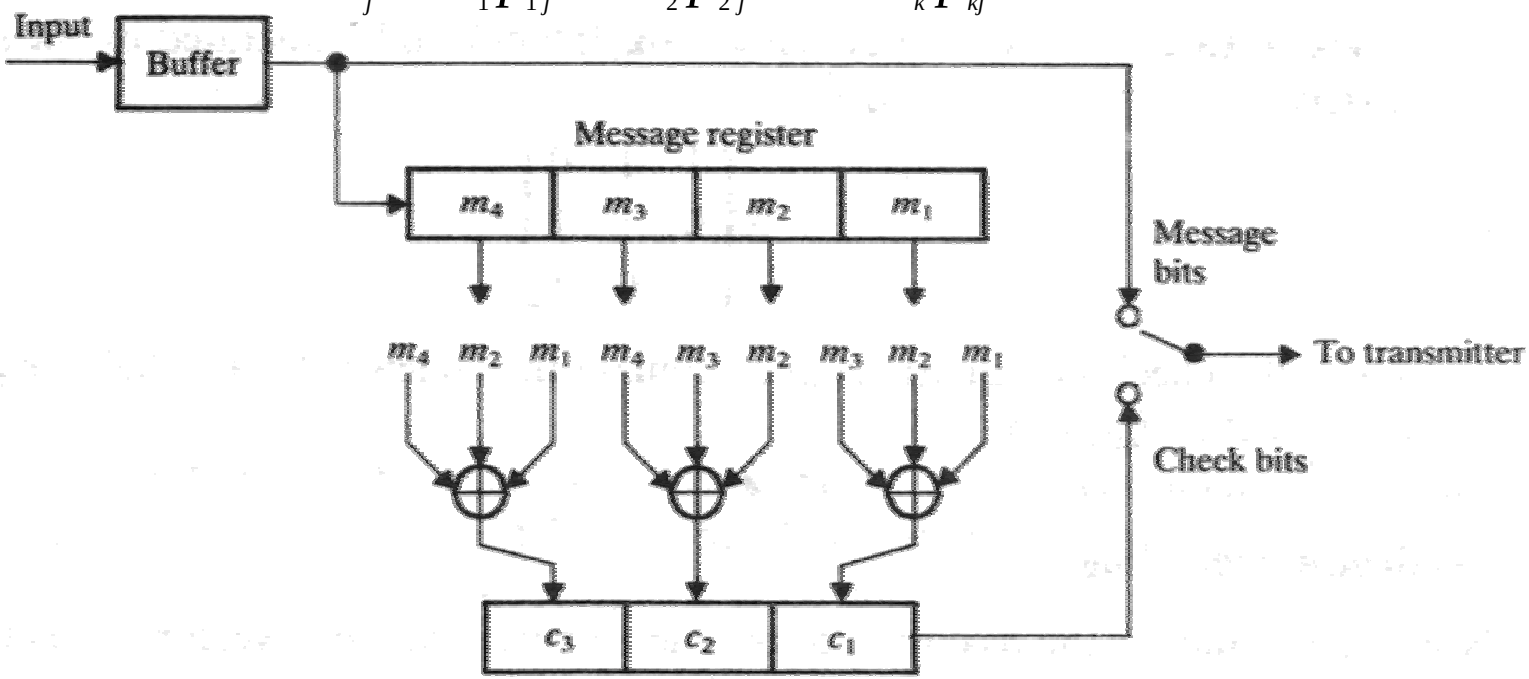
q=n-k check bits generated for each word, for instance

$$c_{1,q} = m_{1,1} p_{1,q} \oplus m_{1,2} p_{2,q} \oplus \cdots \oplus m_{1,k} p_{k,q}$$

(7,4) Hamming Encoder

- Therefore we have a four element message register implementation for the (7,4) Hamming encoder
- The check bits [c_1,c_2,c_3] are obtained by substituting the elements of **P and M** into equation **C**=**MP** *or*

$$c_j = m_1 p_{1j} \oplus m_2 p_{2j} .... \oplus m_k p_{kj}$$



Generated Hamming Codes

- Going through all the combinations of the input vector X, yields then all the possible output vectors

M	C	w(X)	M	C	w(X)
0 0 0 0	0 0 0	0	1 0 0 0	1 0 1	3
0 0 0 1	0 1 1	3	1 0 0 1	1 1 0	4
0 0 1 0	1 1 0	3	1 0 1 0	0 1 1	4
0 0 1 1	1 0 1	4	1 0 1 1	0 0 0	3
0 1 0 0	1 1 1	4	1 1 0 0	0 1 0	3
0 1 0 1	1 0 0	3	1 1 0 1	0 0 1	4
0 1 1 0	0 0 1	3	1 1 1 0	1 0 0	4
0 1 1 1	0 1 0	4	1 1 1 1	1 1 1	7

- Note that for linear codes the minimum distance of each code word is the weight *w* or the number of “1” on each code word (distance to ‘00...0’ code word)
- For Hamming codes the minimum distance is 3

Decoding Block Codes by Hard Decisions

- A brute-force method for error correction of a block code would include comparison to all possible same length code structures and choosing the one with the minimum Hamming distance when compared to the received code
- In practice applied codes can be very long and the extensive comparison would require much time and memory. For instance, to get 9/10 code rate with a Hamming code requires that

$$\frac{k}{n} = \frac{k}{2^q - 1} = \frac{n - q}{2^q - 1} \geq \frac{9}{10}$$

- This equation fulfills if the code length is at least *k=57*, that results *n = 10k/9=63.*
- There are $2^k = 1.4 \cdot 10^{17}$ different block codes in this case! -> Decoding by direct comparison would be quite unpractical!

Syndrome Decoding

- In syndrome decoding a parity checking matrix **H** is designed such that multiplication with a code word **X** produces all-zero matrix:

$$\mathbf{X}\mathbf{H}^T = (0\ 0 \cdots 0)$$

- Therefore error detection of the received signal **Y** can be based on $q=n-k$ bit syndrome:

$$\mathbf{S} = \mathbf{Y}\mathbf{H}^T$$

  that is always zero when a correct code word is received. (Note again that the syndrome does not reveal errors if channel noise has produced another code word!)

- The parity checking matrix is determined by

$$\mathbf{H} = (\mathbf{P}^T \mid \mathbf{I}_q) \quad \text{or} \quad \mathbf{H}^T = \begin{pmatrix} \mathbf{P} \\ \hline \mathbf{I}_q \end{pmatrix}$$

- Note that **P** matrix can be the same that was applied in the transmitter side

Syndrome Decoding (cont.)

- Syndrome decoding can be used for error correction by checking one-bit error patterns for each syndrome
- Example: Consider a (7,4) Hamming code with a parity check matrix

$$\mathbf{H} = (\mathbf{P}^T \mid \mathbf{I}_q) = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- The respective syndromes and error vectors (showing the position of errors by "1") are

S			Error vector $\hat{e}$						
0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0
1	1	0	0	0	1	0	0	0	0
0	1	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	1

$\mathbf{S} = \mathbf{Y}\mathbf{H}^T$ where $\mathbf{Y}$ is any valid code with the error in the position indicated by the respective syndrome

Syndrome is Independent of Code Words

- This design enables that the syndrome depends entirely on error pattern but not on particular code. Consider for instance

$$\mathbf{X} = (1\,0\,1\,1\,0)\ \mathbf{Y} = (1\,0\,0\,1\,1) \Rightarrow \quad \mathbf{E} = (0\,0\,1\,0\,1) \quad (\mathbf{Y} = \mathbf{X} + \mathbf{E})$$

$$\mathbf{S} = \mathbf{Y}\mathbf{H}^T \Rightarrow$$

$$(\mathbf{X} + \mathbf{E})\mathbf{H}^T = \underbrace{\mathbf{X}\mathbf{H}^T}_{\rightarrow 0,\ \text{that follows from the defintion of } \mathbf{H}} + \mathbf{E}\mathbf{H}^T = \mathbf{E}\mathbf{H}^T$$

- Syndrome does not determine the error pattern uniquely because there exists only 2^q different syndromes (syndrome length is q) but there exists 2^k different codes (for each symbol that must be encoded).
- After error correction decoding double errors can turn out even triple errors
- Therefore syndrome decoding is efficient for error correction when channel errors are not too likely, e.g. probability for double errors must be small
- NOTE: Syndrome decoding is anyhow very efficient for error detection also in difficult channels

Note also that $\mathbf{X} = \mathbf{Y} + \mathbf{E}$

Table Lookup Syndrome Decoder Circuit

- The error vector is used for error correction by the circuit shown bellow: