Sistemas de Telecomunicações

6ª Aula

Universidade do Minho
Escola de Engenharia

Outline

- Basic coding techniques:
  - Error detection and Error correction;
  - Block codes;
  - Hamming codes;
  - Cyclic codes;
  - Convolutional codes;

Error Detection and Error Correction

José Cabral
Departamento de Electrónica Industrial
Escola de Engenharia
Universidade do Minho

Definitions

- One of the predictions made in the Shannon channel coding theorem is that a rather sophisticated coding technique can convert a noisy channel (unreliable transmission) into an error-free channel (reliable transmission).
- Message words are arranged as blocks of k bits, which are randomly assigned codewords of n bits, $n > k$, in an assignment that is characterized by the addition of redundancy.

Definitions

- However, what is not completely defined in the theorem is a constructive method for designing such a sophisticated coding technique.
- There are basically two mechanisms for adding redundancy, in relation to error-control coding techniques:
  - Block coding
  - Convolutional coding.

Definitions

- For a given practical requirement, detection of errors is simpler than the correction of errors.

- The decision for applying detection or correction in a given code design depends on the characteristics of the application.

Definitions

- Codes can be designed for detecting errors, the correction is performed by requiring a repetition of the transmission. These schemes are known as automatic repeat reQuest (ARQ) schemes.
- When there is no possibility of requiring retransmission in the case of a detected error, and so the receiver has to implement some error-correction algorithm to properly decode the message. This transmission mode is known as forward error correction (FEC).

The Repetition Code

- One of the simplest ways of performing coding is to repeat a transmitted symbol *n* times.
- If this transmission uses a binary alphabet then the bit '1' is usually represented by a sequence of *n* '1's, while the bit '0' is represented by a sequence of *n* '0's.
  - 1 -> 11..1
  - 0 -> 00..0

The Repetition Code

- If errors happen randomly and with an error probability $P_e = p$ [as happens in the case of the binary symmetric channel (BSC)], the binomial distribution describes the probability of having i errors in a word of n bits:

$$p(i,n) = \binom{n}{i} p^i (1-p)^{n-i} \cong \binom{n}{i} p^i \ , \ p \ll 1 \qquad \textbf{(1)}$$

$$\binom{n}{i} = \frac{n!}{(n-i)!i!} \qquad \textbf{(2)}$$

The Repetition Code

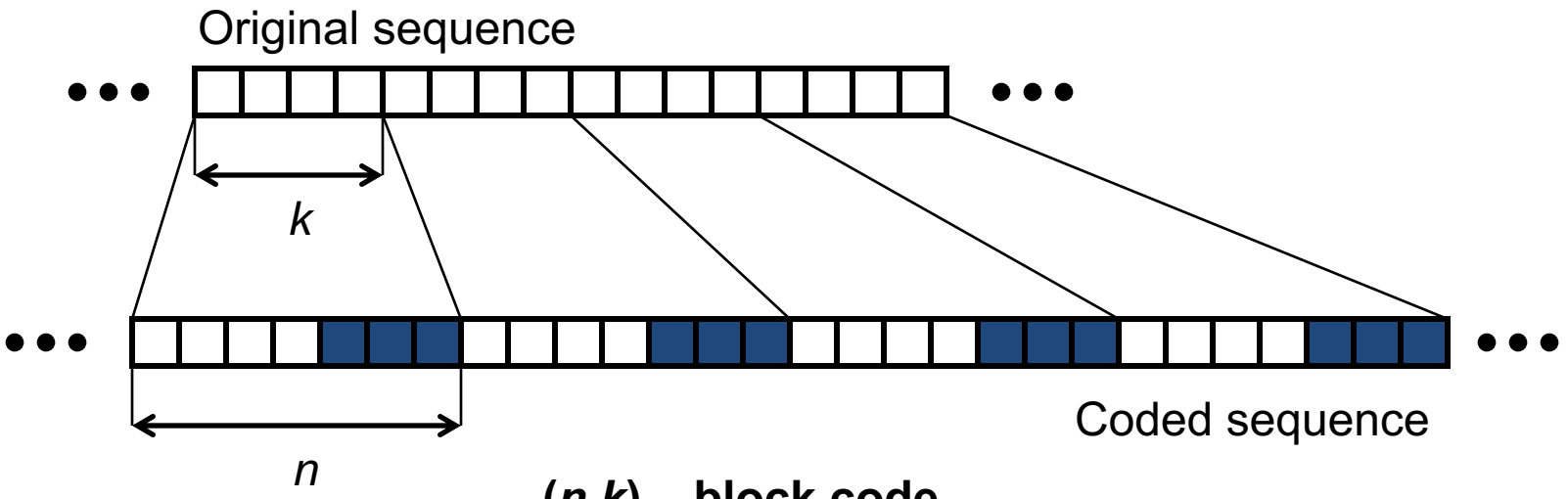
- Usually the value of p is small enough to validate the approximation made in equation (1). On the other hand, and for the same reason, it will be also true that if $p << 1$, then the probability of having i errors is higher than that of having $i + 1$ errors; that is:

$$P(i + 1, n) << P(i, n) \text{ ----→ Why?}$$

Block Codes

José Cabral
Departamento de Electrónica Industrial
Escola de Engenharia
Universidade do Minho

Block codes



Information bits

Parity check bits

(*n*,*k*) – block code

k: information bits (message) - 2^k possible messages

n: coded bits or the bits of the codeword – 2^n possible codewords

n – *k*: redundant bits or parity check bits.

Code rate: $R_c = k / n$

Block Codes - Introduction

- Error-control coding requires the use of a mechanism for adding redundancy to the message word.
- This redundancy addition (encoding operation) can be performed in different ways, but always in a way that by applying the inverse operation (decoding) at the decoder the message information can be successfully recovered.
- The final step in the decoding process involves the application of the decoding procedure, and then the discarding of the redundancy bits, since they do not contain any message information.

Block Codes – Vector Spaces

- Since the 2^k messages are converted into codewords of n bits, this encoding procedure can be understood as an expansion of the message vector space of size 2^k to a coded vector space of larger size 2^n, from which a set of 2^k codewords is conveniently selected.
- Block codes can be properly analysed by using vector space theory.
- A vector space is essentially a set of vectors ruled by certain conditions, which are verified by performing operations over these vectors, operations that are usually defined over a given field F.

Block Codes – Galois Fields

- A very useful vector space for the description of block codes is the vector space defined over the binary field, or Galois field GF(2). Galois fields GF(q) are defined for all the prime numbers q and their powers.
- The binary field GF(2) is a particular case of a Galois field for which $q = 2$. Consider an ordered sequence of n components $(a_0, a_1, \ldots, a_{n-1})$ where each component a_i is an element of the field GF(2), that is, an element adopting one of the two possible values 0 or 1.
- This sequence will be called an n-component vector. There will be a total of 2^n vectors. The corresponding vector space for this set of vectors will be denoted as V_n.

Block Codes – *Modulo-2* addition

- The binary addition operation $\oplus$ is defined for this vector space as follows:
  - if $u = (u_1, u_2, \ldots, u_{n-1})$ and $v = (v_1, v_2, \ldots, v_{n-1})$ are vectors in V_n, then: $u \oplus v = (u_1 \oplus v_1, u_2 \oplus v_2, \ldots, u_n \oplus v_n)$
  - Where $\oplus$ is the classic modulo-2 addition.
- Since the sum vector is also an *n*-component vector, this vector also belongs to the vector space V_n, and so the vector space is said to be closed under the addition operation $\oplus$.
- The addition of any two vectors of a given vector space is also another vector of the same vector space.

Modulo-2 addition and multiplication.

Modulo-2 addition

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$

Modulo-2 multiplication

$0 \bullet 0 = 0$
$0 \bullet 1 = 0$
$1 \bullet 0 = 0$
$1 \bullet 1 = 1$

- V_n is a commutative group under the addition operation.
- The all-zero vector **0** = (0, 0, …, 0) is also in the vector space
- $u \oplus 0 = (u_1 \oplus 0, u_2 \oplus 0, \ldots, u_n \oplus 0) = u$
- $u \oplus u = (u_1 \oplus u_1, u_2 \oplus u_2, \ldots, u_n \oplus u_n) = 0$
- $a \bullet u = (a \bullet u_1, a \bullet u_2, \ldots, a \bullet u_{n-1})$ - product between a vector of the vectorial space $u \in V$ and a scalar of the binary field $a \in GF(2)$
- It can be shown that the addition and scalar multiplication fit the associative, commutative and distributive laws, so that the set of vectors V_n is a vector space defined over the binary field *GF(2).*

Vector Subspaces

- For a given set of vectors forming a vector space V defined over a field F, it is possible to find a subset of vectors inside the vector space V, which can obey all the conditions for also being a vector space.
- This subset S is called a subspace of the vector space V. This non-empty subset S of the vector space V is a subspace if the following conditions are obeyed:
  - For any two vectors in S, u, $v \in S$, the sum vector $(u + v) \in$ S.
  - For any element of the field a $\in F$ and any vector $u \in S$, the scalar multiplication a $\bullet$ $u \in S$.

Vector Subspaces

- Example: The following subset is a subspace of the vector space *V4*: S = {(0000), (1001), (0100), (1101)}
- On the other hand, if $\{v_1, v_2, \ldots, v_k\}$ is a set of vectors of the vector space *V* defined over *F* and $a_1, a_2, \ldots, a_k$ are scalar numbers of the field *F*, the sum:
  - $a_1 \bullet v_1 \oplus a_2 \bullet v_2 \oplus \cdots \oplus a_k \bullet v_k$
- is called a linear combination of the vectors $\{v_1, v_2, \ldots, v_k\}$.
- Addition of linear combinations and multiplication of a linear combination by an element of the field *F* are also linear combinations of the vectors $\{v_1, v_2, \ldots, v_k\}$.

Vector Subspaces

- Theorem: If $\{v_1, v_2, \ldots, v_k\}$ are k vectors in V defined over F, the set of all the linear combinations of $\{v_1, v_2, \ldots, v_k\}$ is a subspace S of V.

- Example: By considering two vectors (1001) and (0100) of the vector space *V4*, their linear combinations form the same subspace *S* as shown in the above example:
  - $0 \bullet (1001) \oplus 0 \bullet (0100) = (0000)$
  - $0 \bullet (1001) \oplus 1 \bullet (0100) = (0100)$
  - $1 \bullet (1001) \oplus 0 \bullet (0100) = (1001)$
  - $1 \bullet (1001) \oplus 1 \bullet (0100) = (1101)$

Vector Subspaces

- A set of k vectors $\{v_1, v_2, \ldots, v_k\}$ is said to be linearly dependent if and only if there exist k scalars of the field *F*, not all equal to zero, such that a linear combination is equal to the all-zero vector:
  - $a_1 \bullet v_1 \oplus a_2 \bullet v_2 \oplus \cdots \oplus a_k \bullet v_k = 0$
- If the set of vectors is not linearly dependent, then this set is said to be linearly independent.
- Example: Vectors (1001), (0100) and (1101) are linearly dependent because
  - $1 \bullet (1001) \oplus 1 \bullet (0100) \oplus 1 \bullet (1101) = (0000)$
- A set of vectors is said to generate a vector space *V* if each vector in that vector space is a linear combination of the vectors of the set.

Dual Subspaces

- If S is a k-dimensional subspace of the n-dimensional vector space V_n, the set S_d of vectors v for which for any $u \in S$ and $v \in S_d$, $u \circ v = 0$ is called the dual subspace of S.
- It is possible to demonstrate that this set is also a subspace of V_n.
- Moreover, it can also be demonstrated that if the subspace S is of dimension k, the dual subspace S_d is of dimension (n − k). In other words:
  - $\dim(S) + \dim(S_d) = n$

Dual Subspaces

- Example: For the vector space V_4 over GF(2), the following set of vectors
  - S = {(0000), (0011), (0110), (0100), (0101), (0111), (0010), (0001)}
- is a three-dimensional subspace of V_4 for which the one-dimensional subspace
  - S_d = {(0000), (1000)} is the dual subspace
- S_d of S.

Linear Block Codes

- Definition: A block code of length n and 2^k message words is said to be a linear block code $C_b(n, k)$ if the 2^k codewords form a vector subspace, of dimension k, of the vector space V_n of all the vectors of length n with components in the field GF(2)
- Encoding basically means to take the 2^k binary message words of k bits each, and assign to them some of the 2^n vectors of n bits.
- Since usually $k < n$, there are more vectors of n bits than those of k bits, and so the selection of the vectors of n bits has to be done using the lowest level of redundancy while maximizing the distance among the codewords.
- The set of 2^k codewords constitute a vector subspace of the set of words of n bits. As a consequence of its definition, a linear block code is characterized by the fact that the sum of any of two codewords is also a codeword.

Generator Matrix *G*

- Since a linear block code $C_b(n, k)$ is a vector subspace of the vector space V_n, there will be k linearly independent vectors that in turn are codewords $g_0, g_1, \ldots, g_{k-1}$, such that each possible codeword is a linear combination of them:
  - $c = m_0 \bullet g_0 \oplus m_1 \bullet g_1 \oplus \cdots \oplus m_{k-1} \bullet g_{k-1}$
- These linearly independent vectors can be arranged in a matrix called the generator matrix *G*:

$$\underset{[k \times n]}{\boldsymbol{G}} = \begin{bmatrix} \boldsymbol{g_0} \\ \boldsymbol{g_1} \\ \vdots \\ \boldsymbol{g_{k-1}} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

Generator Matrix *G*

- This is a matrix mechanism for generating any codeword. For a given message vector:
  - m = (m_0, m_1, …, m_{k-1}), the corresponding codeword is obtained by matrix multiplication:

$$\boldsymbol{c} = \boldsymbol{m} \circ \boldsymbol{G} = (m_0, m_1, \ldots, m_{k-1}) \circ \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

$$= (m_0, m_1, \ldots, m_{k-1}) \circ \begin{bmatrix} \boldsymbol{g_0} \\ \boldsymbol{g_1} \\ \vdots \\ \boldsymbol{g_{k-1}} \end{bmatrix} = m_0 \bullet \boldsymbol{g_0} \oplus m_1 \bullet \boldsymbol{g_1} \oplus \cdots \oplus m_{k-1} \bullet \boldsymbol{g_{k-1}}$$

  - '$\circ$' represents the inner product between vectors or matrices
  - '$\bullet$' represents the multiplication by a scalar in the field GF(2) of a vector of the vector space or subspace used

Generator Matrix *G*

- The rows of the generator matrix *G* generate the linear block code $C_b(n, k)$, or, equivalently, the *k* linearly independent rows of *G* completely define the code.
  - Example: Codewords of a linear block code $C_b(7, 4)$

Message	Codewords
0 0 0 0	0 0 0 0 0 0 0
0 0 0 1	1 0 1 0 0 0 1
0 0 1 0	1 1 1 0 0 1 0
0 0 1 1	0 1 0 0 0 1 1
0 1 0 0	0 1 1 0 1 0 0
0 1 0 1	1 1 0 0 1 0 1
0 1 1 0	1 0 0 0 1 1 0
0 1 1 1	0 0 1 0 1 1 1
1 0 0 0	1 1 0 1 0 0 0
1 0 0 1	0 1 1 1 0 0 1
1 0 1 0	0 0 1 1 0 1 0
1 0 1 1	1 0 0 1 0 1 1
1 1 0 0	1 0 1 1 1 0 0
1 1 0 1	0 0 0 1 1 0 1
1 1 1 0	0 1 0 1 1 1 0
1 1 1 1	1 1 1 1 1 1 1

Generator Matrix *G*

- Example: Consider the following generator matrix (4 × 7) and obtain the codeword corresponding to the vector message *m* = (1001).

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- The corresponding codeword is:

$c = m \circ G = 1 \bullet g_0 \oplus 0 \bullet g_1 \oplus 0 \bullet g_2 \oplus 1 \bullet g_3$

$= (1101000) \oplus (1010001) = (0111001)$

Block Codes in Systematic Form

- It can be seen that the last four bits of each codeword are the same as the message bits; that is, the message appears as it is, inside the codeword.
- In this case, the first three bits are the so-called parity check or redundancy bits.
- This particular form of the codeword is called systematic form. In this form, the codewords consist of the ($n - k$) parity check bits followed by the k bits of the message. The structure of a codeword in systematic form is shown above

- but this can be done the other way round. However, the choice of convention does not modify the properties of a given block code, although some mathematic expressions related to the code will of course adopt a different form in each case.

Block Codes in Systematic Form

- Generator matrix of the systematic linear block code $C_b(n, k)$:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{g_0} \\ \boldsymbol{g_1} \\ \vdots \\ \boldsymbol{g_{k-1}} \end{bmatrix} = \underbrace{\begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} \\ \vdots & \vdots & & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}}_{\text{submatrix } \boldsymbol{P} \text{ k} \times (\text{n} - \text{k})} \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{\text{submatrix } \boldsymbol{I} \text{ k} \times \text{k}}$$

- which, in a compact notation, is $G = [P \quad I_k]$
- Example: linear block code $C_b(7, 4)$

$$\boldsymbol{c} = \boldsymbol{m} \circ \boldsymbol{G} = (m_0, m_1, m_2, m_3) \circ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} c_0 &= m_0 \oplus m_2 \oplus m_3 \\ c_1 &= m_0 \oplus m_1 \oplus m_2 \\ c_2 &= m_1 \oplus m_2 \oplus m_3 \\ c_3 &= m_0 \\ c_4 &= m_1 \\ c_5 &= m_2 \\ c_6 &= m_3 \end{aligned}$$

Parity Check Matrix H

- The 2^{n-k} linear combinations of the matrix H generate the dual code $C_{bd}(n, n-k)$, which is the dual subspace of the code C_b generated by the matrix G.
- The systematic form of the parity check matrix H of the code C_b generated by the generator matrix G is:

$$\boldsymbol{H} = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}}_{\text{submatrix } \boldsymbol{I}\ (\mathrm{n}-\mathrm{k}) \times (\mathrm{n}-\mathrm{k})} \underbrace{\begin{bmatrix} p_{00} & p_{10} & \cdots & p_{k-1,0} \\ p_{01} & p_{11} & \cdots & p_{k-1,1} \\ \vdots & \vdots & & \vdots \\ p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}}_{\text{submatrix } \boldsymbol{P^T}\ (\mathrm{n}-\mathrm{k}) \times \mathrm{k}} = \begin{bmatrix} \boldsymbol{I}_{n-k} & \boldsymbol{P}^T \end{bmatrix}$$

- where P^T is the transpose of the parity check submatrix P.
- The matrix H is constructed so that: $\boldsymbol{G \circ H^T = 0}$

Parity Check Matrix H

- Example: Determine the parity check matrix *H* for the linear block code $C_b(7, 4)$ generated by the generator matrix

$$G = \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}}_{\text{submatrix } P} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{submatrix } I}$$

- the parity check matrix *H* is constructed using the submatrices *P* and *I*:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

- A practical implementation of these codes could be done using combinational logic for the parity check equations

Errors in a noisy channel

- The codeword $c = (c_0, c_1, \ldots, c_{n-1})$ is such that its components are taken from the binary field GF(2), $c_i \in$ GF(2).
- As a consequence of its transmission through a noisy channel, this codeword could be received containing some possible errors.
- The received vector can therefore be different from the corresponding transmitted codeword, and it will be denoted as:
  - $r = (r_0, r_1, \ldots, r_{n-1})$, where it is also true that $r_i \in$ GF(2).

Error pattern

- An error event can be modelled as an error vector or error pattern
  - $e = (e_0, e_1, \ldots, e_{n-1}), \quad e_i \in \mathrm{GF}(2),$
- which is related to the codeword and received vectors as follows:
  - $e = r \oplus c$
- it is possible to do a correction of the received vector in order to determine an estimate of the valid codeword, and this can be done by using the expression
  - $c = r \oplus e$

Syndrome vector

- Since any codeword should obey the condition:
  - $c \circ H^T = 0$
- an error-detection mechanism can be implemented based on the above expression, which adopts the following form:
  - $S = r \circ H^T = (s_0, s_1, \ldots, s_{n-k-1})$ -> ***Syndrome vector***
  - The dimension of the syndrome vector is $1 \times (n - k)$.
- The detection operation is performed over the received vector, so that if this operation results in the all-zero vector, then the received vector is considered to be a valid codeword

Syndrome vector

- Since:
  - $r = c \oplus e$
- In case of the decoder has detected errors.
  - $S = r \circ H^T = (c \oplus e) \circ H^T = c \circ H^T \oplus e \circ H^T = e \circ H^T$
- If the error pattern is the all-zero vector, then the syndrome vector will also be an all-zero vector, and thus the received vector is a valid codeword.
- When the syndrome vector contains at least one non-zero component, it will be detecting the presence of errors in the received vector.

Syndrome error detection

- If the error pattern is equal to a codeword; that is, if the number and positions of the errors are such that the transmitted codeword is converted into another codeword, then the syndrome vector can be the all-zero vector!!!
- This error pattern will not be detected by the syndrome operation. This is what is called an undetected error pattern, and as such is not within the error-correction capability of the code.
- Undetected error patterns are characterized by satisfying the condition $S = e \circ H^T = 0$; that is, these are the error patterns that are equal to one of the codewords ($e = c$). There will be therefore $2^k - 1$ undetectable non-zero error patterns.

Syndrome error detection

- Example: For the same linear block code $C_b(7, 4)$, as seen in previous examples, obtain the analytical expressions for the syndrome vector's bits.
  - If $r = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$

$$\boldsymbol{S} = (s_0, s_1, s_2) = (r_0, r_1, r_2, r_3, r_4, r_5, r_6) \circ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$s_0 = r_0 \oplus r_3 \oplus r_5 \oplus r_6$$

$$s_1 = r_1 \oplus r_3 \oplus r_4 \oplus r_5$$

$$s_2 = r_2 \oplus r_4 \oplus r_5 \oplus r_6$$

Syndrome error detection

- The syndrome vector does not depend on the received vector, but on the error vector.
- These will allow us to evaluate the error vector, which in turn will allow us to do an estimation of a valid codeword.
- However, this set of $(n - k)$ equations does not have a unique solution, but exhibits 2^k solutions.
- This is due to the fact that there are 2^k error patterns that produce the same syndrome vector.
- Because the noise power normally acts with minimum effect, the error pattern with the least number of errors will be considered to be the true solution of this system of equations

Syndrome error detection

- Example: For the linear block code $C_b(7,4)$ of the previous examples, a transmitted codeword
  - c = (0011010)
- is affected by the channel noise and received as:
  - r = (0001010).
- The calculation of the syndrome vector results:
  - S = (001)
- which in terms of the system of equations becomes:
  - $0 = e_0 \oplus e_3 \oplus e_5 \oplus e_6$
  - $0 = e_1 \oplus e_3 \oplus e_4 \oplus e_5$
  - $1 = e_2 \oplus e_4 \oplus e_5 \oplus e_6$

Syndrome error detection

- There are $2^4 = 16$ different error patterns that satisfy the above equations:

e_0	e_1	e_2	e_3	e_4	e_5	e_6
0	0	1	0	0	0	0
1	1	1	1	0	0	0
1	0	0	0	0	0	1
0	1	0	1	0	0	1
0	0	0	1	0	1	0
1	1	0	0	0	1	0
0	1	1	0	0	1	1
1	0	1	1	0	1	1
0	1	0	0	1	0	0
1	0	0	1	1	0	0
1	1	1	0	1	0	1
0	0	1	1	1	0	1
1	0	1	0	1	1	0
0	1	1	1	1	1	0
1	1	0	1	1	1	1
0	0	0	0	1	1	1

- Since the error pattern with *i* errors is more likely than the error pattern of *i + 1* errors, the smallest number of non-zero components will be considered as the true error pattern
  - $c = r \oplus e$ = (0011010) = (0001010) $\oplus$ (0010000)

Minimum Distance of a BC

- Definition: The number of non-zero components $c_i \neq 0$ of a given vector $c = (c_0, c_1, \ldots, c_{n-1})$ of size $(1 \times n)$ is called the weight, or *Hamming weight*, $w(c)$, of that vector. In the case of a vector defined over the binary field *GF(2)*, the weight is the number of '1's in the vector.
- Definition: The Hamming distance between any two vectors $c_1 = (c_{01}, c_{11}, \ldots, c_{n-1,1})$ and $c_2 = (c_{02}, c_{12}, \ldots, c_{n-1,2})$, $d(c_1, c_2)$, is the number of component positions in which the two vectors differ.
  - For instance, if $c_1 = (0011010)$ and $c_2 = (1011100)$, then $d(c_1, c_2) = 3$
- According to the above definitions, it can be verified that:
  - $d(c_i , c_j) = w(c_i \oplus c_j)$
- This minimum value of the distance, evaluated over all the codewords, of the code is called the minimum distance of the code:
  - $d_{min} = min \{ d(c_i , c_j); c_i , c_j \in C_b; c_i \neq c_j \}$

Minimum Distance of a BC

- Since, in general, block codes are designed to be linear, the addition of any two code vectors is another code vector,
- Any codeword can be seen as the addition of at least 2 other codewords,
- Hamming distance is the number of positions in which 2 vectors differ,
- The weight of the sum of 2 vectors is the Hamming distance between these 2 vectors
- $\Rightarrow$ then the weight of a codeword is at the same time the distance between 2 other vectors of that code,
- $\Rightarrow$ then the minimum value of the weight evaluated over all the codewords of a code, excepting the all-zero vector, is the minimum distance of the code
- Therefore, the minimum distance of a linear block code $C_b(n, k)$ is the minimum value of the weight of the non-zero codewords of that code – *see previous examples*

Minimum Distance and *H* matrix

- Theorem:
- Consider a linear block code $C_b(n, k)$ completely determined by its parity check matrix *H*.
- For each codeword of Hamming weight p_H, there exist p_H columns of the parity check matrix *H* that when added together result in the all-zero vector.
- In the same way, it can be said that if the parity check matrix *H* contains p_H columns that when added give the all-zero vector, then there is in the code a vector of weight p_H
- For a linear block code $C_b(n, k)$ completely determined by its parity check matrix *H*, the minimum weight or minimum distance of this code is equal to the minimum number of columns of that matrix which when added together result in the all-zero vector 0.

Minimum Distance and *H* matrix

- Example:
- For the linear block code $C_b(7, 4)$, as seen in previous examples, whose parity check matrix is of the form:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

- *Determine the minimum distance of this code*
- Answer:
- It can be seen that the addition of the first, third and seventh column results in the all-zero vector 0. Hence, and because the same result cannot be obtained by the addition of only two columns, the minimum distance of this code is $d_{min} = 3$.

Error-Detection Capability of a BC

- The *minimum distance* of a code is the minimum number of components changed by the effect of the noise that converts a code vector into another vector of the same code.
- If having transmitted the codeword c the noise transforms this vector in the received vector r, the distance between c and r is the weight of the error pattern $d(c, r) = w(e) = l$, that is, the number of positions that change their value in the original vector c due to the effects of noise.
- If the noise modifies d_{min} positions, then it is possible in the worst case that a code vector is transformed into another vector of the same code, so that the error event is undetectable.
- If the number of positions the noise alters is $d_{min}-1$, it is guaranteed that the codeword cannot be converted into another codeword
- $\Rightarrow$ Error-Detection Capability of a linear block code $C_b(n, k)$ of minimum distance d_{min} is $d_{min} - 1$

Probability of an Undetected Error

- The error-detection capability of a code can be measured by means of the probability that the code fails to determine an estimate of the codeword from the received vector
- The Probability of an undetected error is evaluated using the weight distribution function of the code.
- Since a detection failure happens when the error pattern is equal to a non-zero codeword:

$$p_U(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i}$$

- where A_i is the number of codewords of weight i and p is the error probability for the BSC, for which this analysis is valid.
- When the minimum distance is d_{min}, the values of A_1 to $A_{d_{min}-1}$ are all zero.

Error-Correction Capability of a BC

- The same considerations are used to prove that:

$$t = \left[\frac{d_{\min} - 1}{2} \right]$$

- The code is able to successfully decode any error pattern of weight t , where [_] means the largest integer number no greater than _
- The word or code vector error probability is:

$$p_{we} = \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

Error Correction / Detection

- Summary:

$$d_{\min} \geq l + 1$$ only detection

$$d_{\min} \geq 2t + 1$$ only correction

$$d_{\min} \geq t + l + 1 \quad (l > t)$$ detection and correction

Standard Array

- The standard array is useful to evaluate the error detection and error correction capacity of a block code $C_b(n, k)$.
- The array is constructed in the following way:
  - A row containing the codewords, including and starting from the all-zero vector (0, 0, …, 0), is constructed. This row contains 2^k vectors taken from the whole set of 2^n possible vectors:
    - $c_1 = (0, 0, \ldots, 0)$ c_2 c_3 $\ldots$ C_2^k
  - Then an error pattern e_2 is selected and placed below c_1 and then the sum vector $c_i \oplus e_2$ is placed below c_i
  - This is done with all the error patterns taken from the vector space that have to be allocated, a total of 2^{n-k} vectors.

Standard Array

- In this array the sum of any 2 vectors in the same row is a code vector. There are only $2^n/2^k$ disjoint rows in this array. These rows are the so-called *cosets* of the linear block code $C_b(n, k)$.
- The vector that starts each coset is called the leader of that coset, and it can be any vector of that row.
- Example: For the linear block code $C_b(6, 3)$ with the code words given below, determine the standard array.

- Codewords: 000000 110100 011010 101110 101001
  011101 110011 000111

Standard Array

- In order to minimize the error probability, all the correctable error patterns, which are the coset leaders, will have to be the most likely patterns.
- In the case of a transmission over the BSC, the most likely patterns are those of the lowest possible weight. Thus, each coset leader will be of the lowest possible weight among the vectors of that row.
- The decoding in this case will be *maximum likelihood decoding*, that is, minimum distance decoding, so that the decoded code vector is at minimum distance with respect to the received vector.

Standard Array

- The standard array of the previous example is:

000000	110100	011010	101110	101001	011101	110011	000111
000001	110101	011011	101111	101000	011100	110010	000110
000010	110110	011000	101100	101011	011111	110001	000101
000100	110000	011110	101010	101101	011001	110111	000011
001000	111100	010010	100110	100001	010101	111011	001111
010000	100100	001010	111110	111001	001101	100011	010111
100000	010100	111010	001110	001001	111101	010011	100111
010001	100101	001011	111111	111000	001100	100010	010110

- A linear block code $C_b(n, k)$ is able to detect $2^n - 2^k$ error patterns and correct 2^{n-k} error patterns.
- A $C_b(n, k)$ with d_{min}, all the vectors of weight: $t = [(d_{min}-1)/2]$ or less can be used as coset leaders.
- This is in agreement with the fact that not all the weight $t + 1$ error patterns can be corrected, even when some of them can be.

Standard Array and Syndrome decoding

- The vectors of the same coset have the same syndrome, whereas syndromes for different cosets are different.
- By taking a coset leader as the vector e_i , any other vector of that coset is the sum of the leader vector and the code vector c_i For this case, the syndrome is calculated as:
  - $(c_i \oplus e_i) \circ H^T = c_i \circ H^T \oplus e_i \circ H^T = e_i \circ H^T$
- For each correctable error pattern, there is a different syndrome vector.
- This allows us to implement simpler decoding by constructing a table where correctable error patterns and their corresponding syndrome vectors are arranged
- When the decoder makes the syndrome calculation and knows the syndrome vector, it can recognize the corresponding error pattern.

Hamming Codes

José Cabral
Departamento de Electrónica Industrial
Escola de Engenharia
Universidade do Minho

Hamming codes

- In 1950 R. Hamming introduced a code family:
  - Length n = $2^m - 1$
  - Number of message bits $k = 2^m - m - 1$
  - Number of parity check bits $n - k = m$
  - Error-correction capability $t = 1$, ($d_{min} = 3$)
  - For any positive integer $m \geq 3$
- $H = [I_m \;\; Q]$:
  - I_m is a square matrix of size $m \times m$ and the sub-matrix Q consists of the $2^m - m - 1$ columns formed with vectors of weight 2 or more.
- Examples of Hamming codes: (7,4) , (15,11), (31,26) , ($2^{n-k} - 1, k$)

Hamming codes

- The generator matrix can be constructed using the following expression for linear block codes of systematic form:
  - $G = [Q^T \quad I_{2^m-m-1}]$
- In H, the sum of 3 columns can result in the all-zero vector, and it is not possible for the sum of 2 columns to give the same result, and so the minimum distance of the code is $d_{min} = 3$.
- This means that $t = 1$ and $l = 2$

Hamming codes

- There are $2^m - 1$ correctable error patterns and there exist 2^m cosets, so that the number of possible correctable error patterns is the same as the number of different cosets (syndrome vectors).

- The codes with this characteristic are called perfect codes.

Cyclic Codes

José Cabral
Departamento de Electrónica Industrial
Escola de Engenharia
Universidade do Minho

Definitions

- Cyclic codes are an important class of linear block codes
- Easily implemented using *shift registers*.
- A given linear block code is said to be cyclic if for each of its code vectors the *i*th cyclic rotation is also a code vector of the same code
- Being a linear block code, the sum of any two code vectors of a cyclic code is also a code vector.

Definitions

- Cyclic codes can be represented by polynomials.
- A polynomial representation $c(X)$ of a code vector $c = (c_0, c_1, \ldots, c_{n-1})$ is then of the form:
  - $c(X) = c_0 + c_1 X + \cdots + c_{n-1} X^{n-1}$
- Operations with polynomials defined over a given field are the same as usual.
- In the case of polynomials defined over the binary field $GF(2)$, the operations are addition and multiplication modulo 2.

Definitions

- Addition:
  - $c_1(X) = c_{01} + c_{11}X + \cdots + c_{n-1,1}X^{n-1}$
  - $c_2(X) = c_{02} + c_{12}X + \cdots + c_{n-1,2}X^{n-1}$
  - $c_1(X) \oplus c_2(X) = c_{01} \oplus c_{02} + (c_{11} \oplus c_{12})\, X + \cdots + (c_{n-1,1} \oplus c_{n-1,2})\, X^{n-1}$
- Multiplication
  - $c_1(X) \bullet c_2(X) = c_{01} \bullet c_{02} + (c_{01} \bullet c_{12} \oplus c_{02} \bullet c_{11})\, X + \cdots + (c_{n-1,1} \bullet c_{n-1,2})\, X^{2(n-1)}$

Definitions

- Addition and multiplication of polynomials obey the commutative, associative and distributive laws.
- A polynomial for a code vector of n components is a polynomial of degree $n - 1$ or less.
- Codewords of a given cyclic code $C_{cyc}(n, k)$ will be equivalently referred to as code vectors or code polynomials.

Definitions

- The polynomial expression for the *i*-position right-shift rotated polynomial $c^{(i)}(X)$ of the original code polynomial $c(X)$ is equal to:
  - $c^{(i)}(X) = X^i\, c(X) \bmod (X^n + 1)$
  - mod is the modulo operation, calculated by taking the remainder of the division of $X^i\, c(X)$ and $X^n + 1$.
  - The non-zero minimum-degree code polynomial of a given cyclic code $C_{cyc}(n, k)$ is unique
  - In the non-zero minimum-degree polynomial of a given $C_{cyc}(n, k)$, $g_0 = 1$.

Generator Polynomial of a Cyclic Code

- In a linear $C_{cyc}(n,k)$, there is a unique non-zero minimum-degree code polynomial, and any other code polynomial is a multiple of this polynomial.
  - $g(X) = 1 + g_1X + \cdots + g_{n-k-1}X^{n-k-1} + X^{n-k}$

$g(X)$ Generator Polynomial.

- The non-zero minimum-degree polynomial is of degree $n - k$ and any other code polynomial of the linear $C_{cyc}(n,k)$ is of degree $n - 1$ or less
  - $c(X) = m(X).g(X) = (m_0 + m_1X + \cdots + m_{k-1}X^{k-1}).\ g(X)$
  - where m_i , $i = 0, 1, 2, \ldots, k - 1$, are the bits of the message vector to be encoded

Generator Polynomial of a Cyclic Code

- There are two important relationships between the generator polynomial $g(X)$ and the polynomial $X^n + 1$:
  - if $g(X)$ is a generator polynomial of a given linear $C_{cyc}(n, k)$, then $g(X)$ is a factor of $X^n + 1$,
  - if a polynomial of degree $n-k$ is a factor of $X^n + 1$, then this polynomial generates a linear $C_{cyc}(n, k)$.
- Any polynomial factor of $X^n + 1$ can generate a linear $C_{cyc}(n, k)$.

Cyclic Codes in Systematic Form

- Given a polynomial that fits the conditions for being the generator polynomial of a linear $C_{cyc}(n, k)$, and if the message polynomial is of the form:
  - $m(X) = m_0 + m_1X + \cdots + m_{k-1}X^{k-1}$
- The systematic version of the linear $C_{cyc}(n, k)$ can be obtained by performing the following operations:
  - The polynomial $X^{n-k}.m(X) = m_0X^{n-k} + m_1X^{n-k+1} + \cdots + m_{k-1}X^{n-1}$ is first formed
  - then divided by the generator polynomial $g(X)$:

Cyclic Codes in Systematic Form

- The result is:
  - $X^{n-k}.m(X) = q(X).g(X) + p(X)$
    - $p(X)$ is the remainder polynomial of the division, which has degree $n-k-1$ or less, since degree of $g(X)$ is $n-k$
- Reordering last equation:
  - $X^{n-k}\,m(X) - p(X) = q(X).g(X)$ ----> Factor of $g(X)$
- This procedure allows the code polynomial to adopt the systematic form!!
- When expressed as a code vector is equal to:
  - $c = (p_0, p_1, \ldots, p_{n-k-1}, m_0, m_1, \ldots, m_{k-1})$

Generator Matrix of a Cyclic Code

- As seen previous, a linear $C_{cyc}(n, k)$ generated by the generator polynomial:
  - $g(X) = 1 + g_1X + \cdots + g_{n-k-1}X^{n-k-1} + X^{n-k}$
- is spanned by the k code polynomials:
  - $g(X), X.g(X), \ldots, X^{n-k}.g(X)$
  - which can be represented as row vectors of a generator matrix of dimension $k \times n$:

$$\boldsymbol{G} = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & \cdots & 0 \\ \vdots & \vdots & & & \vdots & \vdots & & & \vdots \\ 0 & 0 & & \cdots & g_0 & g_1 & g_2 & \cdots & g_{n-k} \end{bmatrix}$$

Generator Matrix of a Cyclic Code

- Example:
  - For the linear $C_{cyc}(7, 4)$ generated by the polynomial $g(X) = 1 + X + X^3$, determine G and then convert it into a systematic generator matrix.
- Resolution:

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \quad G' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \quad G'' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

replacing the third row by the addition of the first and third rows,

replacing the fourth row by the addition of the first, second and fourth rows,

  - This last modified matrix G'' generates the same code as that of the generator matrix G.

Syndrome Calculation

- As defined for block codes, the received vector, which is the transmitted vector containing possible errors, is r = (r_0, r_1, ..., r_{n-1}). This is a vector with elements which can also have a polynomial representation:
  - $r(X) = r_0 + r_1X + r_2X^2 + \cdots + r_{n-1}X^{n-1}$
- Dividing this polynomial by $g(X)$ gives:
  - $r(X) = q(X).g(X) + S(X)$
  - the remainder of this division is a polynomial of degree $n - k - 1$ or less.

Syndrome Calculation

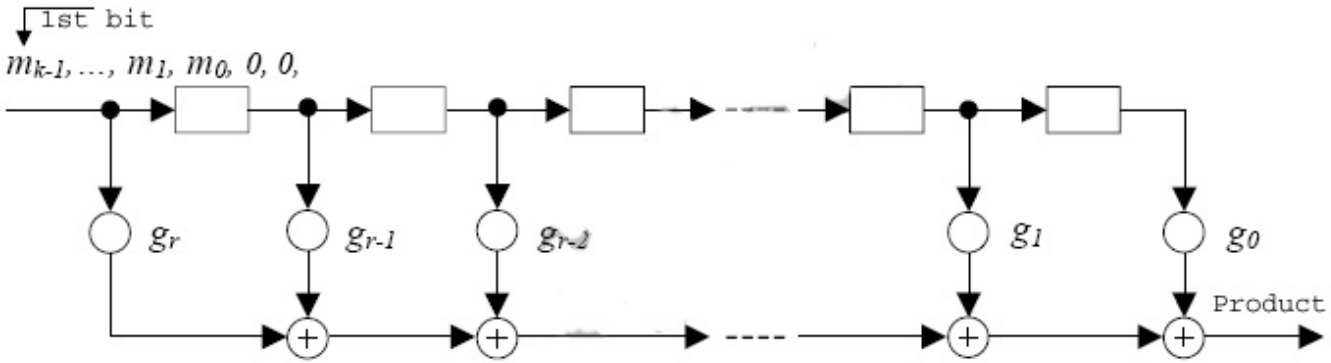
- Since a code polynomial is a multiple of $g(X)$, then if the remainder of the division is zero, the received polynomial is a code polynomial.
- If the division has a non-zero polynomial as the remainder, then the procedure detects a polynomial that does not belong to the code
- The syndrome vector is again a vector of $n - k$ components that are at the same time the coefficients of the polynomial:
  - $S(X) = s_0 + s_1X + \cdots + s_{n-k-1}X^{n-k-1}$.

Decoding of Cyclic Codes

- A table S → e identifying the relationship between the syndromes and the error patterns is constructed
- Syndrome polynomial is evaluated for the received polynomial r (X)
  - dividing this polynomial by the generator polynomial $g(X)$ to obtain the syndrome polynomial.
- The constructed table allows us to identify the error pattern that corresponds to the calculated syndrome.

Cyclic Codes Circuits

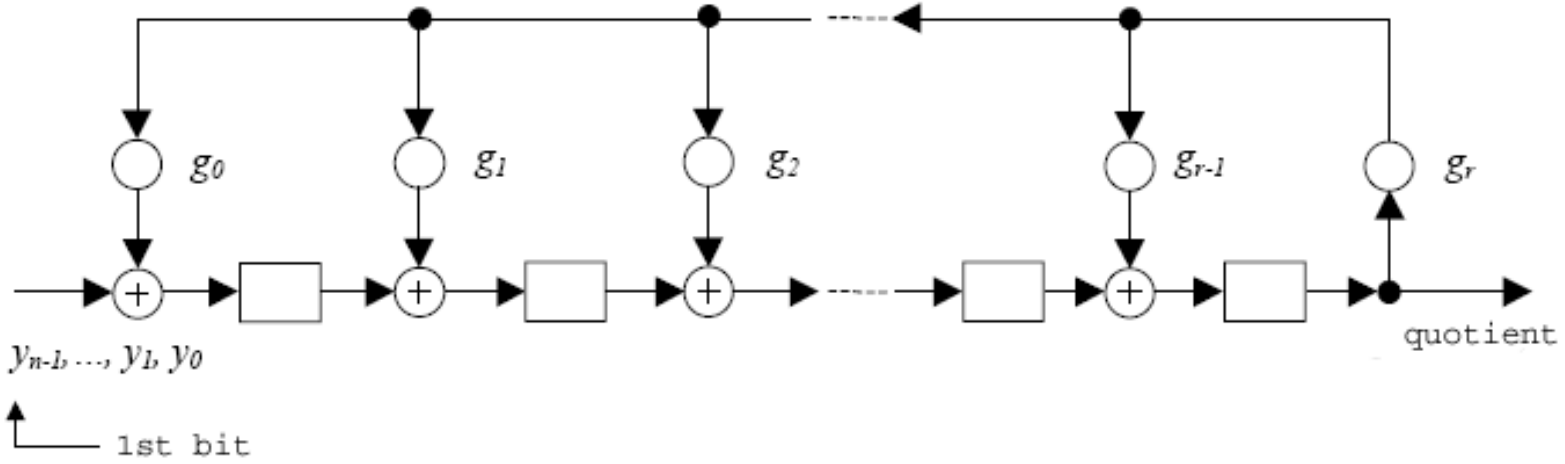
- Circuit to multiply *m(x).G(x)*



- Multiplication ends when the last input bit cross shift-register. It is necessary to continue shifting with a number of “0” equal to the number of levels

Cyclic Codes Circuits

- Circuit to divide *y(x):G(x)*

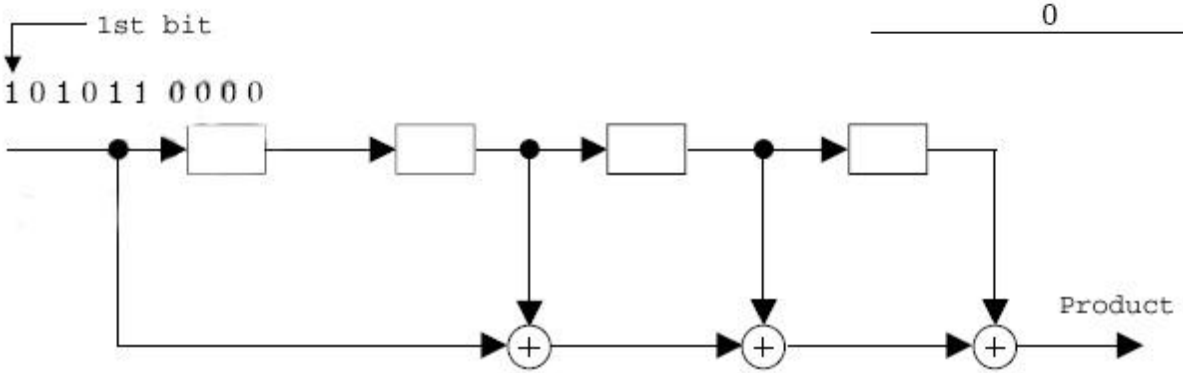


- Number of levels = degree of *g(x)*
- After *n* shifts the quocient is at the output and …
- … the remainder is stored at the register

Cyclic Codes Circuits

- Example:
  - $A(x) = x^5+x^3+x+1$
  - $B(x) = x^4+x^2+x+1$
    - $A(x).B(x) = ?$

	Input	Register state	Output	
(x^5)	1	0000	1	(x^9)
(x^4)	0	1000	0	(x^8)
(x^3)	1	0100	0	(x^7)
(x^2)	0	1010	1	(x^6)
(x)	1	0101	1	(x^5)
1	1	1010	0	(x^4)
	0	1101	0	(x^3)
	0	0110	0	(x^2)
	0	0011	0	(x)
	0	0001	1	(1)

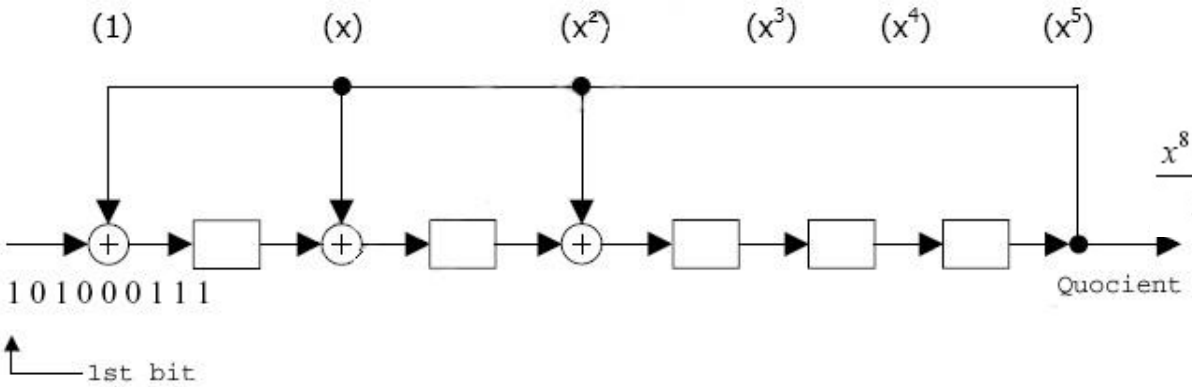


Cyclic Codes Circuits

- Example:
  - $C(x) = x^8 + x^6 + x^2 + x + 1$
  - $D(x) = x^5 + x^2 + x + 1$
    - $C(x) \div D(x) = ?$

Input		Register	Output	
(x^8)	1	00000	—	
(x^7)	0	10000	0	(x^8)
(x^6)	1	01000	0	(x^7)
(x^5)	0	10100	0	(x^6)
(x^4)	0	01010	0	(x^5)
(x^3)	0	00101	0	(x^4)
(x^2)	1	11110	1	(x^3)
(x)	1	11111	0	(x^2)
(1)	1	00011	1	(x)
	—	01101	1	(1)

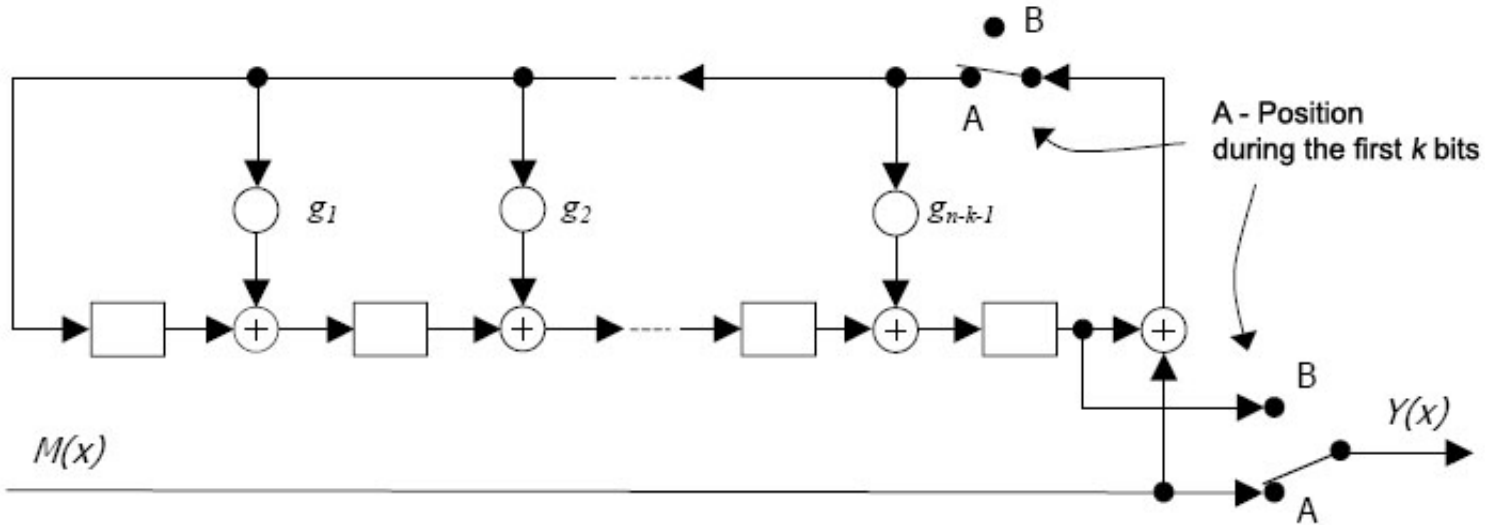
Remainder: $x^4 + x^2 + x$



$$\frac{x^8 + x^6 + x^2 + x + 1}{x^5 + x^2 + x + 1} = x^3 + x + 1 + \frac{x^4 + x^2 + x}{x^5 + x^2 + x + 1}$$

Cyclic Codes Circuits

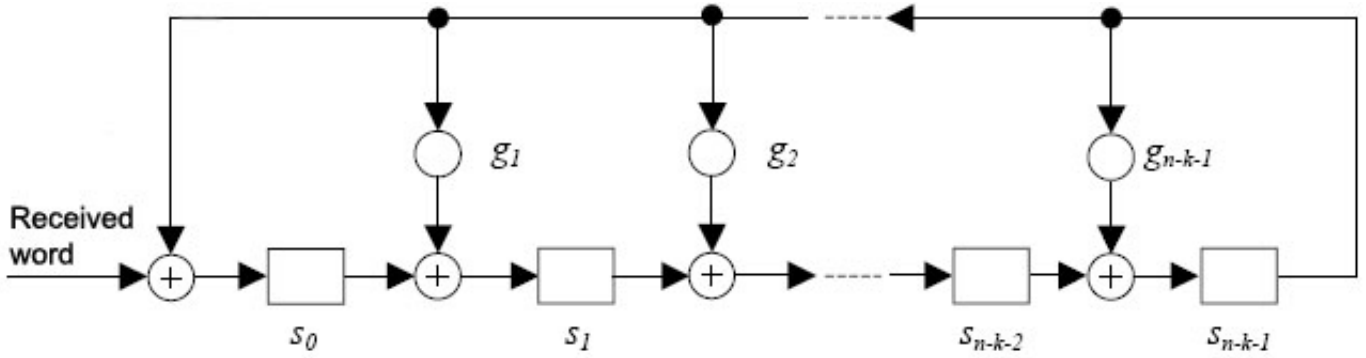
- Systematic Encoder



- During the first *k* bits both switches are in *A* – Position
- During the *n* - *k* parity bits the switches are in *B* – Position
- While the lower switch is in *A*, the *k* information bits go directly to the output
- In *B*, the contents of the shift-register (parity bits) go to the output

Cyclic Codes Circuits

- Syndrome Calculator



- To evaluate the syndrome we only need to divide the received word by G(x). This circuit is the same of that we used to divide two polynomials.

Reading Material

- *S. Lin and D. J. Costello. Error Control Coding. Prentice-Hall, Upper Saddle River, NJ, 2nd edition, 2004 (ISBN 0-13-017973-6).*
- *Sílvio A. Abrantes. Códigos Correctores de Erros em Comunicações Digitais, FEUP Edições, Porto, 2010 (ISBN 978-972-752-127-2).*
- *Simon Haykin. Communication Systems, McMaster Univ, 4th Edition, 2004. (ISBN: 978-0-471-17869-9).*