

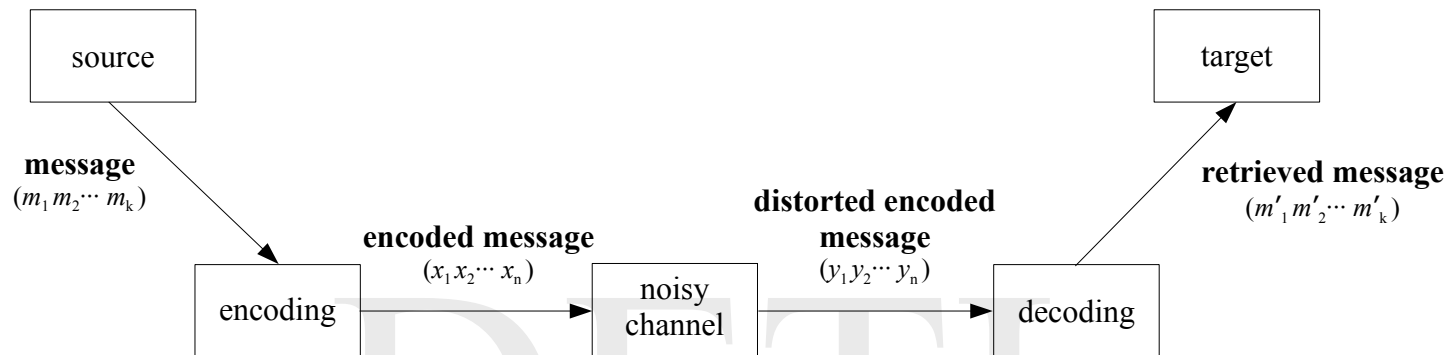


Arquitecturas de Alto Desempenho

Design Principles for Hamming Codes

António Rui Borges

Theoretical Background - 1



A message \mathbf{m} , generated by a given *source*, is expressed in k symbols of the alphabet Σ and is further encoded into a word \mathbf{x} , using a *block code*, through expansion to length n by addition of redundant information; that is, a specific code word is built by the interspersing of symbols of the alphabet Σ to the symbols of the message, following some definite rule.

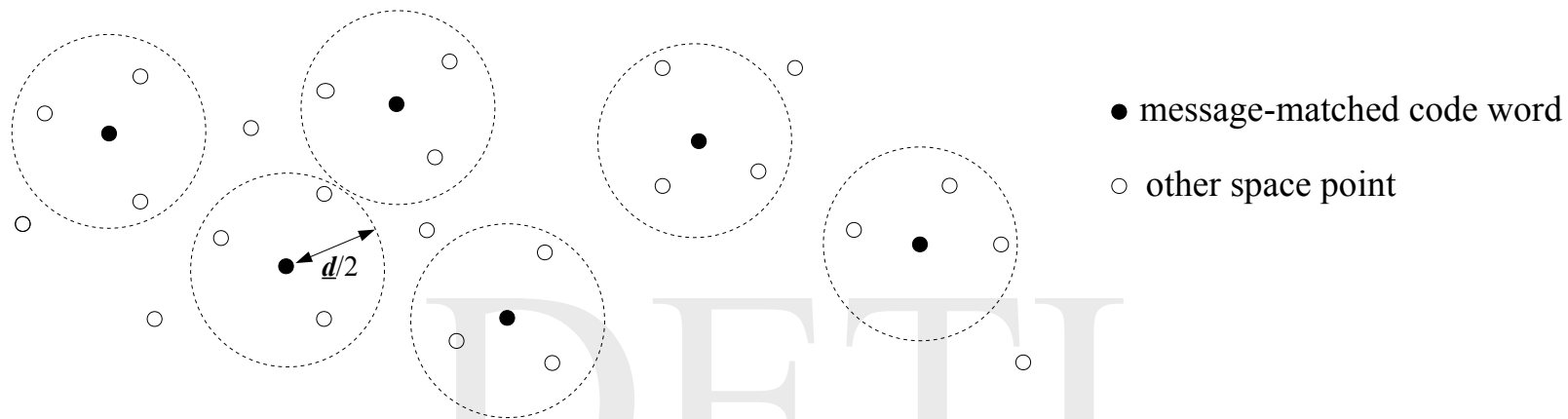
The encoded message \mathbf{x} is next transmitted over a *noisy channel*, where the symbols may be changed according to certain probabilities that are characteristic of the channel. The received message \mathbf{y} is finally decoded into the message \mathbf{m}' , which is retrieved by the *target*. Bear in mind that the *channel* concept must be taken in a broad sense: it can be a data transmitting medium in the traditional sense, such as copper, optical fiber, or air signal propagation, or a data storage device, such as a hard disk, a flash memory, or a dynamic RAM.

Theoretical Background - 2

One can define the *information rate* R , which measures the slowdown of the effective data transmission, as k/n . On the other hand, given the channel characteristics, one defines the *capacity* C of the channel as something which, as Claude Shannon has shown, has the property that, for $R < C$, it is possible to find an *encoding/decoding* scheme such that the probability that $\mathbf{m} \neq \mathbf{m}'$ can be made arbitrarily small. If, however, $R > C$, no such scheme exists.

Claude Shannon, however, did not show how to build such *encoding/decoding* schemes. This has been the pioneering work of Richard Hamming. Presently, however, the case for the best codes in terms of the maximal number of errors that one can correct for a given information rate and code length is not clear. Existence theorems are known, but the exact bound is still an open problem.

Theoretical Background - 3



To understand how one can achieve an *error-correcting* code by increasing the number of symbols in the code word relative to the message word, one may think of each code word as a point in a n -dimensional space where a *metric* is defined, that is, a real-valued function $d(\mathbf{x}, \mathbf{y})$ which, given two arbitrary space points, \mathbf{x} and \mathbf{y} , computes the distance between them.

Now, if one can find an *encoding* scheme which matches each message word to code words that are as widely-spaced as possible within the n -dimensional space, then all space points in the neighborhood of radius $d/2$ of each message-matched code word, where d represents the minimum value of the distance between two arbitrary message-matched code words, can be *decoded* as matching the corresponding message word through the use of a minimum distance criterion.

Hamming codes - 1

Hamming codes are able to detect and correct one symbol of the received message. The metric, $d(\mathbf{x}, \mathbf{y})$ is here defined as the number of positions where the symbols of the code words \mathbf{x} and \mathbf{y} differ, the so-called *Hamming distance*.

In the binary case, the alphabet Σ is the set $\{0, 1\}$ and the space of code words is F_2^n , where F_2 is the Galois field of two elements. By taking the number of redundant symbols, *parity bits*, to be r , one gets

$$\begin{aligned} n &= 2^r - 1 && \text{for the code length} \\ k &= n - r && \text{for the message length} \end{aligned}$$

giving rise to a $[n, k]$ block code, with a minimum distance between code words of 3.

Hamming codes - 2

The encoding process is described by

$$\mathbf{x} = \mathbf{m} \cdot \mathbf{G}$$



where \mathbf{m} is the message, \mathbf{x} its code word and \mathbf{G} the $k \times n$ code generating matrix, defined as

$$\mathbf{G} = \left\| \mathbf{I}_k \mid -\mathbf{A}^T \right\|.$$

The *decoding* process, on the other hand, is described by

$$\mathbf{H} \cdot \mathbf{y}^T = \mathbf{0} \Rightarrow \text{message bits of } \mathbf{y} \text{ are correct (there is no error)}$$

$$\mathbf{H} \cdot \mathbf{y}^T \neq \mathbf{0} \Rightarrow \text{error on bit whose position is the column number whose contents is } \mathbf{H} \cdot \mathbf{y}^T$$

where \mathbf{y} is the received encoded message and \mathbf{H} the $r \times n$ parity check matrix, defined as


$$\mathbf{H} = \left\| \mathbf{A} \mid \mathbf{I}_r \right\|$$

and has as columns all the pairwise linear independent vectors of length r .


Computation of a parity-check matrix for [15,11]

$$\mathbf{H} = \left\| \mathbf{A} \mid \mathbf{I}_4 \right\| =$$

$$= \left\| \begin{array}{cccccccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{array} \right\| \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right\|$$



\mathbf{A}



\mathbf{I}_4

Computation of the associated code generating matrix

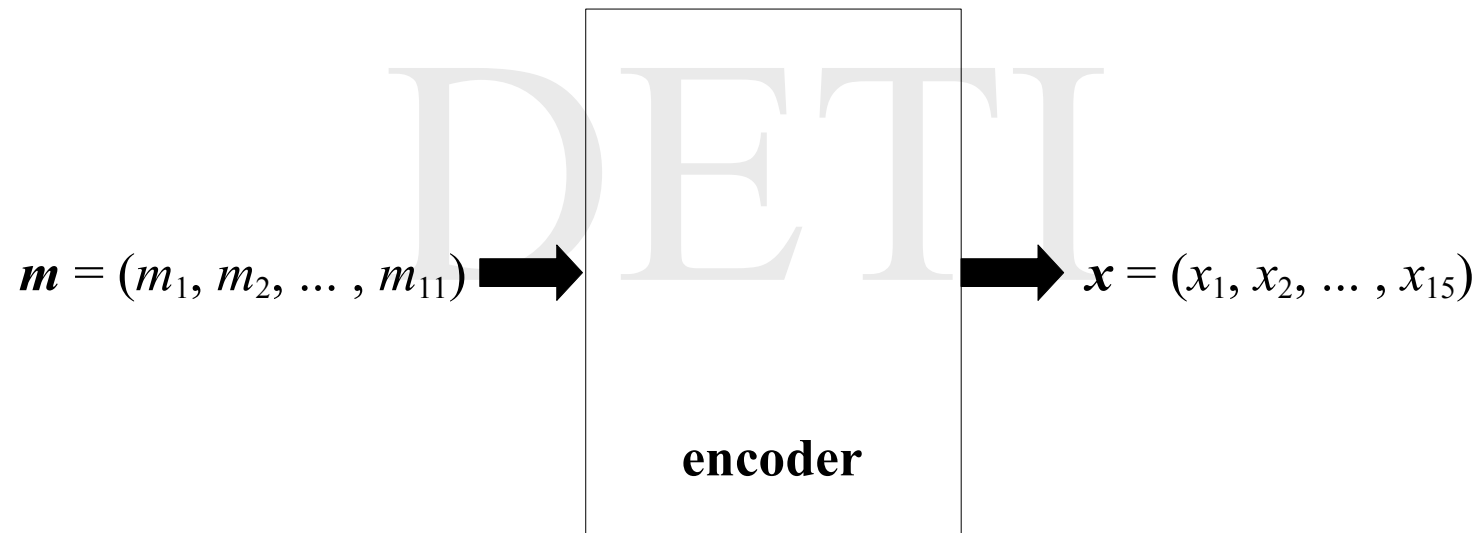
$$\mathbf{G} = \left\| \mathbf{I}_{11} \mid -\mathbf{A}^T \right\| =$$

$$= \left\| \begin{array}{cccccccccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array} \right\|$$

$\underbrace{\hspace{15em}}_{\mathbf{I}_{11}}$

$\underbrace{\hspace{5em}}_{-\mathbf{A}^T}$

Parallel implementation of the encoder for [15,11] - 1



Parallel implementation of the encoder for [15,11] - 2

Straightforward implementation

$\mathbf{x} = \mathbf{m} \times \mathbf{G}$, where matrix \mathbf{G} is fixed

$$x_i = m_i \quad , \quad \text{for } i = 1, 2, \dots, 11$$

$$x_{12} = m_1 \oplus m_2 \oplus m_3 \oplus m_7 \oplus m_8 \oplus m_9 \oplus m_{11}$$

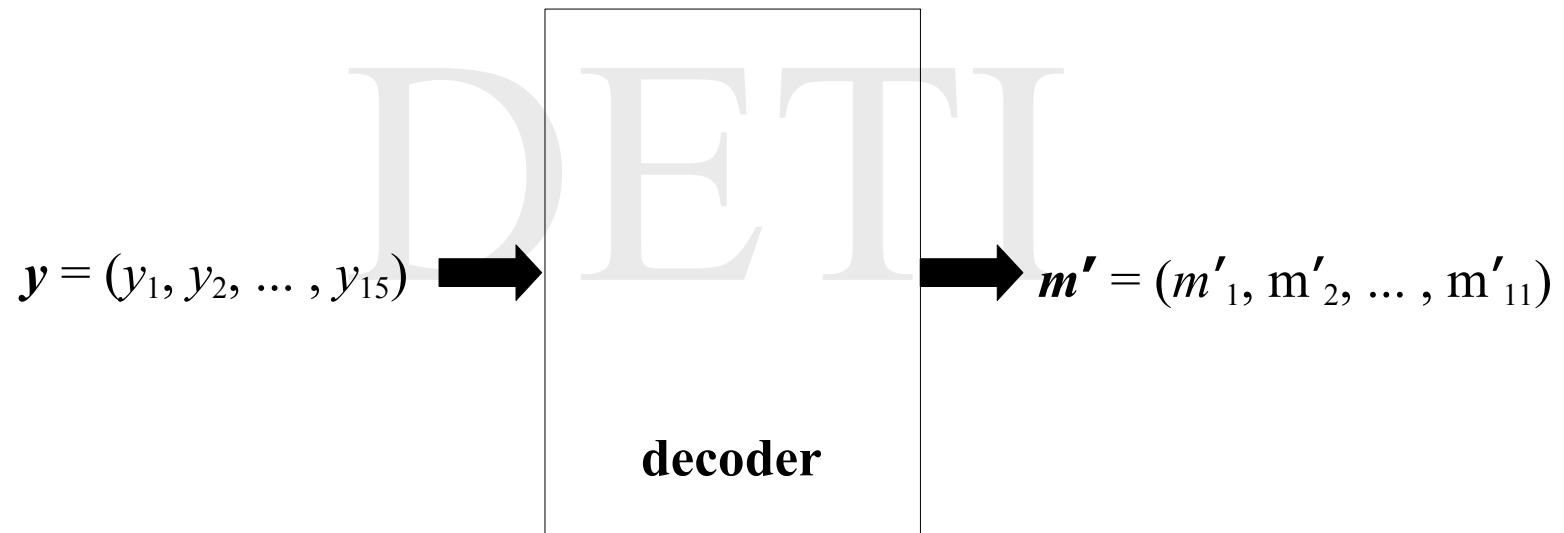
$$x_{13} = m_1 \oplus m_4 \oplus m_5 \oplus m_7 \oplus m_8 \oplus m_{10} \oplus m_{11}$$

$$x_{14} = m_2 \oplus m_4 \oplus m_6 \oplus m_7 \oplus m_9 \oplus m_{10} \oplus m_{11}$$

$$x_{15} = m_3 \oplus m_5 \oplus m_6 \oplus m_8 \oplus m_9 \oplus m_{10} \oplus m_{11}$$

- 24 x-ors are needed
- 6 x-or propagation time delays in the worst case.

Parallel implementation of the decoder for [15,11] - 1



Parallel implementation of the decoder for [15,11] - 2

Straightforward implementation of the error detecting part

$$\mathbf{p} = \mathbf{H} \times \mathbf{y}^T, \text{ where matrix } \mathbf{H} \text{ is fixed}$$

$$p_1 = y_1 \oplus y_2 \oplus y_3 \oplus y_7 \oplus y_8 \oplus y_9 \oplus y_{11} \oplus y_{12}$$

$$p_2 = y_1 \oplus y_4 \oplus y_5 \oplus y_7 \oplus y_8 \oplus y_{10} \oplus y_{11} \oplus y_{13}$$

$$p_3 = y_2 \oplus y_4 \oplus y_6 \oplus y_7 \oplus y_9 \oplus y_{10} \oplus y_{11} \oplus y_{14}$$

$$p_4 = y_3 \oplus y_5 \oplus y_6 \oplus y_8 \oplus y_9 \oplus y_{10} \oplus y_{11} \oplus y_{15}$$

- 28 x-ors are needed
- 7 x-or propagation time delays in the worst case.

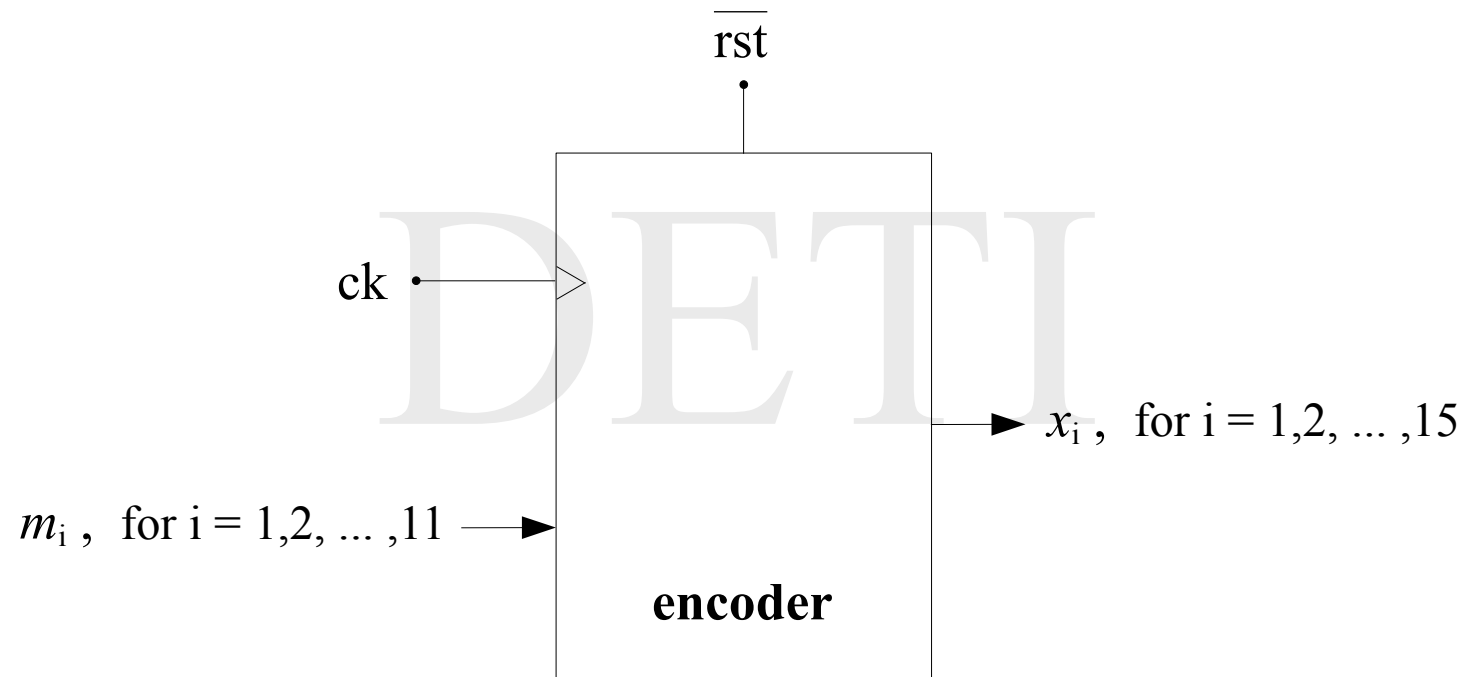
Parallel implementation of the decoder for [15,11] - 3

Implementation of the error correcting part

$$\mathbf{p} = \mathbf{H} \times \mathbf{y}^T \neq 0 \Rightarrow \exists_k \, ml_k = \overline{y_k}$$

$$ml_i = sel_i(\mathbf{p}) \oplus y_i, \text{ for } i = 1, 2, \dots, 11$$

Bit-serial implementation of the encoder for [15,11]



Bit-serial implementation of the decoder for [15,11]

