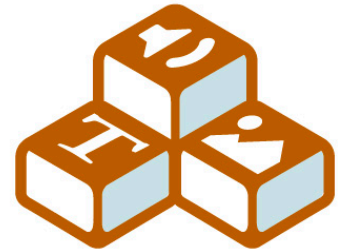


Información Digital

Representación y Codificación



El texto que sigue ha sido seleccionado por el profesorado como una lectura recomendada para el MOOC “Información Digital”, organizado por la Universidad de Granada.

Sólo puede emplearse dentro de los límites de la licencia de uso indicada en el documento. En cualquier caso, siempre que difunda este documento debe citar los autores y referencia originales.



Extraído de:
Introducción a la Informática; A Prieto, A Lloris, JC Torres, 4ª Edic. 2006.
McGraw-Hill.

Introducción a la Informática

18.2.6 Control de errores en comunicaciones digitales

En comunicaciones digitales a un conjunto predeterminado de bits consecutivos lo denominamos **unidad de datos**. Antes de transmitir una unidad de datos se le añaden, en el receptor, redundancias para que el receptor pueda comprobar si durante la transmisión se han producido errores (ruido ambiental, interferencias, etc.). Los errores pueden ser de los siguientes tipos (Tabla 18.6):

- **Errores en bits aislados.**
- **Errores en bits múltiples**, cuando dos o más bits no consecutivos en una secuencia dada (unidad de datos) cambian.
- **Errores en ráfagas**, cuando dos o más bits consecutivos en una unidad de datos cambian.

Tabla 18.6. Errores producidos en la transmisión y formas de detectarlos.

Tipos de errores	<ul style="list-style-type: none"> – Errores en bits aislados. – Errores en bits múltiples. – Errores en ráfagas.
Métodos de detección de errores	<ul style="list-style-type: none"> – Comprobación de redundancia vertical (VCR, <i>Vertical Redundancy Checking</i>). – Comprobación de redundancia horizontal (LCR, <i>Longitudinal Redundancy Checking</i>). – Suma de comprobación (<i>Checksum</i>). – Redundancias cíclicas (CRC, <i>Cyclic Redundancy Checking</i>).

La inclusión de redundancias para control de errores suele realizarse añadiendo al final de la unidad de datos una información calculada en función de su contenido. Esta información adicional se produce en el emisor. Al llegar la unidad de datos junto con los bits de control de error al receptor, éste recalcula los bits de control y comprueba si coinciden con los bits de error recibidos. Si no es así, claramente ha habido un error en la transmisión, hecho que el receptor comunica al emisor para que le retransmita nuevamente la unidad de datos junto con sus bits de control de error.

En esta sección se dan unas ideas sobre los procedimientos más usados para calcular los bits de comprobación de errores (Tabla 18.6). Hay otros métodos, y lo que se pretende con todos ellos es detectar el mayor número posible de errores.

La forma más sencilla de detectar errores consiste en incluir bits de paridad (Sección 3.7). Usualmente lo que se hace es añadir al final de cada unidad de datos un bit, de tal forma que el número total de unos de la unidad transmitida (unidad de datos original junto con su bit de paridad) sea par (o impar). El receptor comprueba si el número de bits uno es par (o impar), y en el caso de que no lo sea ordena que se le retransmita el carácter. Este procedimiento

se denomina **bit de paridad de redundancia vertical, VCR**. Este sistema, como se indicó en la Sección 3.7, tiene el inconveniente de que no se pueden detectar errores cuando se producen en un número par de bits (cuando a causa de errores de transmisión se alteran 2, 4, 6,... bits). Para mejorar la detección de errores, se suele añadir un **bit de paridad de redundancia horizontal, LCR**. Con este procedimiento (Figura 18.17) cada cierto número k de unidades de datos ($k = 8$, en el ejemplo de la figura) se añade un carácter de comprobación. El bit menos significativo de la unidad de comprobación corresponde a la paridad de los k bits menos significativos de las unidades del mensaje, el bit de orden uno del carácter de comprobación es el bit de paridad correspondiente a los k bits de orden uno de las unidades de datos, y así sucesivamente. Recordar que, según se vio en el Capítulo 4, el bit de paridad (criterio par) se obtiene sin más al hacer la operación exclusive-OR (o suma módulo 2) de los bits correspondientes.

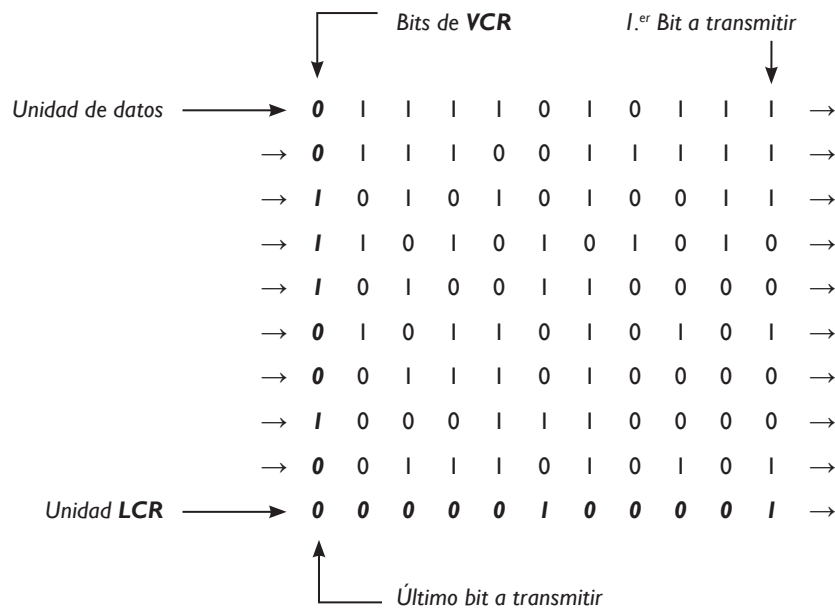


Figura 18.17. Bits de paridad de redundancia vertical (VCR) y de redundancia horizontal (LCR). (Las flechas horizontales indican el orden en que se transmiten los bits).

Otro método de detección de errores muy usado en transmisiones es el de la **suma de comprobación**. Como en el caso de bit de paridad de redundancia horizontal, consiste en dividir la unidad de datos (de m bits), en n bloques de k bits. Las redundancias a incluir se forman sumando los k bloques en complemento a uno, y complementando el resultado final. Los k bits así obtenidos, S_E , se incluyen al final de los m bits de la unidad de datos original. En el receptor, cuando llega una nueva unidad de datos completa, se extraen los primeros n bloques y se efectúa su suma, S_E . Posteriormente se suman la S_E recibida con la S_R calculada y, si la transmisión ha sido correcta, el resultado será cero.

Cuando se producen errores a ráfagas (cambian varios bits seguidos), los procedimientos de detección anteriores son prácticamente inútiles. Uno de los procedimientos más usados consiste en utilizar codificación **CRC** procedimiento también denominado de **codificación polinómica**.

Como en los casos anteriores, las unidad de datos se divide en bloques sucesivos de k bits. A los k bits del mensaje se le añaden r bits de comprobación de error, obtenidos según el procedimiento que se describe a continuación.

Los k bits de cada bloque se tratan como si fuesen los coeficientes de un polinomio, $M(x)$, de orden $k - 1$, en el que las operaciones se hacen en módulo-2. Así, por ejemplo, si el mensaje fuese (para simplificar supóngase $k = 8$):

10010110

el polinomio considerado sería:

[18.9]

$$M(x) \equiv 1 \cdot x^7 + 0 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x^1 + 0 \cdot x^0 = x^7 + x^4 + x^2 + x$$

Por otra parte se utiliza un **polinomio generador**, $G(x)$ de grado r . Este polinomio está predeterminado, y es el mismo en el emisor y el receptor. Los polinomios normalizados que más se suelen utilizar (para $k = 16$) son:

[18.10]

$$\text{CRC-16} \rightarrow G(x) \equiv x^{16} + x^{15} + x^2 + 1$$

[18.11]

$$\text{CRC-CCITT} \rightarrow G(x) \equiv x^{16} + x^{12} + x^5 + 1$$

El objetivo del procedimiento consiste en añadir r bits al mensaje de k bits, de forma tal que el polinomio, $T(x)$ correspondiente a los $k + r$ bits sea divisible por $G(x)$. El receptor verifica si $T(x)$ es divisible por $G(x)$, en caso de no serlo hay un error en la transmisión.

El algoritmo es el siguiente:

- a) Añadir r bits ceros al extremo de menor orden del mensaje. El polinomio correspondiente será: $x^r \cdot M(x)$
- b) Dividir (módulo 2) $x^r \cdot M(x)$ entre $G(x)$:

[18.12]

$$\frac{x^r \cdot M(x)}{G(x)} = C(x) + \frac{R(x)}{G(x)}$$

Donde $C(x)$ es el cociente y $R(x)$ es el resto. $R(x)$ siempre es menor que $G(x)$, es decir, será de grado r o menor.

- c) Restar a $x^r \cdot M(x)$ el valor del resto $R(x)$:

[18.13]

$$T(x) \equiv x^r \cdot M(x) - R(x)$$

Se obtiene así $T(x)$, cuyos coeficientes (unos o ceros) constituyen el mensaje a transmitir. $T(x)$ es siempre divisible por $G(x)$, ya que siempre que se reste al dividendo el resto, lo que resulta es divisible por el divisor. En efecto, de [18.12] y [18.13]:

[18.14]

$$\frac{T(x)}{G(x)} = \frac{x^r \cdot M(x) - R(x)}{G(x)} = C(x)$$

el resto resulta ser cero.

El emisor efectúa la división de $T(x)$ por $G(x)$, y si el resto no es cero hay error de transmisión.

En el algoritmo anterior, según se indicó anteriormente, las operaciones se realizan en módulo-2. Esto facilita la implementación hardware del algoritmo, ya que en las operaciones no hay que considerar acarreo ni débitos. La suma y resta en módulo-2 se reduce a efectuar la operación exclusive-OR bit a bit, y en la división módulo-2 las restas se efectúan módulo-2.

Se demuestra que con este procedimiento, utilizando el polinomio generador CRC-CITT, Expresión [18.11], se detectan:

1. Todos los errores de 1 bit individual.
2. Todos los errores que alteren dos bits.
3. Todos los errores que alteren un número impar de bits.
4. Todos los errores de ráfagas de 16 ó menos bits.
5. El 99.997% de errores en ráfagas de 17 bits.
6. El 99.998% de errores en ráfagas de 18 ó más bits.

Para más detalles pueden verse, por ejemplo, las referencias bibliográficas [Tan03, Sta05 o Leo01].