Hamming

Hoy, el código de Hamming se refiere al (7.4) que Hamming introdujo en 1950. El código de Hamming agrega tres bits adicionales de comprobación por cada cuatro bits de datos del mensaje.

El algoritmo de Hamming (7.4) puede corregir cualquier error de un solo bit, y detecta todos los errores de dos bits.

Para un ambiente en el que el ruido pueda cambiar como máximo 2 bits de 7, el código Hamming (7.4) es generalmente el de pérdida mínima.

El medio tendría que ser muy ruidoso para que se perdieran más de 2 bits de cada 7 (casi el 45% de los bits transmitidos), y habría que considerar seriamente cambiar a un medio de transmision más fiable.

El algoritmo es simple:

1. Todos los bits cuya posición es potencia de dos se utilizan como bits de paridad (posiciones 1, 2, 4, 8, 16, 32, 64, etc.).
2. Los bits del resto de posiciones son utilizados como bits de datos (posiciones 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, etc.).
3. Cada bit de paridad se obtiene calculando la paridad de alguno de los bits de datos. La posición del bit de paridad determina la secuencia de los bits que alternativamente comprueba y salta, a partir de éste, tal y como se explica a continuación.

    * Posición 1: salta 1, comprueba 1, salta 1, comprueba 1, etc.
    * Posición 2: comprueba 1, salta 2, comprueba 2, salta 2, comprueba 2, etc.
    * Posición 4: comprueba 3, salta 4, comprueba 4, salta 4, comprueba 4, etc.
    * Posición 8: comprueba 7, salta 8, comprueba 8, salta 8, comprueba 8, etc.
    * Posición 16: comprueba 15, salta 16, comprueba 16, salta 16, comprueba 16, etc.
    * Y así sucesivamente.

Así pues en la Posición 1, comprobaríamos los bits: 3, 5, 7, 9, 11...; en la Posición 2, los bits: 3, 6, 7, 10, 11, 14, 15...−; en la Posición 4 tendríamos: 5, 6, 7, 12, 13, 14, 15... Así hasta completar la nueva cadena.

Ejemplo

Consideremos la palabra de datos de 7 bits "0110101". Para ver cómo se generan y utilizan los códigos Hamming para detectar un error, observe las tablas siguientes. Se utiliza la d para indicar los bits de datos y la p para los de paridad.

En primer lugar los bits de datos se insertan en las posiciones apropiadas y los bits de paridad calculados en cada caso usando la paridad par.

Cálculo de los bits de paridad en el código Hamming p1 p2 d1 p3 d2 d3 d4 p4 d5 d6 d7
Palabra de datos (sin paridad): 0 1 1 0 1 0 1
p1 1 0 1 0 1 1
p2 0 0 1 0 0 1
p3 0 1 1 0

p4 0 1 0 1
Palabra de datos (con paridad): 1 0 0 0 1 1 0 0 1 0 1

La nueva palabra de datos (con los bits de paridad) es ahora "10001100101". Consideremos ahora que el bit de la derecha, por error, cambia de 1 a 0. La nueva palabra de datos será ahora "10001100100"; cuando se analice el modo en que se obtienen los bits de paridad en los códigos de Hamming se observarán variaciones en la paridad, lo que significará que hay error.

Comprobación de los bits de paridad (con primer bit de la derecha cambiado) p1 p2 d1 p3 d2 d3 d4 p4 d5 d6 d7 Prueba de paridad Bit de paridad
Palabra de datos recibida: 1 0 0 0 1 1 0 0 1 0 0

p1 1 0 1 0 1 0 Error 1
p2 0 0 1 0 0 0 Error 1
p3 0 1 1 0 Correcto 0
p4 0 1 0 0 Error 1

El paso final es evaluar los bits de paridad (recuerde que el fallo se encuentra en d7). El valor entero que representan los bits de paridad es 11, lo que significa que el bit décimo primero de la palabra de datos (bits de paridad incluidos) es el erróneo y necesita ser cambiado.

p4 p3 p2 p1
Binario 1 0 1 1
Decimal 8 2 1 Σ = 11

Cambiando el bit décimo primero 10001100100 se obtiene de nuevo 10001100101. Eliminando los bits de paridad de Hamming se vuelve a obtener la palabra de datos original 0110101.

Observe que en la comprobación de la paridad no se tienen en cuenta los bits de paridad. Si el error se produjera en uno de ellos, en la comprobación sólo se detectaría un error, justo el correspondiente al bit de paridad causante del mismo.

Finalmente, cuando cambien dos bits, en la comprobación de paridad se obtendrá un valor decimal superior a 11, detectándose el error; sin embargo no se podrán saber las posiciones de los dos bits que cambiaron.