FIUSAC. EIME
Laboratorio de Electrónica
Comunicaciones 2
Ing Byron Arrivillaga

Práctica # 1: “Código Hamming”

Objetivos de la práctica “Código Hamming”

- Que el alumno llegue enteneder las bondades de los codigos para correccion de errores
- Entender los pasos del codigo hamming y poder usar este conocimiento para elaborar un programa que pueda enmular este, en datos de prueba..
- Que el estudiante pueda leer un block de datos codificados en el codigo hamming y ser capaz de poder decodificar y detectar y corregir errores en este.
- Que los alumnos puedan escribir un codigo de programacion para decodificar un block de datos codificado.

Contenido teórico

Código Hamming

En informatica, **código de Hamming** es un codigo detector y corrector de errores que lleva el nombre de su inventor. En los datos codificados en Hamming se pueden detectar errores en un bit y corregirlos, sin embargo no se distingue entre errores de dos bits y de un bit (para lo que se usa Hamming extendido). Esto representa una mejora respecto a los códigos con bit de paridad, que pueden detectar errores en solo un bit, pero no pueden corregirlo. (según wikipedia)

Richard Hamming, era un professor que publico el articulo sobre la deteccion y correccion de errores en 1950, se considera hoy en dia, a el codigo Hamming como fundamental para entender la teoria de la codificación, realmente hay muchas aplicaciones practicas de este conocimiento que mencionaremos pueden ser desde modems, hasta comunicaciones satelitales.

La teoria de los codigos Hamming y la matematica que esta implica se encuentra en multiples documentos, por lo que para la formacion en este laboratorio trataremos de simplificar los metodos, de manera que el estudiante de electronica se interese y pueda aondar en el tema buscando mas del tema, que sera ampliado en la clase teorica de este mismo curso.

Conocimiento previo necesario:

- Codigo binario
  - Se requiere conocimiento de codigos binarios, conversion y operaciones basicas
- Distancia entre dos conbinaciones binarias
  - numero de bit que hay que cambiar en una para obtener otra
- Distancia minima de un codigo
  - menor distancia entre dos numeros binarios

Hamming es un codigo binario lineal que representa mensajes.

La nomenclatura de hamming usa es hamming(a,b), donde

a ->es el numero total de bits

b ->es el numero de bits de paridad

de manera que la representacion hamming(7,4), representa 4-bits usados por el mensaje y el codigo sera una palabra de codigo de 7-bit. Dos palabras de codigo distintas difieren en por lo menos tres bits. Como consecuencia de esto, hast dos errores por palabra de codigo hamming pueden ser detectado y un solo error puede ser corregido.

Ejemplo practico de uso :

consideremos el codigo hamming(7,4)

datos -> 0110

Consideraciones iniciales:

Hamming (7,4)

indica que ser seran:

4 bits de informacion

7 bit en total

7-4=3 bit de redundancia

los bit de paridad hamming estan en bit con potencia de 2

2^0	2^1	2^2	2^3
1	2	4	8

considere en binario

Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
0001	0010	0011	0100	0101	0110	0111

primer paso :

Genere una matriz con la siguiente forma definida por hamming

	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	0001	0010	0011	0100	0101	0110	0111
	P1	P2		P4			
datos							
P1							
P2							
P4							

Aqui ubicaremos los bit de paridad en las posiciones definidas por anteriormente, osea las que coninciden con ser potencia de 2

luego colocamos los datos que según el ejemplo son

datos -> 0110

estos los colocamos en orden en los bit restantes de la siguiente forma

	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	0001	0010	0011	0100	0101	0110	0111
	P1	P2	0	P4	1	1	0
datos							
P1							
P2							
P4							

Luego el llenado de datos se efecctua bajando los datos que tengan el bit de paridad hamming en 1, esto es as:

para P1 que esta codificado com 000**1** entonces el bit de paridad hamming esta en 1 debemos comparar con todos los bit de datos

- el bit de datos 3 esta codificado como 001**1** por lo que si es tomado en cuenta
- el bit de datos 5 esta codificado como 010**1** por lo que si es tomado en cuenta
- el bit de datos 6 esta codificado como 0110 por lo que no es tomado en cuenta ya que su bit de paridad hamming no esta en 1.
- el bit de datos 7 esta codificado como 011**1** por lo que si es tomado en cuenta

quedando la siguiente matriz

	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	0001	0010	0011	0100	0101	0110	0111
	P1	P2	0	P4	1	1	0
datos			**0**		**1**	**1**	**0**
P1			**0**		**1**	**X**	**0**
P2							
P4							

continuamos bajando para P2 y P3 tomando que solo tomamos en cuenta los datos que estan en la posicion que tenga el bit de hamming de P2 y P3 según el caso, en 1.

	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	0001	0010	0011	0100	0101	0110	0111
	P1	P2	0	P4	1	1	0
datos			**0**		**1**	**1**	**0**
P1			0		1	X	0
P2			0		X	1	0
P4			X		1	1	0

ahora llenamos los bit de paridad de las siguiente manera,

para el bit 1

	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	0001	0010	0011	0100	0101	0110	0111
	P1	P2	0	P4	1	1	0
datos			**0**		**1**	**1**	**0**
P1			0		1	X	0
P2			0		X	1	0
P4			X		1	1	0

completamos con el operador XOR

P1	1	0	1	X	0

0 XOR 1 XOR 0 = 1
entonces:
1 XOR P1 = 0
P1 = 1

P2	0	X	1	0

0 XOR 1 XOR 0=1
entonces:
1 XOR P2 = 0
P2 = 1

P4	X	1	1	0

1 XOR 1 XOR 0=0
entonces:
0 XOR P4=0
P4 = 0

por lo tanto

	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	0001	0010	0011	0100	0101	0110	0111
	P1	P2	0	P4	1	1	0
datos			**0**		**1**	**1**	**0**
P1	1		0		1	X	0
P2		1	0		X	1	0
P4			X	0	1	1	0
Hamming(7,4)	1	1	0	0	1	1	0

como ultimo paso bajamos todos los datos en la linea de resultados

Hamming(7,4)	1	1	0	0	1	1	0

el codigo resultante es entonces: **1100110**

Decodificacion:

Siguiendo el ejemplo con un codigo resultante
1100110 evaluaremos con un codigo con un error este seria:
1100100 este codigo tiene el error en el bit 6 marcado en rojo para mejor comprencion.

El procedimiento de decodificacion y correccion de errores es muy parecido, construimos una matriz igual con tres olumnas extras, una para la paridad de codificacion, una para paridad de decodificacion, y otra para el calculo de error.

	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	paridad decod	paridad cod	error
	0001	0010	0011	0100	0101	0110	0111			
	P1	P2	0	P4	1	0	0			
datos			**0**		**1**	**0**	**0**			
P1	1		0	X	1	X	0	1	1	Ok(0)
P2		0	0		X	0	0	0	1	Err(1)
P4			X	1	1	0	0	1	0	Err(1)

para generar esta matriz usamos el procedimieto anterior y copiamos nuestros P1,P2,P3 en la columna paridad decodificacion, luego extraemos de la data que llego generada por el codificador en este ejemplo 1100100 , y usando el orden P1 P2 BIT1 P4 BIT2 BIT3 BIT4 quedan P1,P2, P3 = 110
Entonces llenamos los datos.

	paridad		
	paridad		
	decoded	cod	error
P1	1	1	Ok(0)
P2	0	1	Err(1)
P3	1	0	Err(1)

llenamos la columna de error, verificando si los datos son iguales,

Ahora ya teniendo los datos de error verificamos donde esta el error verificando la paridad

P3P2P1=110 (leemos el bit de la colmna error)
110= decimal 6
error entonces en el BIT6

CORRECCION:

Entonces en la trama recibida 1100100 invertimos el bit con error BIT6 1100110

QUEDANDO CORREGIDO EL ERROR.

PRACTICA :

Se coloca una block de datos hamming y se simula un error y se comprueba el poder corregir un error.
Se propone elaborar un codigo en python para realizar el codigo de hamming de 7 bit.