UNIVERSIDAD LA SALLE

Escuela de Ingeniería
Incorporada a la U.N.A.M.

PROYECTO DE UN CIRCUITO DE ELIMINACION DE ERRORES EN LA COMUNICACION ENTRE MICROPROCESADORES

Tesis Profesional

Que para obtener el título de:

Ingeniero Mecánico Eléctrico con la Especialidad en Sistemas Electrónicos y de Comunicaciones

P r e s e n t a n :

Víctor Gamaliel Gómez Guzmán

Guillermo González Grycuk

México, D. F. 1986

I N D I C E

P R O L O G O

Es indudable que el papel que desempeñan las comunicaciones en la vida diaria del hombre es de vital importancia para que ésta sea mejor en todos sus aspectos. De aquí que día con día el hombre tienda a mejorar las formas de comunicación. Basta con tomar un teléfono y digitar unos cuantos números y en breves momentos podremos escuchar la voz de una persona situada en el otro extremo de la tierra. Es suficiente oprimir un botón para ver imágenes de hechos que están ocurriendo a miles de kilómetros de nosotros y en el mismo instante. La computadora tiene miles de cables que comunican su unidad central de proceso con la memoria, sus puertos de entrada y salida con el mundo exterior y gracias a la buena comunicación entre todos estos componentes podemos hacer un número extraordinario de cálculos sin error, obtener la nómina de una empresa, ayudar a la toma de decisiones y hacer un sinnúmero de cosas más.

Debido a esa función tan importante que la comunicación sin errores debe cumplir, es que hemos decidido elaborar nuestro trabajo de tesis, poniendo especial interés en los microprocesadores de 16 bits por la difusión tan extensa que están teniendo en las diversas actividades del hombre.

Estamos seguros que la transmisión de datos seguirá siendo un área donde la ingeniería buscará nuevas y mejores alternativas de solución. Este trabajo pretende ser una pequeña semilla sembrada en ese vasto campo.

CAPITULO I.- I N T R O D U C C I O N

a) NECESIDAD DE LA UTILIZACION DE CODIGOS

La necesidad de utilizar códigos de detección y corrección de errores más eficientes ha crecido debido al problema militar y bancario. La imperiosa necesidad de transmitir datos con un alto indice de seguridad y libre de errores en estos dos campos, ha sido la base del desarrollo de códigos. El costo de los dispositivos electrónicos de estado sólido se ha decrementado casi tan dramáticamente como su tamaño. Esto ha estimulado el desarrollo de computadoras digitales y mecanismos periféricos y esto, a su vez, ha causado un gran incremento en el volúmen de datos comunicados entre máquinas de este tipo. Los sistemas de cómputo no toleran datos con errores y en algunos casos la naturaleza crítica inherente al dato, demanda el uso de transmisión libre de errores o algún tipo de código de detección o corrección de errores en las terminales. En muchos casos la segunda opción es la más económica.

Las causas de error son el ruido, la distorsión, las interferencias y la atenuación del canal de transmisión; debido a ellos, ha habido mejoras significativas en el campo de los códigos de corrección de errores. Actualmente se han creado varias clases de códigos bastante poderosos. En suma, los procedimientos de decodificación que pueden ser implementados con una modesta cantidad de hardware, han sido creados para varios de estos tipos de códigos.

Esos y otros desarrollos han hecho que el uso de códigos de corrección sea absolutamente práctico en los sistemas actuales de comunicación de datos. En el futuro, la perspectiva es que las tendencias mencionadas anteriormente continuarán y que los códigos de detección y corrección de errores llegarán a ser mucho más ampliamente usados.

En la figura 1 se muestra un diagrama de bloques de un sistema de comunicación digital. Un canal típico de transmisiones es una línea telefónica; un dispositivo típico de almacenamiento es una unidad de cinta magnética incluyendo las cabezas de lectura y escritura. La información fuente está compuesta generalmente por dígitos binarios, decimales o alguna forma de información alfabética. El codificador transforma estos mensajes a una señal aceptable por el canal (señales típicas eléctricas con algunas restricciones en potencia, ancho de banda y duración). Estas señales entran en el canal y son perturbadas por ruido. La salida entra en el decodificador, el cual toma una decisión dependiendo

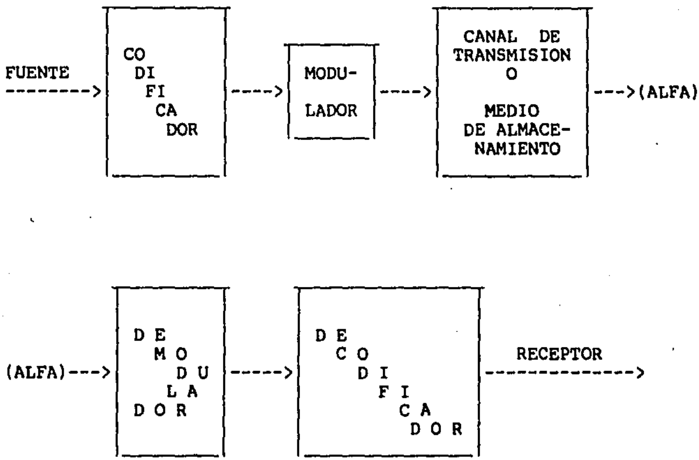


FIG. 1. DIAGRAMA DE BLOQUES DE UN SISTEMA GENERAL DE COMUNICACION DE DATOS O SISTEMA DE ALMACENAMIENTO.

del tipo de mensaje mandado y entrega este mensaje al receptor. El problema de ingenieria de comunicaciones està en el diseño del codificador y decodificador, aunque tambièn se puede agregar el de mejorar el canal de transmisiòn. Se debe tomar en cuenta ademàs que una linea telefònica conduce mejor las señales binarias, si se convierten èstas en señales senoidales. Este proceso es llamado modulaciòn. El proceso contrario, llamado demodulaciòn, se efectùa en el receptor y el aparato que efectùa ambos procesos se llama MODEM.

El anàlisis muestra que los canales de comunicaciòn de la forma de la figura 1, tienen una capacidad definida para transmisiòn de informaciòn. Aùn màs, si el rango de señal de la fuente es menor que esta capacidad definida, entonces es posible escoger un grupo de señales tales que la probabilidad de decodificar en forma errònea sea despreciable (*1).

Esta teoria no indica precisamente còmo deben contruirse estas señales ni tampoco garantiza que tal sistema pueda ser construido en la realidad.

El uso de còdigos correctores de errores es un intento por englobar estos dos ùltimos problemas (capacidad del canal y decodificaciòn libre de errores). En la figura 2 se muestra un sistema de comunicaciòn de datos que emplea un còdigo corrector de errores.

Los convertidores en este sistema traducen los simbolos de un alfabeto a los simbolos de otro. Generalmente ambos alfabetos son pequeños. Una conversiòn tipica seria de decimal a binario. El modulador acepta un simbolo ùnico a su entrada y produce en la salida la correspondiente forma de onda, la cual es el pulso asociado a ese sìmbolo. Debido al modulador, sòlo se maneja un simbolo de canal a la vez y esto es una restricciòn que causa pèrdida en la capacidad del canal.

El demodulador realiza la operaciòn inversa al modulador, o sea, recibe una cierta forma de onda afectada de ruido e intenta asociarla con su correspondiente simbolo original, de tal forma que el simbolo recibido por el modulador sea el mismo que sale del demodulador.

El codificador y el decodificador se encargaràn de implementar el còdigo corrector de errores para cada simbolo recibido por el sistema.

(*1) (Fano 1961; Feinstein 1958; Kelly 1960; Shannon 1949, 1957)

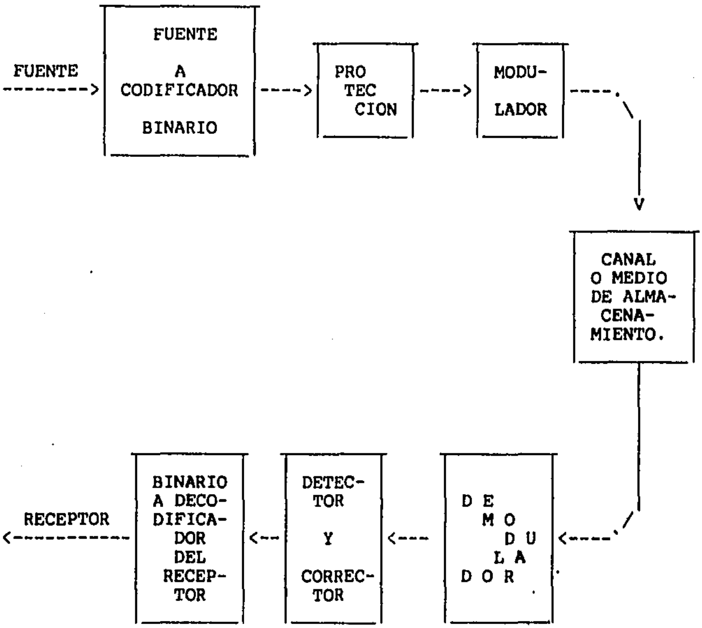


FIG. 2. DIAGRAMA DE BLOQUES DE UN SISTEMA TIPICO DE COMUNICACION DE DATOS QUE EMPLEA UN CODIGO CORRECTOR DE ERRORES.

Aún cuando dividiéramos al receptor en demodulador y decodificador y al transmisor en codificador y modulador, reduciendo así la funcionalidad últimamente accesible a un sistema de transmisión de datos, aún así es posible conseguir una relativa baja probabilidad de decodificación errónea en tal sistema, aunque de esta manera, la capacidad del canal será algo significativamente menor. Desde un punto de vista práctico, el uso de códigos hace posible el diseño y la construcción de terminales de transmisión y almacenamiento de datos altamente efectivas.

En un sistema ideal, el símbolo que sale del codificador y el símbolo recibido en el decodificador debieran compaginar. En un sistema práctico existen algunos errores ocasionales y es el propósito de los códigos el detectar y corregir tales errores. Estos códigos no pueden corregir cualquier error imaginado, sino más bien deben ser diseñados para corregir solamente los patrones de errores más comunes. La mayor parte de la teoría de codificación asume que cada símbolo es afectado independientemente por el ruido, de manera que la probabilidad de error de un patrón dado de errores depende solamente del número de errores. Así, por ejemplo, es que se han desarrollado códigos para corregir patrones de "r" o menos errores en un bloque de "n" símbolos (*2). En el capítulo relativo a códigos de bloque se definen estas variables.

Mientras que este tal vez sea un modelo apropiado para algunos canales, en las líneas telefónicas y en sistemas de almacenamiento por cinta magnética los errores ocurren predominantemente en las interrupciones violentas. Los disturbios que presentan las líneas telefónicas tales como son por relámpagos o transitorios por switcheo, duran mucho más que el tiempo de transmisión de un símbolo. De manera análoga, los defectos en cinta magnética son típicamente mayores que el espacio requerido para almacenar un símbolo. De aquí que por cuestiones de aprovechamiento de capacidad del medio, se requieran códigos para corrección de paquetes de errores.

Los canales de comunicación mostrados en las figuras 1 y 2 tienen la característica de ser unidireccionales. Muy frecuentemente los sistemas de comunicación emplean canales de dos direcciones. Este factor debe considerarse para el diseño de códigos. Por ejemplo, en un canal bidireccional se puede emplear un código de detección de errores. Cuando un error es detectado en una de las terminales, se manda una señal de retransmisión y de esta manera los errores se pueden corregir en forma total, aunque con relativa efectividad debido a los costos por tiempo empleado en retransmisión en el canal.

(*2) Véase Cap. II.c.1 Códigos de Bloque.

En canales unidireccionales se puede reducir la probabilidad de error con códigos correctores de errores, pero no por detección de error y retransmisión. En un sistema de almacenamiento por cinta magnética, por ejemplo, es demasiado tarde pedir una retransmisión cuando ya la cinta ha sido almacenada por espacio de una semana o un mes y los errores son ya entonces detectados cuando se efectúa la lectura de la cinta. Codificar para códigos correctores de errores no es mucho más complejo que para detección de errores, sin embargo, es la decodificación la que requiere equipo complejo. Para transmisión desde un vehículo espacial, por ejemplo, donde la cantidad de equipo en el transmisor es mucho más importante que la cantidad total de equipo, un equipo de codificación para un sistema corrector es probablemente más práctico que una estación de retransmisión por control remoto. Los complejos procedimientos de corrección de errores pueden ser realizados desde la tierra, donde las limitaciones de equipo por espacio no son tan severas. De aquí que haya problemas de sistemas que requieran códigos de corrección de errores sin retroalimentación.

Por otra parte, existen buenas razones para usar detección de errores y retransmisión (cuando es posible) en sistemas prácticos. Por su naturaleza, la detección de errores es una tarea mucho más simple que la correción de errores y requiere de equipo de decodificación más simple. Además la detección de errores con retransmisión es adaptiva ya que se incrementa la redundancia de información en la transmisión, al aparecer errores. Esto hace a veces posible, bajo ciertas circunstancias, el tener mejores resultados con un sistema de esta clase que es teóricamente posible en un canal unidireccional.

La eficiencia es limitada en un sistema que use únicamente detección de errores simple. Si se utilizan códigos de detección de errores cortos, no se llega a detectar todos los errores eficientemente, mientras que si se usan códigos demasiadio largos, se abusará de la retransmisión. Es conveniente una combinación de los patrones de errores más comunes junto con la detección y retransmisión para patrones de errores menores. Algunos sistemas construidos han demostrado claramente el potencial de la combinación de corrección y detección de errores con retroalimentación (*3).

(*3) (Burton y Weldon 1956; Lebow et al., 1963)

b) CAMPOS DE APLICACION MAS COMUNES. TIPOS DE CODIGOS.

El codificador de la figura 2 acepta en su entrada una secuencia continua de digitos de información. A su salida produce otra secuencia con algunos digitos de más que son alimentados al modulador. A la inversa, el decodificador acepta una secuencia de simbolos del canal provenientes del demodulador y lo traduce en una secuencia menor de digitos de información. El codificador y el decodificador operan bajo diferentes reglas, dependiendo del código particular de que se haga uso.

Existen principalmente dos tipos de códigos. El primero es llamado código de bloque debido a que su codificador interrumpe la secuencia continua de digitos de información para formar bloques de "n" simbolos. De esta manera opera independiente sobre los bloques de acuerdo al código particular que se vaya a emplear. A cada bloque de posible información está asociada una palabra "x" del código, donde x > n. El resultado, ahora llamado palabra codificada, se transmite junto con el ruido que le afecta y es decodificado independientemente de cualquier otra palabra codificada.

El segundo tipo de código, llamado código de árbol, opera en la secuencia de información pero sin interrumpirla para formar bloques independientes. El codificador para un código de árbol procesa una secuencia continua de información y asocia a cada secuencia larga de información una secuencia de código que contiene varios digitos de más. El codificador interrumpe la secuencia de entrada a bloques de n_0 simbolos, donde "n_0" es un número pequeño. Después, basado en estos n_0 simbolos y en la información procedente, emite una sección de "k_0" simbolos en la secuencia del código (*4). El nombre "códigos de árbol" viene dado por el hecho de que las reglas de codificación para este tipo de código son en su mayoria descritos convenientemente por una gráfica de árbol o, siendo más especificos, por las ramificaciones.

De las dos clases de códigos, los códigos de bloque son los que tienen considerablemente su teoria más desarrollada. Debido a que los códigos de bloque tienen una estructura matemática establecida, han alcanzado más utilización que los códigos de árbol.

Un subgrupo de los códigos de árbol son los códigos convolucionales. Estos códigos son importantes de mencionar ya que su implementación es mucho más simple que cualquiera de los otros códigos de árbol. Debido a esto, sólo se considerarán en el presente trabajo los códigos de árbol convolucionales.

(*4) Ver Cap. II.c.2. Códigos Convolucionales.

Los códigos de bloque y los códigos convolucionales tienen capacidad similar para corregir errores y fundamentalmente también las mismas limitaciones. Particularizando, el teorema fundamental de Shannon para un canal ruidoso discreto es aplicable para ambos (*5). Este asevera que un canal tiene una capacidad bien definida y que usando el o los códigos apropiados, es posible transmitir información en cualquier rango menor a la capacidad del canal con relativamente poca probabilidad de decodificación errónea.

Para que un código sea efectivo, debe ser largo para que de esta manera pueda contrarrestar los efectos del ruido sobre un número largo de símbolos. Tal código tendría una posible combinación de millones de palabras codificadas y otro tanto de palabras recibidas. La forma de determinar las propiedades importantes de un código es la estructura matemática, ya que hacer tablas de un código sería extremadamente laborioso y complicado si no es que imposible. De la estructura matemática es factible implementar las operaciones de codificar por medio de equipo electrónico.

De este modo, hay tres aspectos principales que constituyen el codificar: (1) Encontrar el código que tenga la habilidad requerida para corrección de errores. Esto generalmente demanda que los códigos sean largos. (2) Encontrar un método práctico para codificar. (3) Encontrar un método práctico para efectuar la decisión en el receptor, esto es, un método para corrección de errores. El procedimiento para atacar el problema es encontrar los códigos que sean comprobados matemáticamente que cumplan con la habilidad requerida para corrección de errores. Por eso estos códigos deben tener una estructura matemática. Esta estructura matemática es aprovechada para encontrar la funcionalidad de los otros dos requerimientos: habilidad para codificar y para decodificar.

En la mayoría de los casos los códigos pueden ser generalizados al caso de "q" símbolos, donde "q" es una potencia de un número primo. Como casi todo el equipo moderno es binario, los códigos binarios han adquirido gran importancia. Además, se pueden construir nuevos códigos binarios a partir de códigos no binarios, especialmente de aquellos en los cuales "q" es una potencia de 2. Si existe especial interés en códigos binarios, solamente se debe reemplazar "q" por 2 y las palabras "símbolo de canal" por "símbolo binario".

(*5) Ver Cap. I.c.6. Canal discreto ruidoso: teorema fundamental de Shannon.

c) INTRODUCCION A LA TEORIA DE INFORMACION

Antes de continuar con cualquier otro punto acerca de la codificación y la decodificación, es necesario que tratemos algunos conceptos de la teoría de información, ya que ella nos explica en forma matemática los errores dentro de un canal de transmisión.

c.1) SIGNIFICADO DE LA TEORIA DE INFORMACION

La teoría de información es la ciencia de los mensajes, puesto que trata de establecer en forma cuantitativa las leyes que gobiernan la generación, transmisión y recepción de los mensajes.

La teoría de información no se ocupa del significado semántico de los mensajes, sino de las probabilidades que tienen en la fuente de información de ser seleccionados para transmisión o de la incertidumbre en el receptor de que los mensajes recibidos correspondan a determinados mensajes transmitidos.

Hasta antes de 1948 en que Claude E. Shannon publicó su libro "A Mathematical Theory of Communication", la ingeniería de comunicaciones tenía como objetivo reconstruir la señal en el extremo receptor del sistema de comunicación tan fielmente como había sido enviada por el transmisor. En la teoría moderna de las comunicaciones (que nace con las ideas de Shannon y de Norbert Wiener) más que reconstruir lo más exactamente posible la señal transmitida el objetivo es recuperar la información que contiene.

Por ejemplo, si la senal que sale del transmisor es de la forma:

```
 ___     ___     ___     ___
|   |___|   |___|   |___|   |
```

que representa el mensaje 1 0 1 0 1 0 1, en el receptor se puede aceptar como buena la senal:

```
/***\.../***\.../***\.../^^^\
```

que aunque difiere bastante de la forma de onda de la senal transmitida, aún puede ser de calidad aceptable para permitir recuperar la información que contiene (1 0 1 0 1 0 1).

En el libro citado, Shannon analizó el problema de cómo representar los mensajes que una fuente puede producir para que lleven la información en un sistema de comunicación que tiene como limitaciones físicas el ancho de banda y el ruido.

El estudio de este problema dio origen a la teoria de información, una rama de la ingenieria de las comunicaciones que se ocupa de los siguientes conceptos:

- La medida de la información
- La capacidad del canal de comunicación
- La codificación.

Estos conceptos se engloban en el teorema fundamental de la teoria de información que dice:

"Dada la fuente de información y un canal de comunicación, existe una técnica de codificación tal que la información puede transmitirse sobre el canal a cualquier velocidad menor que la capacidad del canal y con una frecuencia arbitrariamente pequeña de errores a despecho de la presencia de ruido".

Lo sorprendente del teorema es el postulado de la transmisión de la información libre de errores, aunque el canal sea ruidoso, si se emplea una codificación adecuada.

c.2) MEDIDA DE LA INFORMACION

En el lenguaje diario el significado de la palabra información es el conocimiento, noticia, inteligencia, informe, etc. En una forma más precisa podemos considerar el significado de información como algo que reduce nuestro grado de incertidumbre sobre una situación.

El recibir información por medio de un mensaje implica que tenemos incertidumbre antes que recibamos el mensaje. Si la cantidad de incertidumbre removida en nosotros por la recepción del mensaje es poca, consideramos que dicho mensaje tiene poca información.

Gracias a los trabajos de Nyquist, Hartley y Shannon la información es medible.

La base de la medida de la información está en su definición. Mensajes que reducen en el receptor una gran cantidad de incertidumbre llevan mucha información; mensajes que reducen poca cantidad de incertidumbre llevan poca información. Ahora bien, nuestro grado de incertidumbre sobre una situación es una función inversa de la probabilidad de que tal situación ocurra.

Si la probabilidad de que un evento ocurra es de 100%, no tenemos ninguna incertidumbre sobre su ocurrencia y por lo mismo, un mensaje comunicándonos que ese evento ha ocurrido no reduce en nosotros ninguna incertidumbre y por lo tanto la información del mensaje es cero. Ejemplo de esto es un mensaje comunicándonos que el dia estuvo frio en Alaska.

En cambio si la probabilidad de que un evento ocurra es muy pequeña, nuestra incertidumbre sobre su ocurrencia es grande y por lo mismo, un mensaje comunicándonos que ese evento ha ocurrido reduce en nosotros una gran incertidumbre y por ende tal mensaje posee gran cantidad de información.

Consideremos una fuente de información que genera "n" mensajes cada uno de ellos con igual probabilidad de ser seleccionados para transmisión. En estas condiciones, la probabilidad que tiene un mensaje de ser seleccionado es una función de "n"; cuanto más grande sea "n", hay más mensajes por lo que menor será la probabilidad de que un mensaje cualquiera sea escogido para transmisión. Cuanto más pequeña sea "n" hay menos mensajes y mayor será esa probabilidad.

El caso más sencillo de una fuente de información es aquél en que sólo hay dos mensajes igualmente probables para seleccionar. Aqui únicamente hay dos alternativas: si o no, 0 ó 1, abierto o cerrado, blanco o negro, águila o sol, etc. La información asociada con la seleción de uno de estos mensajes es la cantidad minima de información que una fuente puede generar y se ha tomado como la unidad de medida de la información, designándola con el nombre de BIT (BInary digiT).

Se tiene entonces:

Mensajes probables (n)	Cantidad de información (I)
1	no
2	1
4	2
8	3
n	$\log_2 n$

De lo anterior se puede deducir que la cantidad de información viene dada por la siguiente expresión:

$$I = \log_2 n \qquad (1)$$

Si hay n mensajes equiprobables, la probabilidad de que uno de ellos ocurra es:

$$p = 1/n$$

Entonces:

$$I = \log_2 n = \log_2 (1/p) \qquad (2)$$

c.3) ALGUNAS APLICACIONES.

Supongamos que una fuente genera los números decimales (1,2,3,4,5,6,7,8,9,0), con igual probabilidad. Cuál es la información que lleva cada digito?

Aplicando la ecuación (2), encontramos que:

$$I = \log_2 n = \log_2 (1/p) = \log_2 10 = 3.32 \text{ bits}$$

Hay 3.32 bits de información en cada digito decimal. Esto significa que si los mensajes seleccionados por la fuente (números decimales) van a transmitirse por medio de pulsos binarios, se requieren 3.32 de estos pulsos para representar cada número decimal. En la práctica sabemos que es necesario utilizar 4 pulsos o digitos binarios para codificar cada digito decimal aunque quedan 6 ordenaciones sin utilizar. Esto es conocido como código BCD. Ver figura 3.

Binario	Decimal
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9

FIG. 3. CODIGO B C D .

c.4) MEDIDA DE LA INFORMACION DE MENSAJES CON PROBABILIDADES DESIGUALES DE SELECCION.
CASO DISCRETO.

En la sección anterior hemos estudiado el caso más sencillo de fuentes que tienen la misma probabilidad de selección para cada uno de los mensajes que pueden generar.

Veremos ahora la situación más general y real en la cual los distintos mensajes de la fuente tienen probabilidades diferentes de ser seleccionados para transmisión.

Nos referimos en nuestro estudio a una fuente de información discreta (fuente que genera un número finito de mensajes o símbolos) que puede producir K símbolos posibles. Supongamos que los diversos símbolos ocurren con probabilidades P(1), P(2), ...P(i) ...P(K). Estas son llamadas probabilidades A PRIORI y cada una de ellas es la probabilidad, en la fuente de información, de que un mensaje sea seleccionado para transmisión antes de que cualquier dato del mensaje llegue al receptor.

La información promedio asociada con la ocurrencia o selección de cualquiera de los símbolos se calcula como sigue:

- El contenido de información del i-ésimo mensaje producido por la fuente es:

$$I = \log_2 (1/P(i)) = -\log_2 P(i) \qquad (3)$$

- Si la fuente transmite una secuencia de "n" símbolos (n>>K), el i-ésimo símbolo ocurrirá nP(i) veces y la información total ganada de estas ocurrencias es:

$$\text{Información total de los símbolos i} = \begin{bmatrix} \text{nP(i) ocurrencias de} \\ \text{i en n símbolos} \\ \text{transmitidos} \end{bmatrix} \begin{bmatrix} \text{-log P(i) o} \\ \text{información de un} \\ \text{solo símbolo} \end{bmatrix}$$

$$= -nP(i) \log_2 P(i) \qquad (4)$$

- La información total generada por la fuente con la secuencia de "n" símbolos es la suma de la expresión anterior para todos los valores de i, o sea:

$$\text{Información total de la fuente} = -\sum_{i=1}^{K} nP(i) \log_2 P(i) \qquad (5)$$

- Y el promedio de información I por símbolo será:

$$I = \frac{-\sum_{i=1}^{K} nP(i)\log_2 P(i)}{n} = -\sum_{i=1}^{K} P(i)\log_2 P(i) \qquad (6)$$

$$H(i) = -\sum_{i=1}^{K} P(i)\log_2 P(i) \quad \frac{\text{bits}}{\text{símbolo}} \qquad (7)$$

Esta expresión tiene la misma forma que la empleada en termodinámica estadística para la entropia de sistemas. Por eso se le designa también como entropia de la fuente de información, representándosele por H(i). En resumen, H(i) o I, como es dada por la expresión 7, es la información promedio asociada con la selección de uno cualquiera de los mensajes de la fuente. También se le refiere, decíamos, como entropia de la fuente.

La base de logaritmos tomada en la ecuación (6) determina la unidad de medida de la información. Cuando la base es 2, la unidad es el bit, cuando es 10 la unidad es el Hartley y cuando es e la unidad es el nat. Dado que el sistema de comunicación de datos es binario, la base generalmente usada es 2 por la que la unidad de medida de la información más común es el bit.

Cuando todos los "K" símbolos generados por la fuente son igualmente probables, la ecuación (7) se transforma en:

$$I = -\sum_{K=1}^{K} \log_2 \frac{1}{K} = \log_2 K \qquad (8)$$

que es el resultado obtenido en la ecuación (2). Como se ve, la expresión (7) es enteramente general, abarcando tanto el caso en que los símbolos generados por la fuente tienen la misma probabilidad de ocurrencia como el caso en que esas probabilidades son desiguales.

Hemos visto que la cantidad de información de un mensaje es una función inversa de su probabilidad de ocurrencia o, visto en función del tiempo de transmisión, significa que es conveniente otorgar menor tiempo de transmisión a los mensajes con más alta probabilidad de ocurrencia que a los de poca probabilidad. En 1837 S.F.B. Morse siguió esta filosofía en su código al otorgar palabras de código más cortas a las letras del alfabeto inglés que se repiten más frecuentemente y palabras de código más largas a las letras menos frecuentes.

La probabilidad a que se hizo alusión, de que un mensaje determinado sea transmitido antes de que cualquier dato acerca del mensaje sea recibido es llamada PROBABILIDAD A PRIORI, mientras que la probabilidad en el receptor de que el mensaje recibido sea determinado mensaje transmitido es llamada PROBABILIDAD A POSTERIORI. La probabilidad a posteriori es importante cuando hay ruido presente en el canal de comunicación y la señal es perturbada por él.

Con el fin de ilustrar el concepto de probabilidad a posteriori considérese un sistema transmitiendo tres mensajes A, B y C. Debido al ruido, un 20% de las A transmitidas son interpretadas como B y un 20% como C. Un 50% de las B transmitidas son recibidas como A y un 50% como B. De las C transmitidas 66.7% son recibidas como C y 33.3% como B.

Las frecuencias relativas de transmisión (probabilidad a priori) de A, B y C son 0.5, 0.2 y 0.3 respectivamente. Así, si se transmiten N mensajes, habrá:

0.5N A's transmitidas
0.2N B's transmitidas
0.3N C's transmitidas.

Luego el número de A's recibidas será:

$$(0.6)(0.5N) + (0.5)(0.2N) = 0.4N$$

De estas 0.4N recepciones de A, 0.3N corresponden a transmisiones de A, 0.1N de B y 0 de C. Entonces, las probabilidades a posteriori, cuando A es recibida, de que el mensaje transmitido sea:

$$A \text{ es } \frac{0.3N}{0.4N} = 0.75$$

$$B \text{ es } \frac{0.1N}{0.4N} = 0.25$$

$$C \text{ es } \frac{0}{0.4N} = 0$$

La información se aumenta con el incremento de la probabilidad a posteriori y decrece con el incremento de la probabilidad a priori y se expresa por:

$$\text{Información recibida} = \log \left(\frac{\text{probabilidad a posteriori}}{\text{probabilidad a priori}} \right)$$

Entropia de eventos conjuntos.

La entropia conjunta H(ij) de la fuente o información promedio asociada con la producción de cualquier par de simbolos es:

$$H(ij) = - \sum_{i=1}^{n} \sum_{j=1}^{n} P(ij) \log P(ij)$$

donde n es el número de simbolos discretos que produce la fuente y P(ij) es la probabilidad conjunta de cualquier par de simbolos.

Entropia condicional.

La entropia condicional de la fuente H(j/i) o información promedio asociada con la producción de un simbolo, cuando el simbolo precedente es conocido, es:

$$H(j/i) = - \sum_{i=1}^{n} \sum_{j=1}^{n} P(ij) \log P(j/i)$$

La relación entre H(ij) y H(j/i) está dada por:

$$H(ij) = H(i) + H(i/j)$$

c.5) CANAL DISCRETO NO RUIDOSO.

La habilidad de un canal discreto para transmitir información se mide en términos de lo que es conocido como capacidad del canal C.

La capacidad del canal es la máxima cantidad de información (bits/seg) que el canal puede transmitir.

La capacidad del canal está dada por:

$$C = \log m / T$$

donde "m" es el número de combinaciones o secuencias posibles que se tienen, una de las cuales va a ser transmitida en un tiempo "T". Por ejemplo, supongamos un canal capaz de transmitir los digitos 1 y 0 a un régimen de n digitos/seg. En un tiempo "T" se transmitirán nT digitos los cuales dan 2^{nT} combinaciones o secuencias posibles. Entonces la capacidad para este canal será:

$$C = \log_2(2^{nT}) / T = n \text{ bits/seg}$$

El primer teorema fundamental de la teoria de información de Shannon establece que:

"Dada una fuente con entropia H (bits/simbolo) y un canal con capacidad C (bits/seg), es posible codificar la salida de la fuente en forma de poder transmitir sobre el canal a un régimen promedio de ((C/H) - X) simbolos/seg, donde X es arbitrariamente pequeña. No es posible transmitir a un régimen mayor que C/H".

Con el fin de obtener la máxima transferencia de información de la fuente al canal, es necesario que los mensajes sean codificados en forma que las propiedades estadisticas de la fuente se acoplen a las del canal.

En un canal no ruidoso no hay duda sobre si un mensaje recibido corresponde a un cierto mensaje transmitido. Asi, si el receptor recibe el mensaje A, la probabilidad de que ese mensaje corresponda a una A transmitida es 1, puesto que no hay ruido que pueda dar lugar a una mala interpretación o equivocación.

c.6) CANAL DISCRETO RUIDOSO.

Cuando no hay ruido presente sobre el canal de comunicación las señales transmitidas y recibidas son iguales, por lo tanto tienen la misma entropia. Este no es el caso cuando se genera ruido en el canal.

Vamos a considerar que:

$H(x)$ = entropia de la entrada al canal
$H(y)$ = entropia de la salida del canal
$H(n)$ = entropia del ruido
$H(x/y)$ = entropia de la señal de entrada cuando la senal de salida es conocida.

Los regimenes de entropia (velocidad de información en bits/seg) son $H`(x)$, $H`(y)$, $H`(n)$ y $H`(x/y)$ respectivamente.

Claude E. Shannon demostró que el régimen de información recibida R es:

$$R = H`(x) - H`(x/y) = H`(y) - H`(n)$$

es decir, que el régimen de información recibida es igual al régimen de información transmitida menos la incertidumbre o entropia que existe en el receptor sobre la señal transmitida después que la señal recibida ha sido inspeccionada. Shannon mostró que $H`(x/y)$ es el régimen al cual una información correctora extra debe ser proporcionada al receptor para vencer el efecto del ruido y permitir una interpretación correcta de la señal recibida.

c.7) CAPACIDAD DE UN CANAL.

La capacidad de un canal se define como el régimen máximo al cual el canal puede transmitir información. Supóngase que un canal puede transmitir pulsos de igual longitud de K diferentes valores a un régimen de n pulsos por segundo, entonces la información que lleva cada pulso es:

$$I = \log_2 (K) \text{ bits/pulso}$$

y el régimen de información por segundo (capacidad del canal) es:

$$C = n \log_2 (K) \text{ bits/seg.}$$

Si el régimen de información recibida R es maximizado para un canal teniendo ciertas caracteristicas (tipo de ruido presente, ancho de banda del canal, limitación sobre la forma de señal) este valor máximo de régimen de transmisión de información es igual a la capacidad del canal.

Suponiendo que el ruido en el canal es blanco gaussiano con potencia promedio N y valor medio cero, entonces, dado que la función de densidad de probabilidad del ruido es:

$$P(n) = (1/(\mathrm{SIGMA}*\mathrm{SQRT}(2*\mathrm{PI})))*(e**(-n^2/(2*(\mathrm{SIGMA}^2))))$$

la entropia del ruido en unidades naturales llega a ser:

$$H(v) = - \$_{-\sim}^{-} P(v)*\ln P(v)dv = \ln (\mathrm{SIGMA}*\mathrm{SQRT}(2*\mathrm{PI}*e))$$

donde $\$$ significa "integral".

Ahora bien, SIGMA = valor rms del ruido = SQRT (N), luego:

$$H(v) = \ln \mathrm{SQRT}\ (2*\mathrm{PI}*e*N)$$

y si el ancho de banda del canal es B Hertz:

$$H`(n) = 2*B*\ln (2*\mathrm{PI}*e*N) = B*\ln (2*\mathrm{PI}*e*N)\text{unidades naturales/seg}$$

Suponiendo que la señal transmitida sea de potencia limitada a S watts, entonces:

$$H`(y) = B*\ln (2*\mathrm{PI}*e*(S + N))$$

y la capacidad del canal estará dada por:

$$C = R\ \text{máxima} = (H`(y)-H`(n)) = B*\ln(1+S/N)\ \text{unidades naturales/seg}$$

$$C = \log_2 (1 + S/N)\ \text{bits/seg.}$$

Esta ecuación da un limite teórico a la capacidad o régimen de transmisión de información del canal ruidoso, limite teórico establecido por el ancho de banda del canal y el ruido generado.

Debe recalcarse que la capacidad del canal representa la máxima cantidad de información que puede ser transmitida por segundo por el canal. Para lograr este máximo, la información debe estar codificada en la forma más adecuada.

Si queremos mantener constante la capacidad del canal, el problema a resolver es encontrar la relación S/N.

La siguiente tabla es medida de acuerdo a la teoria de Shannon para una relación señal-ruido de 30 decibeles.

CANAL	RELACION S/N (dB)	ANCHO DE BANDA(Hz)	CAPACIDAD DEL CANAL(bits/s)
Linea telefónica	30	3000	30000
Radio AM	30	10000	99672
Radio FM	30	200000	1.993×10^{6}
Televisión comercial	30	6 millones	5.980×10^{7}
Sistema de microondas	30	20 millones	1.993×10^{8}
Sistema de cable coaxial	30	57 millones	5.681×10^{8}
Sistema de guia de onda	30	70 millones	6.977×10^{8}
Sistema de rayos Lásser	30	10 trillones	9.967×10^{19}

CAPACIDAD DE VARIOS CANALES DE COMUNICACION, REL. S/N CTE.

CAPITULO II.- CODIGOS PARA DETECCION Y CORRECCION DE ERRORES

a) DEFINICIONES Y OBJETIVOS DE LA CODIFICACION

a.1) SISTEMA DE TRANSMISION DE DATOS

La transmisiòn de datos es el "movimiento de informaciòn codificada por medio de sistemas de comunicaciòn elèctrica". En estos sistemas se entiende por datos, la informaciòn que ha sido tomada de documentos fuente, tales como òrdenes de pago, nòmina de personal, etc., o bien, de un medio de almacenamiento tal como cinta magnètica, tarjeta perforada, disco, etc.

Ha sido el crecimiento de los centros de procesamiento de datos, el factor principal para el desarrolo notable de los sistemas de transmisiòn de datos. En efecto, con la descentralizaciòn de las dependencias gubernamentales y de las grandes empresas, es necesario transportar los datos desde lugares remotos al computador central. Este problema fue resuelto en los primeros sistemas mediante el empleo del correo o utilizando mensajeros. Posteriormente con la introducciòn de los sistemas en tiempo real se hizo necesario un medio de comunicaciòn que permitiera a un usuario establecer una "conversaciòn" con la computadora utilizando un aparato terminal, es decir, se requerìa respuesta inmediata de la computadora a las consultas del usuario. El sistema de transmisiòn de datos resolviò este problema introduciendo la informaciòn de la terminal (codificada en forma digital) a un dispositivo (el modem) que permite la transmisiòn de informaciòn digital sobre canales telefònicos normales. En el lado de la computadora otro modem realiza la operaciòn inversa a la del modem del lado de la terminal, dando a la computadora la informaciòn digital transmitida. La misma tarea realizan los modems cuando el flujo de la informaciòn es de la computadora a la terminal.

Los sistemas de transmisiòn de datos comprenden tres tipos de enlaces:

Computadora - Computadora
Computadora - Terminal
Terminal - Terminal.

La terminal es un aparato de impresiòn tal como el teletipo o teleimpresor, un aparato de pantalla de tubo de rayos catòdicos con teclado, una lectora de tarjetas perforadas, una lectora de cinta de papel o magnètica, etc.

El esquema básico de un sistema de transmisión de datos del tipo Computadora - Terminal, es el siguiente:

```
 ______________      _________________      _________      ____________
|              | |  |                 |    |         |    |            |
| COMPUTADOR   | |  |  CONTROLADOR    |    |         |    | CANAL DE   |
|              |<-->|       DE        |<-->|  MODEM  |<-->| COMUNICA   |
|    CPU       | \  | COMUNICACIONES  |    |         |    | CIONES     |
|______________| /  |_________________|    |_________|    |____________|
                 |                                              |
                 v                                              |
             INTERFACE                                          |
                       ____________      _________              |
                      |            | |  |         |             |
                      |            | |  |         |             |
                      |  TERMINAL  |<------>| MODEM |<-----------
                      |            | \  |         |
                      |____________| /  |_________|
                                     |
                                     v
                                 INTERFACE
```

La terminal convierte los mensajes introducidos por el usario en una serie de combinaciones de pulsos y no pulsos de corriente eléctrica de acuerdo con un código determinado.

El modem convierte la señal digital proveniente de la terminal en una señal analógica apropiada para poder transmitirse en un canal de comunicación (usualmente una línea telefónica) diseñado especialmente para transmitir señales de tipo analógico.

El canal de comunicación es la vía por la cual se va a transmitir la señal y puede ser un par de hilos físicos, un cable coaxial, un canal de radio de alta frecuencia, un canal de microondas o una fibra óptica.

En el lado de la computadora, el modem convierte las señales de tipo analógico que recibe en señales digitales, que pasan al controlador de comunicaciones.

El controlador de comunicaciones puede ser desde un minicomputador programable con memoria hasta un aparato sin memoria que solo desarrola algunas funciones lógicas básicas como la de recibir los carácteres del modem en serie y transmitirlos en paralelo a la computadora.

La interface es el conjunto de conexiones que hacen posible el acoplamiento entre el modem y la terminal o entre el modem y el controlador de comunicaciones.

a.2) DEFINICION DE BAUD Y DE BIT/SEGUNDO.

La velocidad de señalización es el número máximo de pulsos por segundo que puede transmitir una máquina, sean o no de información y ya sean binarios, cuaternarios o de cualquier otro valor. El BAUD es la unidad de velocidad de señalización.

La velocidad de señalización se encuentra tomando el recíproco de la longitud, en segundos, del pulso más corto utilizado en la creación de un carácter.

Velocidad de información es el número máximo de pulsos binarios de información por segundo o BITS/segundo que puede transmitir una máquina. El BIT es la unidad de información y es la cantidad de información derivada del conocimiento de la ocurrencia de uno de dos eventos equiprobables, exhaustivos y exclusivos.

El número de bauds es igual al número de bits/segundo únicamente cuando todos los pulsos son de información, binarios y de la misma duración.

Si en cambio no todos los pulsos dl mensaje son de información (algunos son de paridad, de arranque y de parada, etc.) y si no son binarios (pueden ser pulsos que tomen cuatro o más valores diferentes), entonces el número de bauds es diferente al número de bits/seg.

Ejemplos.

Ejemplo 1.- Un computador genera información a razón de 12,000 bits/seg y el dispositivo que genera el código de protección agrega 3 bits de protección por cada 5 de código, entonces, la velocidad de información es 12,000 bits/seg y la velocidad de señalización es:

$$Vs = 12{,}000 + ((12{,}000/5)*3) = 19{,}200 \text{ bauds}$$

Ejemplo 2.- Un computador genera 12,000 bits/seg en formato NRZ y el codificador de linea lo convierte a 4 niveles; cada nivel representa 2 bits de la siguiente forma:

A	B	SALIDA
0	0	-15 V
0	1	- 5 V
1	0	+15 V
1	1	+ 5 V

Entonces, la velocidad de información es de 12,000 bits/seg y la velocidad de señalización es de 6,000 bauds ya que cada pulso representa 2 bits.

a.3) CARACTERISTICAS DE UN BUEN CODIGO.

1.- Transparencia.- debe poder codificar cualquier carácter o arreglo de carácteres, de manera que el receptor los reconozca sin dificultad.

2.- Capacidad de detección o corrección de errores.

3.- Igualar la entropia de la fuente a la capacidad del canal.

4.- Adecuar el ancho de banda de la señal al ancho de banda del canal.

5.- Enviar al receptor una suficiente información de base de tiempo para sincronizar su reloj.

Para satisfacer estas caracteristicas se han desarrollado:

a) Códigos de transmisión
b) Códigos detectores y correctores de error
c) Códigos de longitud óptima
d) Códigos de linea.

b) CLASIFICACION Y CARACTERISTICAS DE CADA UNO DE LOS TIPOS.

b.1) CODIGOS DE TRANSMISION.

Un código de transmisión de datos define una configuración de bits para cada carácter alfanumérico que debe ser transmitido. El número de bits necesario para representar cada carácter depende del número total de carácteres diferentes que la fuente puede generar.

Supóngase que los carácteres sean solo A y B, entonces basta un bit (un pulso binario) para representar a cada uno de ellos, o sea:

	bit	pulso binario
A	0	----
B	1	|¯|

Si los símbolos generados por la fuente son cuatro A, B, C y D, no basta un solo bit para representarlos sino que son necesarios 2. Con 2 bits se tienen 4 ordenaciones posibles (2**2 = 4). Cada uno de estos arreglos es posible utilizarlo para representar un carácter en la siguiente forma:

	bit	pulso binario
A	00	------
B	01	___|¯|
C	10	|¯|___
D	11	|¯¯¯|

Si son 8 los símbolos diferentes que puede generar la fuente, se deben utilizar 3 bits para representar cada símbolo. Con 3 bits se logran 2**3 = 8 ordenaciones diferentes, así:

	bit	pulso binario		bit	pulso binario
A	000	---------	E	100	|¯|_____
B	001	_____|¯|	F	101	|¯|_|¯|
C	010	__|¯|__	G	110	|¯¯¯|__
D	011	__|¯¯¯|	H	111	|¯¯¯¯¯|

En general, con m bits se logran 2**m ordenaciones por lo que se pueden representar otros tantos símbolos o carácteres diferentes.

Si usamos la expresión 2**m = n ordenaciones y tomamos el logaritmo de base 2 en ambos lados de la ecuación, se obtiene:

$$\log_2 n = m$$

lo que significa que un alfabeto consistiendo de n carácteres diferentes requiere

$$\log_2 n = m \text{ bits}$$

para codificarlo en forma binaria.

Los códigos más usuales para hacer adecuados los símbolos del lenguaje para su transmisión por medio de un sistema de comunicación son:

- El Código Baudot
- El Código ASCII.

De aplicación más restringida para transmisión están los códigos BCD y EBCDIC.

b.1.1) CODIGO DE MORSE

Este código no se usa en la transmisión de datos pero es interesante conocerlo por su importancia histórica.

Este código fue el primer código de transmisión de datos y se elaboró alrededor del año de 1838. En él ya estaban aplicadas algunas de las ideas fundamentales de la teoría moderna de la codificación, como es la referente a igualar la entropía de la fuente a la entropía del canal, haciendo que los carácteres más usuales (las vocales) tengan códigos más cortos.

El Código Morse siempre se ha transmitido a mano (2 a 3 carácteres por segundo) y cuando se quiso diseñar una máquina automática para codificar y decodificar se encontró que es más conveniente que el número de bits por carácter sea uniforme, lo que no es posible con los pulsos del Código Morse.

De esta manera, el Morse se ha quedado para transmisión manual y se han elaborado códigos uniformes para codificación y transmisión automática.

b.1.2) CODIGO BAUDOT

El Código Internacional No. 2 de 5 unidades del CCITT, comunmente conocido como código Baudot, es una variante del Código Baudot usado ampliamente desde hace más de 45 años.

Es también conocido como código telegráfico de 5 unidades y prácticamente todos los sistemas telegráficos del mundo operando a 50 y 75 bauds utilizan este código o alguna variante de él.

El Código Baudot representa cada carácter con un conjunto de cinco bits. Sin embargo, con 5 bits solo se tienen 2**5 = 32 ordenaciones posibles que resultan insuficientes para codificar las 26 letras del alfabeto (latino), los 10 números decimales, los signos de puntuación y matemáticos y las figuras.

Por esta razón, para ampliar el número de simbolos que se pueden representar con este código, se utilizan dos carácteres para indicar:

- Cambio a letras
- Cambio a figuras.

El receptor interpreta todos los bits que van después de un carácter de cambio a letras como letras y todos los que siguen a un carácter de cambio a figuras como figuras, ya sean números, signos, etc.

Asi, cada secuencia de 5 bits en el Código Baudot tiene dos significados dependiendo de por cuál carácter de CAMBIO hayan sido precedidos. Esto se aclara con la siguiente explicación.

Con 5 bits se pueden obtener 2**5 ordenaciones de las que descontamos la (00000) que no tiene significado, quedando 2**5 - 1 = 31 posibles ordenaciones. De esas descontamos 5 para los movimientos del carro:

1.- Subir carro (letras)
2.- Bajar carro (figuras)
3.- Retroceder carro
4.- Cambio de renglón
5.- Espacio.

Las 26 restantes posibilidades, como ya se dijo tienen doble significado, lo que nos da la posibilidad de codificar 26x2 = 52 carácteres alfanuméricos.

El alfabeto latino tiene 27 letras a las que sumamos 10 digitos, por lo que nos sobran 15 ordenaciones para codificar signos de puntuación y algunos otros carácteres. Casualmente hay 26 letras esenciales en el inglés (no existe la Ñ) por lo que cada arreglo de 5 bits equivale a una letra o a un signo.

En la transmisión de información empleando el código BAUDOT, la sincronía entre el transmisor y el receptor se establece con cada carácter. Así, cada carácter codificado de acuerdo al código, va precedido por un pulso binario de arranque (generalmente un cero) y seguido por un pulso de parada (el cual siempre es un uno).

- La duración del pulso de arranque es igual a la de los pulsos de información.

- La duración del pulso de parada es igual a 1, 1.42, 1.5 ó 2 veces la duración de los pulsos de información. Es muy común que el pulso de parada dure 1.42 veces los de información.

El código Baudot es el código telegráfico por excelencia. Todos los sistemas de Telegrafía, Telex y algunas terminales de computadora emplean este código. Sus limitaciones principales son:

- El limitado número de ordenaciones que es permitido para representar carácteres (52).

- No permite bits de chequeo de paridad.

Para complementar esta explicación, diremos que por costumbre los unos se llaman "marcas" y los ceros "espacios".

C O D I G O   B A U D O T

LETRA		FIGURA
A	10000	1
B	00110	8
C	10110	9
D	11110	0
E	01000	2
F	01110	NA
G	01010	7
H	11010	+
I	01100	NA
J	10010	6
K	10011	(
L	11011	=
M	01011	)
N	01111	NA
O	11100	5
P	11111	%
Q	10111	/
R	00111	-
S	00101	.
T	10101	NA
U	10100	4
V	11101	`
W	01101	?
X	01001	,
Y	00100	3
Z	11001	:
LS	00001	LS
FS	00010	FS
CR	11000	CR
LF	10001	LF
ER	00011	ER
NA	00000	NA

LS = Letter Shift
LF = Line Feed
FS = Figure Shift
CR = Carriage Return
ER = Error
NA = Not Asigned
Space = LS o FS

b.1.3) CODIGO ASCII

Por la extensión de las máquinas de informática de los EE.UU. en el mercado mundial, el código ISO (International Standard Organization) de 7 bits es más comunmente conocido como código ASCII (American Standard Code for Information Interchange).

El código ASCII, que se empezó a usar ampliamente en 1967, surgió como una respuesta a la necesidad de contar con un código que sirviera para el intercambio de información entre sistemas de procesamiento de información, sistemas de comunicación y equipo periférico asociado. El Código Baudot no es útil para este propósito por el número limitado de combinaciones que permite y por la falta de una estructura lógica que facilite la comunicación con el computador.

El código ASCII o alguna variante de él, es universalmente empleado por todos los equipos periféricos para la comunicación con computadoras. Emplea 7 bits para representar cada símbolo alfanumérico y generalmente se agrega un octavo bit para comprobar paridad con el fin de detectar errores.

El conjunto de 7 bits permiten 2**7 = 128 arreglos posibles, los cuales pueden ser agrupados en 4 subconjuntos de carácteres de la siguiente forma:

1. Control
2. Números y signos
3. Letras mayúsculas
4. Letras minúsculas.

El subconjunto de carácteres de control está formado por 32 elementos que pueden ser agrupados en las 4 categorias siguientes:

a) Separadores de información
b) Controles de aparato
c) Realizadores de formato
d) Controles de comunicación.

Los carácteres separadores de información son utilizados para separar y codificar la información en un sentido lógico. Dichos carácteres son los siguientes:

FS - Separador de archivo
GS - Separador de grupo
RS - Separador de registro
US - Separador de mitad.

Los carácteres de control de aparato son DC1, DC2, DC3 y DC4 y tienen por objeto el control de aparatos asociados con el procesamiento de datos o sistemas de comunicación, especialmente para aparatos de switcheo ON y OFF.

Los carácteres realizadores de formato controlan el arreglo o posicionamiento de la información en aparatos de impresión o de pantalla. Entre ellos están:

LF (Line Feed) que controla el movimiento de la posición impresora a la siguiente linea de impresión.
CR (Carriage Return) que controla el movimiento de la posición impresora a la primera posición de la misma linea de impresión.
BS (Back Space) que controla el movimiento de la posición impresora en espacio de impresión de regreso sobre la misma linea impresa.

Los carácteres de control de comunicación tienen por objeto controlar o facilitar la transmisión de información sobre el sistema de comunicación. Algunos de estos carácteres son:

SOH (Start of Heading). Se usa al comienzo de una secuencia de carácteres que constituyen una dirección sensible a la máquina o un enrutamiento de información.
ETX (End of Text). Se emplea para terminar una secuencia de carácteres comenzados con STX.
STX (Start of Text). Precede a una secuencia de carácteres que es tratada como una entidad.
EOT (End of Transmission). Se utiliza para indicar la conclusión de una transmisión que puede contener uno o más textos y cualquier encabezamiento.
ENQ (Enquiry). Es usado como una petición de una respuesta desde una estación remota.
ACK (Acknowledge). Es un carácter que envia el receptor como una respuesta afirmativa a un transmisor.
NAK (Not Acknowledge). Es enviado por el receptor como una respuesta negativa a un transmisor.
SYN Se utiliza por un sistema de transmisión sincrona en la ausencia de cualquier otro carácter para proporcionar una señal de la cual el sincronismo puede ser logrado.
ETB (End of Transmission Block). Indica el fin de un bloque de datos para propósitos de comunicación.

A continuación se ilustran algunas aplicaciones de los carácteres de control de comunicaciones.

En una transmisión SINCRONA la secuencia para la transmisión de un bloque de carácteres es:

--------------------> BLOCK PARITY CHARACTER

S	S	S		S		E	E	B
Y	Y	O	ENCABEZAMIENTO	T	TEXTO	T	O	P
N	N	H		X		X	T	C

<---------

S	S	A
Y	Y	C
N	N	K

En una transmisión ASINCRONA la secuencia de carácteres para la transmisión de un texto es:

C P U

(El computador pregunta a una terminal si tiene mensaje que transmitirle y si lo tiene se lo envie).

-------->1

S	E	E	A
T	O	O	D
X	T	T	R

DIRECCION DE
LA TERMINAL

2 <----------

B	E		A	S
P	T	TEXTO	D	T
C	X		R	X

T E R M I N A L
(Si la terminal tiene mensaje que transmitir el computador contesta enviando el texto).

--->3

A
C
K

C P U
(Contesta a la terminal que el mensaje se recibió sin error).

En la transmisión asincrona cada carácter del código ASCII es precedido por un bit de arranque y seguido por 1 ó 2 bits de parada.

Entre las caracteristicas principales que han hecho adecuado al código ASCII para el intercambio de información de computadoras y terminales, podemos citar las siguientes:

a) Es un código computable porque los carácteres están estructurados en una forma secuencial binaria apropiada para las computadoras digitales.

b) El conjunto de carácteres escogidos contiene los carácteres gráficos de los principales lenguajes de programación.

c) Suponiendo que los 128 carácteres no satisfagan los requerimientos de aplicación, el código ASCII preveé funciones de escape para permitir la expansión del código a más de 128 carácteres.

Las funciones de escape son los carácteres de control siguientes:

DLE - Data Link Escape
ESC - Escape
SO - Shift Out
SI - Shift In

d) Algunas indicaciones del código ASCII facilitan su operación por la computadora o terminal y su manejo por los usuarios. Algunas de estas indicaciones son:

1) Cuando el 7o. nivel es un 1, indica una letra mayúscula o uno de 6 símbolos especiales.
2) Cuando el 6o. nivel es 1, indica un número, uno de los 21 símbolos gráficos o un espacio.
3) Cuando el 6o. y el 7o. niveles son 0, indica que son carácteres de control.
4) Cuando el 6o. y el 7o. niveles son 1, indica letras minúsculas, símbolos especiales o la función DELETE.

En cuanto a la comprobación de paridad que se realiza con el 8o. bit de cada carácter, se recomienda usar:

- Paridad IMPAR para transmisión síncrona
- Paridad PAR para transmisión asíncrona.

b_0,b_1,b_2,b_3 \ b_4,b_5,b_6	000	100	010	110	001	101	011	111
0000	NUL	DLE	SPACE	0	@	P		p
1000	SOH	DC1	!	1	A	Q	a	q
0100	STX	DC2	"	2	B	R	b	r
1100	ETX	DC3	#	3	C	S	c	s
0010	EOT	DC4	$	4	D	T	d	t
1010	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
1110	BEL	ETB	'	7	G	W	g	w
0001	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
0101	LF	SUB	*	:	J	Z	j	z
1101	VT	ESC	+	;	K	[	k	{
0011	FF	FS	,	<	L	\	l	¦
1011	CR	GS	-	=	M	]	m	}
0111	SO	RS	.	>	N	∧	n	~
1111	SI	US	/	?	O	_	o	DEL

NUL =All zeros
SOH =Start of heading
STX =Start of text
ETX =End of text
EOT =End of transmission
ENQ =Enquiry
ACK =Acknowledgment
BEL =Bell or attention signal
BS =Back space
HT =Horizontal tabulation
LF =Line feed
VT =Vertical tabulation
FF =Form feed
CR =Carriage return
SO =Shift out
SI =Shift in
DLE =Data link escape

DC1 =Device control 1
DC2 =Device control 2
DC3 =Device control 3
DC4 =Device control 4
NAK =Negative acknowledgment
SYN =Synchronous / idle
ETB =End of transmitted block
CAN =Cancel (error in data)
EM =End of medium
SUB =Start of special sequence
ESC =Escape
FS =Information file separator
GS =Information group separator
RS =Information record separator
US =Information unit separator
DEL =Delete

FIGURA 4. CODIGO ASCII STANDARD.

b.1.4) CODIGO EBCDIC

El otro código binario, además del ASCII, desarrollado para representar símbolos alfabéticos, numéricos y especiales en sistemas digitales, es el código EBCDIC (Extended Binary Coded Decimal Interchange Code).

El código EBCDIC utiliza 8 bits para representar un símbolo, mientras que el código ASCII se presenta en dos formas: ASCII standard de 7 bits y ASCII extendido o modificado de 8 bits. En este último, el bit más significativo es 1. Esto indica que los siguientes 7 bits forman una palabra del código. El ASCII standard se puede obtener del extendido con solo eliminar en la palabra el bit más significativo.

El código EBCDIC se desarrolló en IBM para usarlo en los sistemas 360 y 370 exclusivamente, aunque ahora se utiliza en otras computadoras a fin de ser compatibles al equipo IBM.

Existen 2**8 = 256 combinaciones para el EBCDIC, algunas de las cuales, como se puede ver en la figura 5, no se han asignado aún. Se pueden ir asignando combinaciones a medida que el desarrollo de la tecnología en computación las requiera.

b_3, b_2, b_1, b_0	Segundo dígito Hexadecimal	00				01				10				11				b_7, b_6
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	b_5, b_4
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Primer Dígito Hexadecimal
0000	0	NUL	DLE	DS		SP	&	-									Ø	
0001	1	SOH	DC1	SOS				/		a	j			A	J		1	
0010	2	STX	DC2	FS	SYN					b	k	s		B	K	S	2	
0011	3	EXT	TM							c	l	t		C	L	T	3	
0100	4	PF	RES	BYP	PN					d	m	u		D	M	U	4	
0101	5	HT	NL	LF	RS					e	n	v		E	N	V	5	
0110	6	LC	BS	ETB	UC					f	o	w		F	O	W	6	
0111	7	DEL	IL	ESC	EOT					g	p	x		G	P	X	7	
1000	8		CAN							h	q	y		H	Q	Y	8	
1001	9		EM							i	r	z		I	R	Z	9	
1010	A	SMM	CC	SM			!		:									
1011	B	VT	CU1	CU2	CU3	.	$	,	#									
1100	C	FF	IFS		DC4	<	*	%	@									
1101	D	CR	IGS	ENQ	NAK	(	)	-	'									
1110	E	SO	IRS	ACK		+	;	>	=									
1111	F	SI	IUS	BEL	SUB	|	¬	?	"									

FIGURA 5. CODIGO EBCDIC.

ACK =Acknowledge
BEL =Bell
BS =Backspace
BYP =Bypass
CAN =Cancel
CC =Cursor control
CR =Carriage return
CU1 =Customer use 1
CU2 =Customer use 2
CU3 =Customer use 3
DC1 =Device control 1
DC2 =Device control 2
DC4 =Device control 4
DEL =Delete
DLE =Data link escape
DS =Digit select
EM =End of medium
ENQ =Enquiry
EOT =End of transmission
ESC =Escape
ETB =End of transmission block
ETX =End of text
FF =Form feed
FS =Field separator
HT =Horizontal tab
IFS =Interchange file separator

IGS =Interchange group separator	SM =Set mode
IL =Idle	SMM =Start of manual message
IRS =Interchange record separator	SO =Shift out
IUS =Interchange unit separator	SOH =Start of heading
LC =Lower case	SOS =Start of significance
LF =Line feed	SP =Space
NAK =Negative acknowledge	STX =Start of text
NL =New line	SUB =Substitute
NUL =Null	SYN =Synchronous Idle
PF =Punch off	TM =Tape mark
PN =Punch on	UC =Upper Case
RES =Restore	VT =Vertical Tab
RS =Reader stop	
SI =Shift in	

CARACTERES ESPECIALES

| =OR lógico
¬ =NOT lógico

b.2) CODIGOS DETECTORES DE ERROR. DEFINICIONES.

Antes de entrar en detalles, definiremos los conceptos de "corrección" y "detección" de errores.

Se dice que un código detecta un error cuando precisa la existencia de éste en un mensaje recibido, pero no sabe cuál es el bit erróneo. Su corrección solo se hará solicitando retransmisión del mensaje.

Se dice que un código es corrector de errores porque puede precisar el bit erróneo. Una vez precisado el bit equivocado, basta con cambiarlo por el bit contrario y ya está hecha la corrección. No es necesario repetir la transmisión del mensaje para corregir.

Ahora hablaremos un poco sobre el importante concepto de redundancia. REDUNDANCIA es la transmisión de más información de la estrictamente necesaria para la comunicación de un mensaje. La redundancia puede estar en la fuente de información, en el código, en la señal transmitida, en el canal o en el sistema de comunicación.

Para ilustrar la idea, consideremos que los mensajes que se van a transmitir son SI y NO. La mínima información a transmitir a un receptor para darle a conocer cualquiera de los dos mensajes es 1 bit. En esta forma los mensajes son codificados como:

	bit	pulso binario
SI	1	
NO	0	

Al transmitir los dos mensajes sobre un canal ruidoso (en la práctica todos lo son) por medio de un solo bit, el ruido puede ocasionar que el espacio sea recibido como marca o viceversa, haciendo que un NO transmitido sea interpretado como un SI por el receptor o un SI como un NO.

Si en lugar de transmitir cada uno de los dos mensajes por medio de un solo bit se emplean 2 como se indica a continuación:

	bit	pulso binario
SI	1 1	
NO	0 0	

un solo bit erróneo en la señal recibida no es suficiente para convertir un SI en un NO o viceversa, ya que los mensajes 1 0 y 0 1 no están permitidos dentro del lenguaje del transmisor y receptor, por lo que éste se da cuenta que hubo un error al recibir una de dichas secuencias. Si todavía más, los mensajes SI y NO son codificados como:

	bit	pulso binario
SI	1 1 1	
NO	0 0 0	

entonces, si hay un solo bit errôneo, el receptor no solo se da cuenta de que hubo un error sino que lo ubica, pudiendo corregirlo. Asi, si la secuencia de bits que detecta el receptor es alguna de las siguientes, puesto que hay solo un bit errôneo, corresponden al mensaje 0 0 0:

0 0 1,
0 1 0,
1 0 0.

Si, en cambio, recibe las secuencias 0 1 1 , 1 0 1 ó 1 1 0, éstas corresponderán al mensaje 1 1 1.

Como se ve, el precio que hay que pagar para lograr vencer los efectos del ruido, es el de requerir un mayor tiempo de transmisión que el minimo necesario para cada mensaje.

Los bits adicionales introducidos en cada mensaje fuera del minimo requerido para representarlo, con el fin de permitir al receptor detectar y/o corregir, se denominan comunmente bits redundantes o información redundante.

Todos los lenguajes del hombre son redundantes, unos más que otros, pero en general se considera que poseen una redundancia del 80%. Esto significa que solo una quinta parte de los simbolos de una página son realmente necesarios para transmitir la información. Sin embargo, como en la transmisión de datos, aqui también la redundancia es necesaria para entender el significado de los mensajes, aunque a veces se peque de exceso de ella.

REDUNDANCIA Y EFICIENCIA.

La eficiencia E de un código se define como:

$$E = \frac{\text{Bits de información (sin considerar los de redundancia)}}{\text{Total de bits en la representación del carácter.}}$$

Mientras que la redundancia R se expresa por:

$$R = 1 - E.$$

La eficiencia y la redundancia entran en conflicto en un sistema de comunicación. Cuanto mayor redundancia se introduzca, menor será la eficiencia, pero cuanta más redundancia sea recibida, menor será la probabilidad de error debido al ruido. Debemos, por lo tanto, encontrar un código que proporcione la máxima transmisión libre de error con el minimo de redundancia.

El código BAUDOT, por ejemplo, tiene una eficiencia de 100% y una redundancia 0, por eso la protección contra el ruido es nula. Eso hace que un bit erróneo en un carácter se convierta en otro carácter válido dentro del código sin que el receptor tenga forma de darse cuenta que hubo un error. Esto se ilustra en la siguiente secuencia:

```
Carácter CR transmitido 1 1 0 0 0        |‾‾‾‾‾|______

más ruido en el canal                    ..........^...
                                          ......
se convierte en la senal recibida        /      \../\..

que es detectada como                    |‾‾‾‾‾|_|‾|_

y que corresponde al carácter H           1  1  0  1  0.
```

La eficiencia del código ASCII es de 87.5% y su redundancia es de 12.5% , ya que cada carácter contiene 7 bits de información más un bit de paridad o redundante.

Como se puede observar, la eficiencia en el código ASCII es menor que en el código BAUDOT, aunque la redundancia es menor en éste último. Sin embargo, tiene la ventaja de que un bit erróneo en el carácter recibido no lo convierte en otro carácter válido del código y el receptor se da cuenta de que hubo un error en la transmisión.

DISTANCIA DE HAMMING.

En los sistemas que emplean códigos detectores de error, la corrección del mismo se hace por la repetición en el transmisor del bloque de bits por indicación previa del receptor de que ha habido error. Esta es la forma usada por la mayoría de los sistemas de transmisión de datos para corregir el error. Otro método para corregir el error una vez detectado, es agregar suficiente redundancia a los mensajes transmitidos para permitir que el propio receptor haga la tarea de corrección sin recurrir a la repetición del mensaje.

La Distancia de Hamming entre dos caracteres codificados es el número de bits en el cual ellos difieren, es decir, es el número de bits que deben cambiar en un carácter para dar por resultado el otro carácter. Por ejemplo, la distancia de Hamming entre los caracteres 10010 y 01010 es 2, puesto que éste es el número de bits en el cual ellos difieren.

La Distancia Mínima de Hamming de un código es el número mínimo de bits que deben cambiar en un carácter codificado para dar por resultado otro carácter válido del código.

La distancia de Hamming del código Baudot es 1, ya que basta un cambio de un bit del carácter para dar por resultado otro carácter válido. En el código ASCII (incluyendo el octavo bit de paridad) la distancia mínima de Hamming es 2.

La cantidad de bits erróneos que se pueden detectar y corregir y que un código permite realizar, es función de la llamada distancia de Hamming. La relación es la siguiente:

$$M - 1 = C + D$$

donde: M = distancia de Hamming
C = numero de bits erróneos que pueden corregirse.
D = número de bits erróneos que pueden detectarse.

C no puede ser mayor que D, puesto que para corregir un bit de error es necesario haberlo detectado antes. En la siguiente figura se muestran los valores de M de uno hasta cinco y los valores correspondientes de C y D.

Distancia Minima de Hamming	Propiedades de combatir el error	
M	C	D
1	0	0
2	0	1
3	0	2
	1	1
4	0	3
	1	2
5	0	4
	1	3
	2	2

De la figura anterior se puede observar que con una distancia mínima de Hamming de 1, no es posible detectar ni corregir ningún bit erróneo (como sucede en el código Baudot). Con una distancia minima de 2, es posible detectar un error pero no corregirlo (sin retransmisión de mensaje). Ejemplo de esto son los códigos de paridad. Con una distancia minima de 3, se pueden detectar dos errores o en forma alterna detectar un error y corregirlo.

Los códigos de Hamming, de Hagelberger y de Bose-Chaudhuri son algunos tipos de códigos correctores de error.

CODIFICACION PARA IGUALAR LA ENTROPIA DE LA FUENTE A LA CAPACIDAD DEL CANAL.

Suele decirse que un código es a un canal de comunicación lo que un transformador es a una transmisión eléctrica, es decir, un medio de mejorar su eficacia operativa. Asi como se usa un transformador adecuado a la resistencia de carga del transmisor para obtener un máximo de transferencia de potencia desde un generador hasta el receptor, también se necesita un código adecuado a la estructura estadistica del lenguaje usado en la transmisión.

El canal no puede transmitir información a su capacidad máxima si la fuente de información es redundante. Shannon probó que es posible codificar la salida de la fuente redundante en forma tal que el régimen de transmisión de información de la fuente se aproxime a la capacidad del canal. El principio básico de la codificación para igualar la fuente al canal, es que los mensajes improbables son representados por señales más grandes que los mensajes que ocurren más frecuentemente. La codificación es más eficiente, es decir, el régimen de transmisión de información tiende a ser igual a la capacidad del canal, cuando las longitudes de las señales son proporcionales al logaritmo de base 2 del inverso de las probabilidades de los mensajes correspondientes.

b.3) TIPOS DE CODIGOS DETECTORES DE ERROR.

En cualquier canal de comunicación hay factores tales como el ruido y la distorsión que alteran la senal de información transmitida. Cuando la señal es voz, la alteración no es tan dañina por la naturaleza redundante de una comunicación humana. No ocurre lo mismo si la comunicación es de datos y los extremos transmisor y receptor son máquinas en lugar de seres humanos. En este caso la protección de los datos transmitidos de posibles alteraciones o distorsiones en el canal de comunicación es de gran importancia, recurriéndose con este fin a la utilización de códigos detectores y correctores de error.

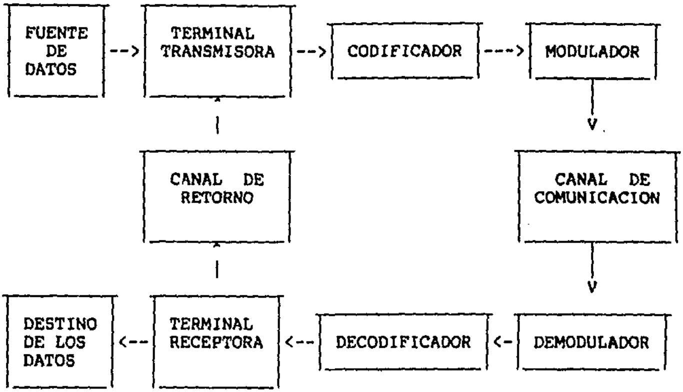
Los códigos detectores de error más usuales son:

- Paridad par e impar,
- M fuera de N,
- Paridad vertical y horizontal,
- Ciclico o polinomial.

La manera de corregir los errores una vez detectados, puede realizarse solicitando al transmisor la repetición del mensaje erróneo o efectuándose la corrección en el propio receptor sin intervención del transmisor. Este último modelo requiere de técnicas de codificación más sofisticadas, así como la transmisión de tantos o más bits de control de errores como de bits de datos. Esta es una de las razones por las cuales es menos utilizado en la práctica en nuestros dias.

El método de Retransmisión del mensaje erróneo es, en la actualidad, el más usado en la corrección de errores. La disponibilidad de un canal de retorno del receptor al transmisor para indicarle a éste que se ha recibido un error y solicitarle la retransmisión del bloque de datos erróneo, al mismo tiempo que se transmite el mensaje normal, aumenta la eficiencia de transmisión. La solicitud de retransmisión puede realizarse también a través del canal principal sin necesidad del canal de retorno, interrumpiéndose para ello la transmisión de datos de transmisor a receptor, lo que disminuye la eficiencia de la transmisión.

La siguiente figura es un diagrama a bloques de un sistema de transmisión de datos con detección de errores y corrección por el método de retransmisión:



La terminal transmisora estructura la secuencia de datos en bloques, los almacena al mismo tiempo que los transmite hasta tener la evidencia de que fueron recibidos correctamente por el receptor o si hay notificación de error, los retransmite; además, agrega bits de control y de sincronización y controla normalmente el circuito de transmisión.

Los bloques de datos pasan luego al codificador donde se adicionan los bits de paridad que son determinados en base a una ley de los bits de datos. Los bits de paridad permiten al receptor conocer si ha habido o no error en la transmisión. Normalmente el codificador es parte de la terminal transmisora pero como no es este siempre el caso, se ha representado por un bloque diferente o independiente.

Si cada bloque de datos contiene k bits de información y n-k bits de paridad, se dice que el código es un código de bloque (n, k) con una eficiencia $R = k/n$.

El bloque codificado de datos es adaptado por el modulador a la forma analógica de modo que pueda transmitirse adecuadamente sobre el canal de comunicación.

En el extremo receptor el bloque demodulado es entregado a la terminal receptora a través del decodificador, el cual realiza las operaciones lógicas para determinar si hay error y también indicar cuál es el bit equivocado. Una vez revisado y corregido, el bloque es entregado al destino de los datos y la terminal receptora notifica a la terminal transmisora a través del canal de retorno o del canal principal (si no existe aquél o si no se utiliza), que el bloque ha sido correctamente recibido; esta notificación se hace por medio del carácter ACK (Acknowledge o reconocimiento positivo, sin error). Si hay discrepancia quiere decir que hubo error en la transmisión, entonces la terminal transmisora es notificada con un carácter NAK (Not Acknowledge) y el bloque de datos es retransmitido. Cada bloque de datos transmitido es almacenado en la terminal transmisora hasta que un carácter ACK o un NAK es recibido.

b.3.1) CODIGO DETECTOR DE ERROR DE PARIDAD PAR E IMPAR.

En este método se agrega un bit llamado de paridad a cada grupo de datos para hacer que el número de bits 1 ó bits 0 sea par o impar. Si, por ejemplo, se acuerda paridad par sobre los unos, con cada juego de datos transmitidos se transmite al final un bit que puede ser 1 ó 0, dependiendo de que el número de unos del carácter transmitido sea par o impar. Si este número es impar, el bit que se transmite al último (bit de paridad) será 1. En cambio, si el número de unos en el carácter es par, el bit de paridad será 0. De este modo el número de bits 1 en cada carácter codificado, incluyendo al bit de paridad, es par, lo cual permite al receptor determinar si ha habido error en la transmisión, ya que sabe que cada carácter debe contener siempre un número par de unos.

Como ejemplo ilustrativo consideremos 5 caracteres numéricos representados por el código de transmisión BCD 8421. Si se usa paridad par sobre los unos, el bit de paridad que se transmite al final de cada carácter será el indicado en la siguiente figura:

Digito decimal	Representación BCD 8421 incluyendo la paridad
1	0 0 0 1 1
2	0 0 1 0 1
3	0 0 1 1 0
5	0 1 0 1 0
7	0 1 1 1 1

bits de paridad.

Nótese que si hay un número par de bits erróneos en cada carácter, el receptor no podrá detectarlos. Con este código solo es posible detectar un número impar de errores.

Sin embargo, la probabilidad de que haya 2 ó más errores en un carácter es pequeña, como lo prueba el siguiente cálculo.

Sean: n = número de bits en un carácter transmitido.

P = probabilidad de que un bit sea recibido erróneamente.

1-P = probabilidad de que un bit sea recibido correctamente.

La probabilidad de que el primer bit del carácter sea erróneo y todos los demás correctos, es:

$$P(1-P)_1\ (1-P)_2\ \ldots\ (1-P)_{n-1} = P(1-P)^{**}(n-1)$$

y la probabilidad de que el segundo sea erróneo y todos los demás correctos será, igualmente:

$$P(1-P)_1\ (1-P)_2\ \ldots\ (1-P)_{n-1} = P(1-P)^{**}(n-1)$$

Por tanto, la probabilidad de que haya un bit erróneo cualquiera en los n bits transmitidos, será la probabilidad de que el primero sea el erróneo, más la probabilidad de que el segundo sea el erróneo, más la probabilidad de que sea el tercero, etc., es decir, esta probabilidad será:

$$nP(1-P)^{**}(n-1)$$

En general, la probabilidad de que haya r errores en n bits transmitidos está dada por:

$$\frac{n!}{r!\,(n-r)!}\, P^{r}\,(1-P)^{n-r}$$

La siguiente tabla muestra las diferentes probabilidades de error al transmitir 16 bits y suponiendo que la probabilidad de que un bit transmitido sea recibido erróneamente es de 0.5 .

r = No. de bits errôneos	Probabilidad de ocurrencia
0	1.5259×10^{-5}
1	2.4414×10^{-4}
2	1.8311×10^{-3}
3	8.5450×10^{-3}
4	2.7771×10^{-2}
5	6.6650×10^{-2}
6	1.2219×10^{-1}
7	1.7456×10^{-1}
8	1.9638×10^{-1}

Las ventajas de los códigos de paridad par e impar son:

a) Código simple y de fácil implementación.

b) No está restringido el número de combinaciones posibles.

c) Se agrega solo un bit de redundancia en cada carácter.

La gran desventaja de este tipo de códigos es que un número par de bits errôneos no es detectado.

b.3.2) CODIGO M FUERA DE N

En este tipo de códigos detectores de error hay una relación constante entre el número de bits 1 y el número de bits 0 que se combinan para codificar un carácter. Esta relación constante entre ceros y unos permite al receptor determinar si uno o más bits han sido alterados por el ruido u otros factores en el canal de transmisión.

El número de combinaciones posibles Q (número de carácteres que se pueden representar) de un código M fuera de N está dado por la expresión:

$$Q = \frac{N!}{M!\ (N-M)!}$$

donde:

N = número total de bits de cada carácter

M = número de bits que son 1.

Los códigos M fuera de N más usuales son:

- 2 fuera de 5.
- 3 fuera de 7.
- 4 fuera de 8.

El código 2 fuera de 5 representa cada carácter por una combinación de dos bits 1 y tres bits 0. El número de combinaciones posibles en este caso se obtiene aplicando la ecuación anterior y da:

$$Q = \frac{5!}{2!\,(5-2)!} = \frac{1*2*3*4*5}{(1*2)(1*2*3)} = 10$$

Como se ve en la siguiente figura, cada combinación difiere de otra en dos dígitos binarios al menos (la distancia de Hamming es 2). En esta forma un bit erróneo en un carácter lo convierte en otro carácter no válido dentro del código, lo cual permite al receptor detectar el error.

1 1 0 0 0

1 0 1 0 0

1 0 0 1 0

1 0 0 0 1

0 1 1 0 0

0 1 0 1 0

0 1 0 0 1

0 0 1 1 0

0 0 1 0 1

0 0 0 1 1

La limitación del código 2 fuera de 5 es el poco número de combinaciones que permite, sólo 10, por lo que su uso común es en transmisión de dígitos decimales.

b.3.3) CODIGO 3 FUERA DE 7.

Este código está diseñado para el código Baudot y emplea 35 combinaciones o palabras de código para las 32 ordenaciones que se deben transmitir. Para un funcionamiento del sistema en forma síncrona, sin utilización de los pulsos de arranque y parada, se pueden emitir las palabras de código de 7 bits en los mismos intervalos de tiempo que los carácteres de 5 bits.

El código 3 fuera de 7 encuentra su mayor aplicación en el sistema ARQ (Automatic Request) desarrollado por H. C. A. Van Deuren para la detección y corrección de error en los sistemas telegráficos.

El sistema ARQ requiere un enlace duplex, es decir, un canal principal para la transmisión de información en una dirección y un canal de retorno para la transmisión simultánea de una señal de receptor a transmisor, por medio de la cual el primero le indica al segundo que ha detectado error y le pide, por tanto, que retransmita el mensaje.

Como se ve, el transmisor precisa disponer de memoria para almacenar el bloque de datos transmitido hasta que reciba una confirmación o un reconocimiento negativo de parte del receptor a través del canal de retorno. El sistema ARQ con el código 3 fuera de 7, es utilizado en los enlaces internacionales del Sistema Telex de México.

En el código 3 fuera de 7 siempre hay una relación constante de 3 unos y 4 ceros ó 3 ceros y 4 unos en cada carácter.

CODIGO 4 FUERA DE 8.

Aquí cada carácter es una combinación de 4 bits 0 y 4 bits 1. Cualquier error que altere esta relación es detectado.

El número de palabras de código del código 4 fuera de 8 son:

$$Q = \frac{N!}{M!\,(N-M)!} = \frac{8!}{4!\,(8-4)!} = 70$$

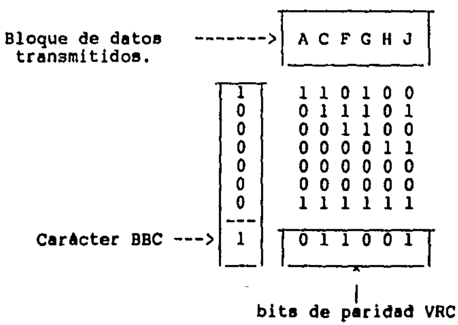
Este código ofrece mayor protección contra el ruido que los métodos de comprobación de paridad, pero la redundancia necesaria para lograrlo no lo ha hecho de uso general.

b.3.4) CODIGO VRC Y LRC.

Los métodos de detección de error VRC (Vertical Redundancy Check) y LRC (Longitudinal Redundancy Check) son usados en la transmisión de carácteres en código ASCII y por lo tanto son de los más empleados en la comunicación de datos. Pueden utilizarse en forma independiente o como es común en forma asociada.

En el método VRC en cada carácter del código ASCII a ser transmitido se hace una comprobación de paridad y se agrega al final un bit de paridad que puede ser 1 ó 0 según el número de bits 1 del carácter y la paridad escogida (par o impar).

En el método LRC, los carácteres sucesivos del bloque de datos transmitidos son sumados (suma de mod-2) (*6) al contenido de un registro BCC (block character check) y son simultáneamente transmitidos sobre la linea de comunicación. Así cada bit del carácter BCC sirve como una comprobación de paridad sobre todos los bits de datos que están en las posiciones correspondientes a esos bits, incluyendo el bit de posición de paridad VRC. La siguiente figura ilustra el método VRC y LRC con carácteres del código ASCII.



El carácter J es el primero transmitido y A es el último. Después de que el último carácter del mensaje ha sido sumado al registro BCC, el contenido de éste, que constituye el carácter BCC, es transmitido inmediatamente en el canal de comunicación.

(*6) Véase Cap. II.c.1.- Base matemática de los códigos de bloque

En esta forma la secuencia de la transmisión (considerándola síncrona) es:

$$\longrightarrow$$

S Y N	S Y N	S O H	ENCABEZAMIENTO	S T X	TEXTO	E T B	B C C

El registro BCC es puesto a 0 (borrado) al final de cada transmisión del bloque de datos después del carácter ETB.

En el extremo receptor se hace la comprobación de la paridad sobre cada carácter recibido. Después, es también sumado (suma de módulo 2) al contenido de un registro BCC. Cuando el último carácter de datos (indicado por el carácter ETB) ha sido recibido, el contenido del registro BCC debe ser idéntico al carácter BCC generado y enviado por el transmisor. Por eso cuando este carácter es sumado al registro BCC del receptor, el resultado debe ser igual a cero. De otro modo ha habido error en la transmisión.

El método VRC/LRC detecta todos los modelos de error que involucran un número impar de bits erróneos dentro de al menos uno de los caracteres del bloque de datos (por medio del método VRC) o un número impar de bits erróneos en al menos una de las posiciones de bits de todos los caracteres (a través del método LRC).

Un modelo de error que no se encuentre dentro de las condiciones anteriores no será detectado.

EFICIENCIA DEL METODO VRC/LRC.

Si: X = número de caracteres en el bloque de datos,
v = número de bits por carácter,
K = número de bits de información en el bloque,
n = número total de bits en el bloque,

la eficiencia del método VRC/LRC se calcula de la forma siguiente:

$$K = Xv$$

$$n = Xv + X + v + 1 = (X + 1)(v + 1)$$

luego la eficiencia es:

$$E = \frac{K}{n} = \frac{Xv}{(v+1)\ (X+1)} = \frac{v}{v+1}\ \frac{X}{X+1}$$

Para bloques de datos muy grandes (X muy grande), la eficiencia se aproxima a:

$$E = \frac{v}{v+1}$$

que es la máxima eficiencia de transmisión que puede lograrse con este método.

b.4) DETECCION DE ERROR POR EL CODIGO CICLICO, DE BLOQUE O POLINOMIAL.

Este método de detección de error es usado en la transmisión de carácteres en el código EBCDIC. También es empleado comunmente en los circuitos detectores y correctores de error en un sistema de almacenamiento de información de un computador, donde éste bien puede ser una unidad de disco, un tambor o una cinta magnética. En la transmisión de fotografias enviadas desde el espacio también se emplean códigos ciclicos.

Históricamente, los códigos detectores y correctores de error han sido divididos en dos áreas:

- Códigos de Bloque.

- Códigos Convolucionales.

En los códigos de bloque los n-k bits de paridad de bloque, checan solamente los bits de información precedentes a ellos en el tiempo. El bloque de n bits es llamado una palabra de código.

En los códigos convolucionales, los n-k bits de paridad checan también los bits de información que aparecen en bloques precedentes.

Un grupo importante de los códigos de bloque son los códigos ciclicos o polinomiales que tienen dos principales ventajas:

1) Poseen una estructura matemática suficiente para constituir una herramienta poderosa en manos de los técnicos de la codificación que les permite diseñar códigos con buenas propiedades de detección y corrección de errores.

2) La naturaleza ciclica de los códigos usualmente simplifica considerablemente el equipo de codificación y decodificación requerido.

Se denominan ciclicos porque tienen la propiedad de que cada cambio ciclico de una palabra de código es otra palabra válida del código. Asi, si:

Palabra de código = $(a_{n-1}, a_{n-2}, \ldots, a_0)$

entonces: $(a_{n-2}, a_{n-3}, \ldots, a_{n-1})$,

$(a_{n-3}, a_{n-4}, \ldots, a_{n-2})$, etc.

son también palabras de código.

Para entender el principio de detección de error del código ciclico, se considerará que la secuencia de k digitos binarios (0 y 1) que forman el bloque de datos a transmitir, representan los coeficientes de un polinomio M(X).

Por ejemplo, supóngase que el bloque de datos a transmitir sea:

1 1 1 0 1

entonces el polinomio M(X) será:

$$M(X) = 1X^4 + 1X^3 + 1X^2 + 0X + 1X^0.$$

Para realizar la codificación cíclica del bloque de datos para transmisión, se requiere la utilización de otro polinomio que se denominará polinomio generador P(X). El polinomio generador tiene grado r y un término $X^0 = 1$.

Los pasos para la codificación son:

1.- El polinomio M(X) representativo del bloque de datos es multiplicado por X**r con el fin de anexar una secuencia de r ceros a los bits de información en las posiciones de orden más bajo. Esto da:

M(X) X**r = N(X).

2.- El resultado anterior es dividido por P(X), lo cual da un cociente Q(X) y un residuo R(X). Matemáticamente se tiene:

$$\frac{N(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$$

o sea: $N(X) = Q(X)\ P(X) + R(X)$,

donde + = suma de módulo 2 o suma or exclusiva.

3.- El residuo es sumado a N(X) en forma binaria, colocando así r términos en las r posiciones más bajas donde había r ceros. El mensaje codificado es entonces expresado como:

$$F(X) = N(X) + R(X) = N(X) - R(X) = Q(X)\ P(X)$$

ya que la sustracción es igual que la adición en módulo 2.

Como se ve, F(X) es un múltiplo entero de P(X), es decir, es exactamente divisible por el polinomio generador P(X). Esta propiedad de F(X) es lo que permite la detección de error en el receptor. Cuando el mensaje codificado es recibido en el receptor, es dividido por el polinomio generador P(X). Si el residuo de esta división es diferente de cero, es que ha habido error en la transmisión. Si el residuo de la división es cero, la transmisión fue tal que el método no detectó error alguno.

Supóngase que E(X) es el polinomio de error que puede o no ser cero. Entonces, el mensaje recibido será:

$$F(X) + E(X) = Q(X)\ P(X) + E(X)$$

y al dividir por el polinomio generador obtenemos:

$$\frac{Q(X)\ P(X) + E(X)}{P(X)} = Q(X) + \frac{E(X)}{P(X)}$$

Como podemos ver, cualquier modelo de error E(X) que no es mùltiplo entero de P(X) serà detectado. Si, en cambio, E(X) es exactamente divisible por P(X), el error no serà detectado pues el residuo serà igual a cero. Con este mètodo, una ràfaga (burst) de error de ancho b<=r siempre serà detectada. (r es el grado del polinomio generador).

Una ràfaga de error es una secuencia de bits de la cual 2 ò màs son erròneos, con la secuencia siendo precedida y seguida por secuencias comparativamente màs grandes de bits correctos. El ancho de una ràfaga de error abarca del primero al ùltimo bit erròneo de la secuencia.

EJEMPLO:

Supòngase que se quiere transmitir la secuencia de bits siguiente: 1 1 1 0 1. Para proteger esa informaciòn necesitamos crear el polinomio generador P(X) con un tèrmino X^0 y de grado r. Para nuestro ejemplo seleccionamos un polinomio generador de grado r = 3, por lo que tenemos:

$M(X) = 1\ 1\ 1\ 0\ 1 = X^4 + X^3 + X^2 + 1$

$P(X) = 0\ 1\ 0\ 0\ 1 = X^3 + 1$

$N(X) = M(X)*(X**r) = 1\ 1\ 1\ 0\ 1\ 0\ 0\ 0 = X^7 + X^6 + X^5 + X^3.$

Si dividimos el polinomio correspondiente a N(X) entre el polinomio generador P(X), obtendremos el cociente Q(X) y el residuo R(X) siguientes:

$Q(X) = 1\ 1\ 1\ 1\ 0 = X^4 + X^3 + X^2 + X$

$R(X) = 0\ 0\ 1\ 1\ 0 = X^2 + X$

por lo que el mensaje codificado F(X) queda expresado de la siguiente forma:

$F(X) = Q(X)\ P(X) = 1\ 1\ 1\ 0\ 1\ 1\ 1\ 0 = X^7 + X^6 + X^5 + X^3 + X^2 + X$

Supongamos ahora que hay un polinomio de error $E(X) = X^4$. Con este polinomio de error, el mensaje que llegarà al lado receptor serà:

$F(X) + E(X) = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0 = X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X.$

En transmisiones sin error, el polinomio correspondiente al mensaje recibido es exactamente divisible entre el polinomio generador. En este caso el residuo de la división no es cero, por lo tanto existe un error en la transmisión.

Como se puede observar, este código no nos indica cuál es el bit que tiene error y por eso no podemos corregirlo. Nótese que si el error hubiera sido un polinomio múltiplo de P(X), el código no sería capaz de detectarlo.

C) CODIGOS DE BLOQUE Y CONVOLUCIONALES.

C.1) BASE MATEMATICA DE LOS CODIGOS DE BLOQUE

Introducción.

En este capitulo se mostrará una técnica de codificación para información binaria basada en agregar digitos redundantes a la secuencia de información que forma una palabra. Con estos digitos redundantes aumenta la seguridad de localizar los errores en la secuencia de digitos binarios transmitida. Este tipo de códigos protege palabras utilizando la idea central de verificación de paridad. Debido a este procesamiento por palabras es que se les denomina "Códigos de Bloque".

Ventajas.

Entre las ventajas que presentan estos códigos se pueden mencionar las siguientes:

- Pueden tanto detectar como corregir errores.
- Se puede agregar cualquier número de digitos redundantes; la capacidad de detección y corrección de errores depende de este número de digitos redundantes.
- Son fácilmente realizables.

Evidentemente el código a utilizar depende del canal a través del cual se enviará la información. Para transmitir por un canal muy ruidoso se necesita mucha protección para la información (o sea, muchos digitos redundantes); para transmitir por un canal con probabilidad de error (por digito) muy baja, se necesitan digitos redundantes en menor número para obtener, en promedio, el mismo número de errores, por secuencia de palabras, que en el caso anterior.

Definiciones.

Sea $u = (u_1, u_2, \ldots, u_n)$ un vector de información o mensaje.

Al vector $x = (f_1(u), f_2(u), \ldots, f_n(u)) = uG$ se le llamará palabra codificada o palabra transmitida, donde

$$f_i = \sum_{i=1}^{n} u_j . g_{ji}$$

y G es una matriz formada por los elementos g_{ji} y que se definirá más adelante.

Al conjunto de vectores $\underline{x}$ se le llamará código y se dirá que $\underline{x}$ es una palabra del código.

Si la palabra transmitida tiene "r" digitos, donde

$$r = n + m,$$

entonces, los últimos $m = r - n$ digitos se llaman digitos redundantes; los primeros "n" digitos son los digitos de información.

Obsérvese que cambiar uno o más digitos de una palabra codificada $\underline{x}$ es equivalente a sumarle (mod - 2) a $\underline{x}$ un vector $\underline{z}$ con unos en las posiciones en que se cambiaron los digitos de $\underline{x}$ y ceros en las demás.

Por ejemplo, sea:

$$\underline{x} = (1\ 0\ 1\ 1\ 0\ 0\ 1\ 0).$$

Llámese $\underline{y}$ al vector $\underline{x}$ con uno o varios digitos alterados:

$$\underline{y} = (1\ 1\ 0\ 1\ 0\ 1\ 1\ 0).$$

Entonces el vector $\underline{z}$ que satisface $\underline{x} = \underline{y} + \underline{z}$ es:

$$\underline{z} = (0\ 1\ 1\ 0\ 0\ 1\ 0\ 0).$$

Utilizando estas propiedades se llamará a $\underline{z}$ vector de errores y a $\underline{y}$ palabra recibida. Nótese que si

$$\underline{z} = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$

para todas las palabras transmitidas, entonces se puede concluir que el canal de transmisión no introduce errores y además $\underline{x} = \underline{y}$, o sea, que la palabra recibida es igual a la transmitida.

Nótese que $\underline{y} = \underline{x} + \underline{z}$ o su equivalente es $\underline{z} = \underline{x} + \underline{y}$.

Al número de digitos diferentes de cero en una secuencia se le llamará peso de la secuencia; al minimo número de digitos diferentes de cero en las sumas (mod-2) de dos palabras cualesquiera de un código, se le llamará distancia del código.

Un código que genera palabras de longitud "r" a partir de mensajes de longitud "n" se llama código (r,n).

En álgebra mod-2 son válidas las leyes asociativa, distributiva y conmutativa de la aritmética.

CODIFICACION.

Como se mencionó anteriormente, las palabras del código (r,n) se forman a partir de combinaciones lineales de los elementos de u; éstas pueden ser descritas a través de un conjunto de ecuaciones de la forma:

$$f_i(u) = x_i = \sum_{j=1}^{n} u_j \, g_{ji} \qquad \begin{array}{l} 1 <= i <= r \\ 1 <= j <= r \end{array} \qquad (1)$$

donde:

x_i es la i-ésima componente de $\underline{x}$,

$\sum$ es la sumatoria mod-2 sobre el conjunto de índices j,

g_{ji} representa un conjunto de dígitos binarios arbitrarios pero fijos.

Se hablará de un código sistemático (r,n) si los dígitos g_{ji} forman una matriz identidad de dimensión n para:

$$1 <= j <= n$$

$$1 <= i <= n.$$

Obviamente la ecuación (1) representa un conjunto de ecuaciones lineales; éstas pueden ser escritas matricialmente como:

$$\underline{x} = uG \qquad (2)$$

G es una matriz de dimensiones n x r cuyos elementos son los números g_{ji} del conjunto descrito anteriormente.

Esto es:

$$G = \begin{vmatrix} g_{11} & g_{12} & \cdots & g_{1r} \\ g_{21} & g_{22} & \cdots & g_{2r} \\ \vdots & \vdots & & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nr} \end{vmatrix} \qquad (3)$$

y si el código es sistemático:

$$G = \begin{vmatrix} 1 & 0 & \dots & 0 & g_{1,n+1} & \cdots & g_{1,r} \\ 0 & 1 & \dots & 0 & g_{2,n+1} & \cdots & g_{2,r} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 & g_{n,n+1} & \cdots & g_{n,r} \end{vmatrix} \quad (4)$$

Lo cual puede ser escrito como:

$$G = \left[I_n \mid G' \right] \quad (5)$$

donde I_n es la matriz identidad de dimensión n.

Debido a que a través de la ecuación (2) se generan las palabras del código, a la matriz G se le llama matriz generadora.

DECODIFICACION.

Por simplificar matemáticamente el proceso de decodificación, se utilizarán códigos sistemáticos.

La ecuación (1), para un código sistemático, puede ser expresada como:

$$x_j = u_j \qquad \text{para} \quad 1 <= j <= n \quad (6)$$

$$x_i = \sum_{j=1}^{n} g_{ji}\, u_j \qquad \text{para} \quad n < i <= r \quad (7)$$

y sustituyendo (6) en (7):

$$x_i = \sum_{j=1}^{n} g_{ji}\, x_j \qquad \text{para} \quad n < i <= r \quad (8)$$

o similarmente:

$$x_i + \sum_{j=1}^{n} g_{ji}\, x_j = 0 \quad \text{para } n < i <= r \tag{9}$$

La ecuación (9) puede ser desarrollada y escrita en forma matricial quedando:

$$(x_1\ x_2\ \ldots\ x_r) \left| \begin{array}{cccc} g_{1,n+1} & g_{1,n+2} & \cdots & g_{1,r} \\ g_{2,n+1} & g_{2,n+2} & \cdots & g2,r \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ g_{n,n+1} & g_{n,n+2} & \cdots & g_{n,r} \\ \hline 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \ldots & 1 \end{array} \right| = 0$$

o equivalentemente:

$$\underline{x} \left| \begin{array}{c} G' \\ \hline Im \end{array} \right| = 0 \tag{10}$$

Si se define la matriz H por medio de:

$$H = \left| \begin{array}{c} G' \\ \hline Im \end{array} \right| \tag{11}$$

entonces, (10) se transforma en:

$$\underline{x}H = 0. \tag{12}$$

Es evidente que si una palabra se genera a partir de (2), ésta debe satisfacer (12); por otro lado, una palabra que satisface (12) debe haber sido generada por medio de (2).

Es importante hacer notar que el conjunto de palabras de un código de bloque es mayor que el número de posibles mensajes a codificar; si se codifican 2**n mensajes de longitud "n" con un código (r,n), donde r>n, entonces evidentemente 2**r > 2**n; esto significa que existen secuencias de longitud "r" que no pertenecen al código (r,n); se tratará que estas secuencias sean precisamente la suma de palabras transmitidas y vectores de error.

De aquí se puede concluir que los elementos de la matriz generadora deben ser fijados de tal forma que, si se desea corregir errores de orden "1" (los vectores de error tienen peso "1"), la distancia entre la palabra recibida y cualquier palabra del código, excepto la transmitida, debe ser mayor que la distancia entre la palabra recibida y la transmitida, o sea, mayor que 1. Este criterio se llama criterio de máxima similitud.

Como se recordará: $\underline{y} = \underline{x} + \underline{z}$ (13)

Pero si $\underline{y}$ no pertenece al código:

$$\underline{y}H <> 0 \qquad (14)$$

La ecuación (14) encierra la idea básica para la detección de errores; la ecuación (13) se utilizará para su corrección.

Defínase el síndrome S en la siguiente forma:

$$S = \underline{y}H \qquad (15)$$

De lo anterior se concluye que si S = 0, entonces $\underline{y}$ es una palabra del código; si S <> 0, entonces $\underline{y}$ no pertenece al código y la palabra transmitida $\underline{x}$ fue alterada por el vector de error $\underline{z}$ (debido evidentemente al ruido del canal).

Esto es: $\underline{y} = \underline{x} + \underline{z}$,

resultando el síndrome: $S = (\underline{x}+\underline{z})H = \underline{x}H + \underline{z}H$ (16)

y debido a la ecuación (12): $S = \underline{z}H <> 0$ (17)

porque $\underline{z}$ no pertenece al código que tiene una distancia mayor que cualquier vector de error. (Se supone que el canal tiene probabilidad no nula de error por dígito; sin embargo, la probabilidad de alterar "d" dígitos de una palabra, que resultaría en otra palabra del código, es insignificante).

El síndrome es un vector de dimensión |m = r - n|, o sea, que existen solo 2**m síndromes diferentes; si se asocia a cada posible síndrome un vector de error del conjunto de posibles errores (que tiene 2**r elementos), una vez conocido el síndrome se podrá determinar la secuencia de error que probablemente lo originó. Por tanto, los vectores de error a los que se les asociará un síndrome, deberán ser los de mayor probabilidad de ocurrencia (errores sencillos).

Supóngase que el vector de error z_l tiene alta probabilidad de ocurrencia; asóciesele el síndrome S_l. La palabra transmitida es x_l. La recibida, evidentemente es: $y_l = x_l + z_l$. A partir de y_l, se encuentra S_l y de éste, z_l. En esta forma se determina

$$x_l = y_l + z_l.$$

Por las propiedades de detección de la matriz H, a ésta se le llama matriz de verificación.

EJEMPLO.

Se desea codificar mensajes binarios de 3 dígitos en forma sistemática. El canal de transmisión requiere que se corrijan por lo menos todos los errores de peso unitario (errores sencillos); se supone además que todos los errores de peso 2 tienen igual probabilidad de ocurrencia.

Solución.

El número de mensajes es $2^3 = 8$. Como el código debe corregir todos los errores sencillos y éstos pueden aparecer en cualquier posición, se necesita un número mínimo (3 + m) de síndromes, donde m = número de dígitos redundantes.

Por tanto, se agregarán 3 dígitos redundantes para tener en total 6 dígitos por palabra y 8 diferentes síndromes.

Como H debe tener todos sus renglones diferentes para que al multiplicar por cada uno de los errores sencillos se obtengan síndromes distintos, se iniciará el diseño del código con esta matriz cuya estructura está dada por (11):

$$H = \left| \begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right| = \left| \begin{array}{c} G' \\ \hline I_3 \end{array} \right|$$

$$G = \left[I_3 \mid G' \right] = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{array} \right]$$

A partir de G se pueden determinar las funciones

$$(f_i)_{i=1}^{6} \qquad \text{para } u = (1\ 0\ 1):$$

Estas son:

$$x_1 = f_1\ u = u_1 = 1$$

$$x_2 = f_2\ u = u_2 = 0$$

$$x_3 = f_3\ u = u_3 = 1$$

$$x_4 = f_4\ u = u_1 + u_3 = 0$$

$$x_5 = f_5\ u = u_1 + u_2 = 1$$

$$x_6 = f_6\ u = u_2 + u_3 = 1$$

El siguiente paso es asignar un síndrome a cada error probable, a través de una tabla de decodificación; de acuerdo al planteamiento, los errores más probables son los más sencillos:

TABLA DE DECODIFICACION

SINDROME S	VECTOR DE ERRORES $\underline{z}$
0 0 0	0 0 0 0 0 0
0 0 1	0 0 0 0 0 1
0 1 0	0 0 0 0 1 0
1 0 0	0 0 0 1 0 0
1 0 1	0 0 1 0 0 0
0 1 1	0 1 0 0 0 0
1 1 0	1 0 0 0 0 0

Debido a que todos los errores de peso 2 son equiprobables, se escogió arbitrariamente úno cuyo síndrome fuera el que faltaba en la tabla: 1 1 1.

Así, si se recibe $\underline{y} = (1\ 0\ 1\ 0\ 1\ 0)$

entonces $S = (0\ 0\ 1)$

y $\underline{z} = (0\ 0\ 0\ 0\ 0\ 1)$

por lo cual $\underline{x} = \underline{y} + \underline{z} = (1\ 0\ 1\ 0\ 1\ 1)$

y $u = (1\ 0\ 1)$

Este código detecta y corrige todos los errores sencillos y un error doble.

C.2) BASE MATEMATICA DE LOS CODIGOS CONVOLUCIONALES.

Introducción.

Debido a que la información digital no siempre está estructurada en forma de palabras (como en PCM), es necesario estudiar otro tipo de códigos que permitan la protección de la información en forma serial (tal y como se obtendria a partir de un modulador Delta).

Se protegeria la información digito a digito intercalando en la secuencia de información digitos redundantes. A un código convolucional que contiene "k" digitos de chequeo por cada "n" digitos de información, se le llamará código convolucional (k,n).

Definiciones, Codificación y Decodificación.

Sea $u = (u_1, u_2, \ldots, u_n)$ un vector de información.

A la secuencia

$$\underline{x} = (x_{11}, x_{12}, x_{13}, \ldots, x_{1k}, x_{21}, \ldots, x_{2k}, \ldots, x_{31}, \ldots)$$

se le llama secuencia codificada o secuencia transmitida; x_{ij} es generado a través de combinaciones lineales de los "l" digitos anteriores a u_{i+1}.

Esto es:

$$x_{ij} = f_j\ (u_{i-1}, \ldots, u_i) \qquad (18)$$

donde f_j (, ,) es una combinación lineal de los argumentos. A la secuencia

$$\underline{y} = (y_{11}, y_{12}, y_{13}, \ldots, y_{1k}, y_{21}, \ldots, y_{2k}, \ldots, y_{31}, \ldots)$$

se le llama secuencia recibida.

Es evidente que si el canal de transmisión no tiene ruido, entonces:

$$\underline{x} = \underline{y} \qquad (19)$$

Por otra parte, si el canal tiene ruido y el código se disena en forma adecuada a partir de $\underline{y}$, se obtiene la secuencia t tal que:

$$u = t \qquad (20)$$

donde a t se le llamará secuencia decodificada.

La secuencia decodificada t se obtiene de la siguiente forma:

$$t = (t_1, t_2, \ldots)$$

donde:

$$t = f(a)\ (y_{1,1}\ ,\ \ldots,\ y_{1,j}\ ,\ y_{2,1}\ ,\ y_{2,2}\ ,\ \ldots\ y_{2,j}\ ,\ \ldots) \quad (21)$$

y f(a) es una función lógica de los argumentos.

Se ilustrará por medio de un ejemplo la codificación y decodificación con este tipo de códigos.

EJEMPLO.

Se desea transmitir una secuencia de información

$$u = (1\ 0\ 1\ 1\ 0\ \ldots)$$

protegida por medio de un código convolucional (2,1).

El canal tiene una probabilidad de error tal que la probabilidad de tener 3 errores en 3 digitos es mucho menor a la probabilidad de tener 2 errores en 3 digitos y ésta, a su vez, es mucho menor a la probabilidad de tener 1 error cada 3 digitos.

Solución.

Una posible codificación es:

$$x_{i1} = u_i = x_{i2}$$

resultando la secuencia transmitida:

$$\underline{x} = (u_1,\ u_1,\ u_1,\ u_2,\ u_2,\ u_2,\ \ldots)$$

o sea:

$$\underline{x} = (111\ 000\ 111\ 111\ 000\ 111\ \ldots)$$

La decodificación y corrección se efectúan de la siguiente manera:

$$t_a = \left| \begin{array}{lll} y_{a1} & \text{si} & y_{a1} = y_{a2} \\ y_{a2} & \text{si} & y_{a2} = y_{a3} \\ y_{a3} & \text{si} & y_{a3} = y_{a1} \end{array} \right| \quad (22)$$

La relación (22) es precisamente la función lógica mencionada en (21).

Supóngase que la secuencia recibida sea:

$$u = (111\ 001\ 011\ 111\ 001\ \ldots)$$

Usando t_a como se definió en (22), resulta que:

$$t = (111\ 000\ 111\ 111\ 000\ \ldots)$$

la cual es igual al mensaje.

El código del ejemplo anterior es apropiado para el canal descrito en la hipótesis; si fuese muy probable la aparición de dos errores en 3 dígitos consecutivos, éste código obviamente podría provocar que t <> u con una probabilidad considerable.

Realización.

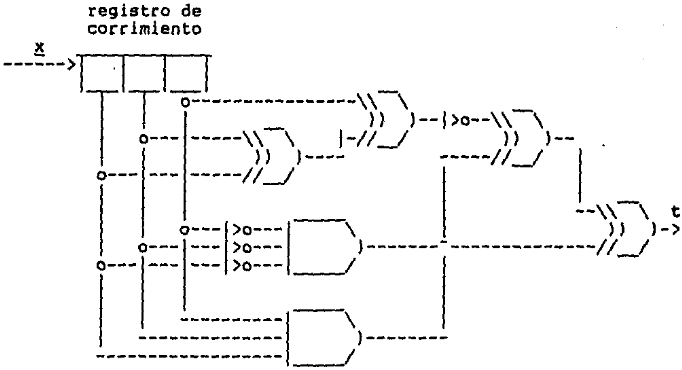
El codificador puede ser realizado con el siguiente circuito:

```
                     -->------.
                    |          .
      ...u2u1       |         .   /                   x
 -----------------)---|-->---.  <----(----o-------------------
                    |         .    v
                    |          .
                     -->------.
```

Un mismo dígito es alimentado al canal de transmisión 3 veces para que 3 dígitos consecutivos de x sean iguales.

La frecuencia del conmutador debe ser 3 veces mayor que la duración de un dígito (u)**(-1).

La decodificación puede ser realizada con el circuito mostrado a continuación:



Fácilmente se puede demostrar que el circuito mostrado decodifica correctamente de acuerdo a la siguiente tabla:

$y_{a,3}$	$y_{a,2}$	$y_{a,1}$	t_a
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

La duración del dígito t_a es tres veces mayor que la duración de cualquier dígito de y.

CAPITULO III.- SELECCION DEL CODIGO EN LA APLICACION A MICROPROCESADORES.

a) PLANTEAMIENTO DEL PROBLEMA

Es sabido que en la transmisión de datos el principal problema con el que nos enfrentamos es la pérdida de información y ello repercute en un gasto adicional de recursos materiales y humanos. El hecho de no detectar errores de transmisión a tiempo significa trabajar con datos falsos, los cuales tienen que ser corregidos tarde o temprano, implicando así un tiempo de reproceso que se traduce en gastos que pueden ser evitados si contamos con un medio adecuado para detectar esos errores en el momento oportuno.

La probabilidad de que no haya errores en el mensaje recibido es mayor que la probabilidad de que llegue un bit erróneo y ésta a su vez es mayor que la probabilidad de que lleguen dos bits erróneos y así sucesivamente. De aquí se puede inferir que la probabilidad de tener más de dos errores es despreciable; ésta es una razón poderosa para que en la práctica lo más usual sea detectar un solo error y ya sea corregirlo o retransmitir el mensaje completo. Con el método de paridad simple, el más comunmente usado para detección, no se pueden detectar números pares de errores, lo cual limita su aplicación.

La técnica convencional en los sistemas de comunicación para recuperar información ha sido hasta ahora la retransmisión.

Esto es debido a que el método más usual de detectar errores en la transmisión es el chequeo de paridad simple, el cual no permite corregir los errores en el lado receptor, sólo detectarlos.

Cabe mencionar que es posible detectar y corregir todos los errores que pudieran ocurrir al transmitir un dato, pero para lograr esto se requerirían muchos bits redundantes. Esto causaría que en la transmisión hubiera un mínimo de información acompañada de excesiva protección, disminuyendo de esta forma la eficiencia del sistema.

El objetivo que se persigue con este trabajo es el poder corregir los errores sencillos en el lado receptor sin necesidad de retransmitir y tener la capacidad de detectar errores dobles, en cuyo caso tendrá lugar la retransmisión. Para cumplir dicho objetivo se seleccionará el código más apropiado para la aplicación específica a microprocesadores de 16 bits, por ser los más ampliamente usados en este momento.

b) POSIBLES SOLUCIONES: SOFTWARE VERSUS HARDWARE.

Es obvio que se tiene que seleccionar un código corrector de error, de entre varios que se conocen, con el fin de aplicarlo en la solución del problema planteado. Se han mencionado los aspectos fundamentales que determinarán cuál es el código óptimo. Es posible analizar varios puntos, pero la facilidad de aplicación es el que interesa comentar en este momento, por la finalidad que se tiene en este trabajo.

Una vez seleccionado el código se tiene que realizar prácticamente. Para hacerlo, existen dos formas conocidas: o se hace un programa en lenguaje ensamblador para guardarlo en la memoria de cada uno de los microprocesadores que actúan en la transmisión y recepción, o se arma un circuito que realice las mismas funciones que el programa.

Si se quisiera solucionar el problema a través de software, se requerirían dos programas: uno en el microprocesador transmisor, que se encargaría de generar la protección para los bits de información y otro en el microprocesador receptor, más completo que el primero, que haría el trabajo de detección y tomaría la decisión de corregir el bit erróneo o mandar señal de retransmisión.

En el caso del microprocesador, se tiene limitación de memoria, por lo que resulta inoperante elaborar los programas que realicen las funciones de detección y corrección equivalentes en hardware. Aunado al problema de memoria, el tiempo de proceso requerido para la ejecución de los programas haría muy lento el flujo de información en el sistema.

Por lo anteriormente expuesto, se ha optado por solucionar el problema a través de hardware, además de que ésta es la forma más comúnmente utilizada en la práctica.

c) SELECCION DEL CODIGO. VENTAJAS Y DESVENTAJAS.

Como hemos visto, existe un gran número de códigos correctores y nuestra tarea ahora será seleccionar el que más convenga a este trabajo, de acuerdo a los objetivos que en un principio nos fijamos.

Analizando las características de los dos tipos principales de códigos correctores, decidimos no trabajar con códigos convolucionales ya que su uso es más eficiente cuando se pretende la transmisión de información en alguna forma diferente a la palabra. En este caso trabajaremos con palabras de 16 bits. De esta importante eliminación, nos concretaremos a seleccionar de entre los códigos de bloque el que más se adapte a nuestras necesidades.

- Códigos de Repetición: tienen una gran capacidad de corrección de error, pero solamente aceptan un bit de mensaje por bloque ya que los demás bits son de redundancia. Por el mismo hecho de tener tantos bits de protección, su eficiencia en cuanto a transmisión de información es bastante pobre por lo que evitaremos su utilización.

- Códigos de Simple Chequeo de Paridad: este tipo de código emplea un solo bit de redundancia en cada palabra, por lo que su eficiencia de transmisión es muy alta pero su capacidad de correción es nula y solamente puede detectar un número impar de errores, ya que al haber un número par de ellos la paridad no se altera. Estas razones hacen imposible poder tomar en cuenta este tipo de código para resolver nuestro problema.

- Código de Hamming: este tipo de código es bastante bueno para corrección de errores sencillos por su simplicidad y fácil aplicación. Su desarrollo matemático es excelente y bastante fácil de aplicar, por lo que es más ampliamente usado en la práctica que cualquier otro tipo de código corrector. Existe una gran variedad de códigos basados en el de Hamming, cuyo desarrollo y aplicación son reducidos por ser códigos de propósito muy especial. Este tipo de código no es muy aplicable en la corrección de errores en paquete o múltiples, como los convolucionales.

- Códigos BCH (Bose-Chaudhuri-Hocquenghen): Junto con los códigos de Hamming, forman el conjunto más importante dentro de los códigos de bloque. Son bastante más complejos que los anteriormente mencionados ya que corrigen errores múltiples. Su aplicación es más reducida que el chequeo de paridad simple y que los códigos de Hamming, aunque su uso es recomendable en caso de requerir corrección múltiple.

Existen muchísimos más códigos correctores, pero invariablemente adolecen de algún problema como lo es su poco desarrollo, su propósito especial o su dificultad de utilización.

Como puede verse en lo anteriormente expuesto, el tipo de código que más se acerca a satisfacer los requerimientos del presente trabajo es el código de Hamming, por su simplicidad de aplicación, su desarrollo matemático y por ser uno de los más ampliamente utilizados. El código BCH sería excelente si quisiéramos corrección de errores múltiples, objetivo que no corresponde al presente trabajo, por lo que prescindiremos de dicho código.

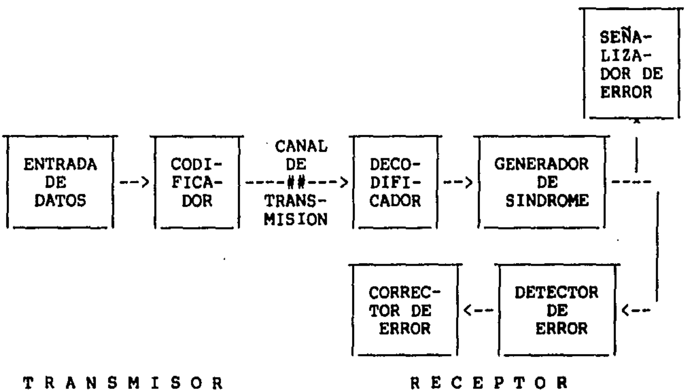
En el siguiente capítulo describiremos a detalle la forma en que trabaja el código que hemos seleccionado y se presentará el circuito diseñado en base a dicho código.

CAPITULO IV.- SOLUCION DEL PROBLEMA.

a) DESCRIPCION GENERAL, DIAGRAMAS Y ECUACIONES.

En el capítulo tercero hemos presentado la base matemática de los códigos de bloque, entre los cuales se encuentra el Código de Hamming, por lo que desarrollaremos nuestro proyecto a partir de dicha explicación.

El siguiente diagrama de bloques muestra las diferentes etapas de que va a constar el sistema final y que iremos desarrollando a lo largo de este capítulo:



En primer lugar vamos a tener una matriz generadora de bits de chequeo en el lado transmisor, los cuales dependerán de la información de la palabra original que, como hemos acordado, será de 16 bits. Para generar dicha matriz es necesario primero definir las ecuaciones a partir de las cuales obtendremos los bits de chequeo. Las ecuaciones serán definidas tomando en cuenta las siguientes reglas básicas:

- Todos y cada uno de los bits de datos de entrada deberá participar en el mismo número de ecuaciones;

- No se puede tomar en cuenta más de una vez al mismo bit de dato en una misma ecuación;

- Se tendrán tantas ecuaciones como sean necesarias de acuerdo a la capacidad de detección y corrección que se pretenda obtener.

En el capitulo II inciso b.2) de esta tesis hemos mostrado una tabla conteniendo las distancias de Hamming necesarias en un código para obtener diferentes capacidades de detección y corrección.

Nuestro objetivo es poder detectar cualquier error doble o sencillo y corregir todos los errores sencillos, por lo que necesitamos tener una distancia de Hamming mayor o igual a 4. Para obtener esto y suponiendo que el código que se utilice en la transmisión tenga una distancia de Hamming de 1, los bits de chequeo que generemos deben aumentarla a 4 por lo menos.

Analizando las diferentes posibilidades, encontramos que el número óptimo de ecuaciones es 6 con 8 bits de datos participando en cada una de ellas, con lo que se tiene a cada uno de los 16 bits de datos en 3 ecuaciones diferentes por lo que la distancia de Hamming del código aumenta a 4 y entonces será posible detectar y corregir errores sencillos y detectar los dobles.

El circuito que pretendemos diseñar no tomará en cuenta las señales provenientes directamente del microprocesador, ya que sería difícil generar o simular errores en la transmisión en el momento de hacer las pruebas con el circuito, por lo cual hemos decidido demostrar su funcionamiento prescindiendo de micros, simulando los errores con interruptores colocados en el circuito mismo. Consideramos que con esta opción se puede obtener un mejor resultado didáctico al demostrar el funcionamiento del circuito. Ya que no contamos con micros, haremos nuestro circuito suponiendo que se reciben 16 bits en paralelo. En caso de que la información viniera en serie, bastaría utilizar un convertidor serie-paralelo para cubrir los requerimientos de entrada del circuito.

Las ecuaciones para generar los bits de chequeo las hemos definido, de acuerdo a las reglas básicas enumeradas anteriormente, de la siguiente forma:

$$
\begin{aligned}
BC0 &= \overline{BD0 + BD1 + BD3 + BD4 + BD8 + BD9 + BD10 + BD13} \\
BC1 &= \overline{BD0 + BD2 + BD3 + BD5 + BD6 + BD8 + BD11 + BD14} \\
BC2 &= \overline{BD1 + BD2 + BD4 + BD5 + BD7 + BD9 + BD12 + BD15} \\
BC3 &= \overline{BD0 + BD1 + BD2 + BD6 + BD7 + BD10 + BD11 + BD12} \\
BC4 &= \overline{BD3 + BD4 + BD5 + BD6 + BD7 + BD13 + BD14 + BD15} \\
BC5 &= \overline{BD8 + BD9 + BD10 + BD11 + BD12 + BD13 + BD14 + BD15}
\end{aligned}
$$

Estos 6 bits de chequeo son añadidos a la palabra original y transmitidos juntos en la forma que más convenga al caso en particular, serie o paralelo, o para que en el lado receptor se tomen los 16 bits de información recibidos y con ellos se generen otros 6 bits de chequeo con las mismas ecuaciones utilizadas en el lado transmisor y que son las que se muestran líneas arriba. De esta forma, tendremos dos grupos de 6 bits de chequeo, que serán exactamente iguales solamente en el caso de no tener errores en la transmisión. A cada bit del primer grupo de bits de chequeo lo llamaremos BCno y a cada bit del segundo grupo BCnr, donde 0<= n <=5.

Ahora bien, tenemos que realizar una comparación bit a bit de los dos grupos de bits de chequeo y al resultado de cada comparación le llamaremos bit de sindrome BSn. La comparación es hecha con sumas módulo 2 de la siguiente forma:

$$
\begin{aligned}
BS0 &= \overline{BC0o + BC0r} \\
BS1 &= \overline{BC1o + BC1r} \\
BS2 &= \overline{BC2o + BC2r} \\
BS3 &= \overline{BC3o + BC3r} \\
BS4 &= \overline{BC4o + BC4r} \\
BS5 &= \overline{BC5o + BC5r}
\end{aligned}
$$

Es fácil demostrar que una compuerta NOR EXCLUSIVO es suficiente para obtener cada bit de sindrome:

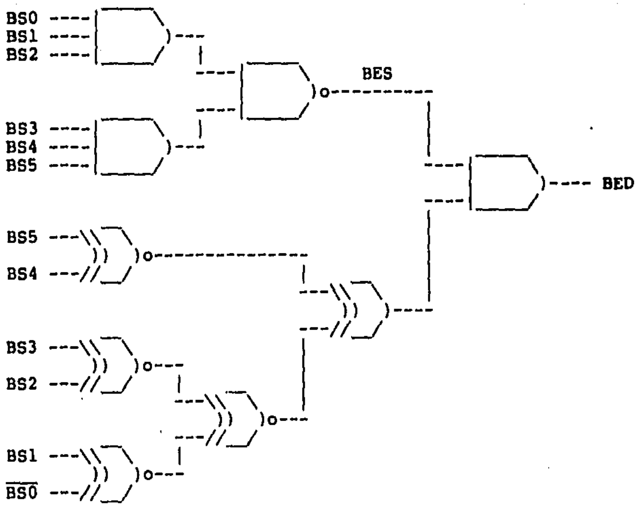
```
BCno -----\\‾‾\
           ))  )o----- BSn
BCnr -----//__/
```

El resultado de las comparaciones nos permitirá determinar si existe o no error en la palabra recibida. Las ecuaciones necesarias para realizar lo anterior son las siguientes:

$$BES = \overline{BS0 \;.\; BS1 \;.\; BS2 \;.\; BS3 \;.\; BS4 \;.\; BS5}$$

$$BED = BES \;.\; (\overline{\overline{BS0 + BS1} + \overline{BS2 + BS3}} + \overline{BS4 + BS5})$$

En estas ecuaciones BES es la bandera de Error Sencillo y BED es la Bandera de Error Doble y pueden ser añadidas al sistema final con el siguiente circuito:



Como se puede observar en el circuito, tendremos un indicador luminoso en cada salida que nos señalará el tipo de error ocurrido; si el error es doble, no podrá ser corregido por lo que se tendria que mandar una señal al lado transmisor con el fin de que retransmitan el mensaje. Si el error es sencillo, hay dos posibilidades: que el bit erróneo sea de información o de chequeo. Si es de chequeo no habrá corrección puesto que los de información son correctos, pero si es de información, tendremos que pasar por la etapa correctora.

Si alguno de los 22 bits transmitidos se modifica, quiere decir que un error ha ocurrido y los indicadores nos dirán el tipo de error. Cualquier error sencillo en alguno de los 16 bits de información, cambiará el valor de 3 de los 6 bits de síndrome y cualquier error sencillo en alguno de los bits de chequeo, cambiará el valor de un solo bit de síndrome. Por último, cualquier error doble cambiará el valor de un número par de bits de chequeo y, consecuentemente, de bits de síndrome; en este caso no se podrán corregir los errores y los dos indicadores estarán encendidos.

Se puede notar que utilizando la compuerta AND, basta con tener un solo bit de síndrome diferente a 1 para que la salida de la compuerta sea 0 y al negarla se encienda BES. En caso de no haber error, todos los bits de síndrome serán 1 lógico, por lo que al negar la salida BES seguirá apagada.

También es importante observar que si hay una pérdida total de información y recibimos solamente ceros, el circuito detectará error doble y dicha información no será aceptada como correcta. Esto es muy útil en caso de tener una gran atenuación en la línea.

Ya que hemos detectado un error sencillo, debemos pasar ahora a la fase correctora que consta de dos partes: identificación del error y corrección del mismo.

Para identificar el error, debemos generar una serie de bits correctores basados en los bits de síndrome que cada bit de información dañado pueda modificar. Partiremos del hecho de que cuando no existe error alguno, todos los bits de síndrome tienen el valor 1 lógico y que cuando hay algún error sencillo solo 3 bits de síndrome toman el valor 0 lógico. Entonces vamos a invertir el valor de los bits de síndrome involucrados en caso de error de cada bit de información por separado. A los bits correctores los llamaremos CORRn, donde $0 <= n <= 15$.

Para obtener los bits CORRn necesitamos tomar en cuenta los siguientes puntos importantes:

- Si no hay error, todos los bits BSn tienen un valor 1 lógico;

- Si hay algún error sencillo, solamente 3 bits de síndrome modifican su valor a 0 lógico, dependiendo de las ecuaciones en que el bit de dato erróneo participe;

- Todos los bits CORRn tendrán un valor 1 lógico en caso de no haber error en ninguno de los bits de datos;

- Solamente uno de los bits CORRn deberá valer 0 lógico en caso de error sencillo de uno de los 16 bits de datos y debe ser el que corresponda al bit BDn dañado.

Tomando esto en cuenta podemos ahora encontrar las ecuaciones para los bits CORRn y son las siguientes:

$$CORR0 = \overline{\overline{BS0} \cdot \overline{BS1} \cdot BS2 \cdot \overline{BS3} \cdot BS4 \cdot BS5}$$

$$CORR1 = \overline{\overline{BS0} \cdot BS1 \cdot \overline{BS2} \cdot \overline{BS3} \cdot BS4 \cdot BS5}$$

$$CORR2 = \overline{\overline{BS0} \cdot BS1 \cdot BS2 \cdot BS3 \cdot \overline{BS4} \cdot \overline{BS5}}$$

$$CORR3 = \overline{\overline{BS0} \cdot \overline{BS1} \cdot BS2 \cdot BS3 \cdot \overline{BS4} \cdot BS5}$$

$$CORR4 = \overline{BS0} \cdot BS1 \cdot \overline{BS2} \cdot BS3 \cdot \overline{BS4} \cdot BS5$$

$$CORR5 = BS0 \cdot \overline{BS1} \cdot \overline{BS2} \cdot BS3 \cdot \overline{BS4} \cdot BS5$$

$$CORR6 = BS0 \cdot \overline{BS1} \cdot BS2 \cdot \overline{BS3} \cdot \overline{BS4} \cdot BS5$$

$$CORR7 = BS0 \cdot BS1 \cdot \overline{BS2} \cdot \overline{BS3} \cdot \overline{BS4} \cdot BS5$$

$$CORR8 = \overline{BS0} \cdot \overline{BS1} \cdot BS2 \cdot BS3 \cdot BS4 \cdot \overline{BS5}$$

$$CORR9 = \overline{BS0} \cdot BS1 \cdot \overline{BS2} \cdot BS3 \cdot BS4 \cdot \overline{BS5}$$

$$CORR10 = \overline{BS0} \cdot BS1 \cdot BS2 \cdot \overline{BS3} \cdot BS4 \cdot \overline{BS5}$$

$$CORR11 = BS0 \cdot \overline{BS1} \cdot BS2 \cdot \overline{BS3} \cdot BS4 \cdot \overline{BS5}$$

$$CORR12 = BS0 \cdot BS1 \cdot \overline{BS2} \cdot \overline{BS3} \cdot BS4 \cdot \overline{BS5}$$

$$CORR13 = \overline{BS0} \cdot BS1 \cdot BS2 \cdot BS3 \cdot \overline{BS4} \cdot \overline{BS5}$$

$$CORR14 = BS0 \cdot \overline{BS1} \cdot BS2 \cdot BS3 \cdot \overline{BS4} \cdot \overline{BS5}$$

$$CORR15 = BS0 \cdot BS1 \cdot \overline{BS2} \cdot BS3 \cdot \overline{BS4} \cdot \overline{BS5}$$

Para obtener estos datos es suficiente utilizar una compuerta NAND con seis entradas.

Se puede observar que cuando todos los bits recibidos son correctos, los bits de síndrome valen 1 lógico y, consecuentemente, todos los bits CORRn valen 1 lógico también, por lo que los bits recibidos deben ser respetados.

Para corregir el bit erróneo, vamos primero a analizar el comportamiento de la compuerta NOR EXCLUSIVO. Es fácil comprobar que con esta compuerta podemos modificar uno de los datos de entrada si el otro dato lo dejamos fijo con un valor 0 lógico de la siguiente forma:

```
A ---\\‾‾‾\    B
      ))   )o---
0 ---//___/
```

A	B
0	1
1	0

De igual forma podemos respetar el bit de dato si el otro lo dejamos fijo con un valor de 1 lógico:

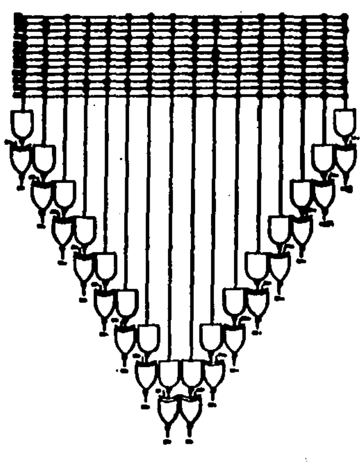
```
A ---\\‾‾\     B
       ))  )o---
1 ---//__/
```

A	B
0	0
1	1

Aplicando estas dos propiedades del NOR EXCLUSIVO es que podremos llevar a cabo la corrección de un dato en caso de ser necesario, utilizando como datos de entrada a la compuerta los bits de corrección CORRn con su respectivo bit de dato recibido BDnr, como se muestra a continuación:

```
CORRn ---\\‾‾\
           ))  )--- BDn
BDnr  ---//__/
```

El diagrama siguiente muestra la etapa correctora completa:



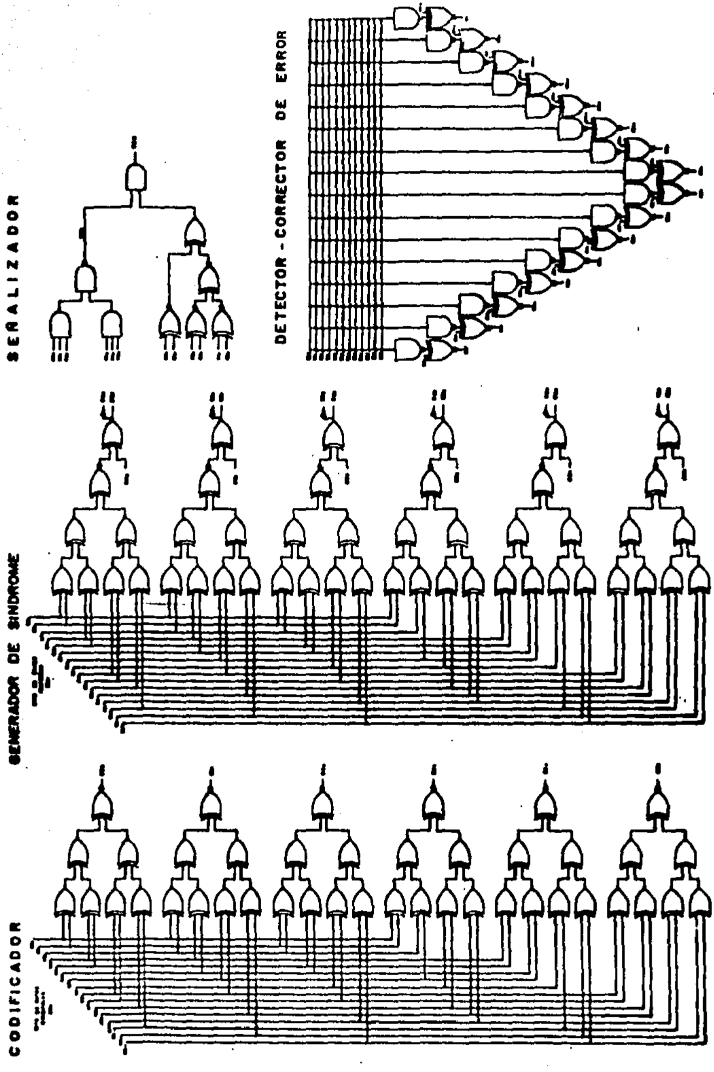


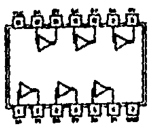
DIAGRAMA GENERAL

b) CIRCUITOS.

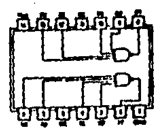
El siguiente paso es encontrar los elementos necesarios para armar el circuito y hacer el diseño de los circuitos impresos adecuados.

Utilizaremos los circuitos integrados que a continuación se mencionan y cuyos diagramas también presentamos:

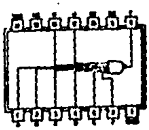
SN7404



SN7420

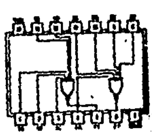


SN7430

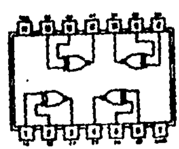


SN74LS136

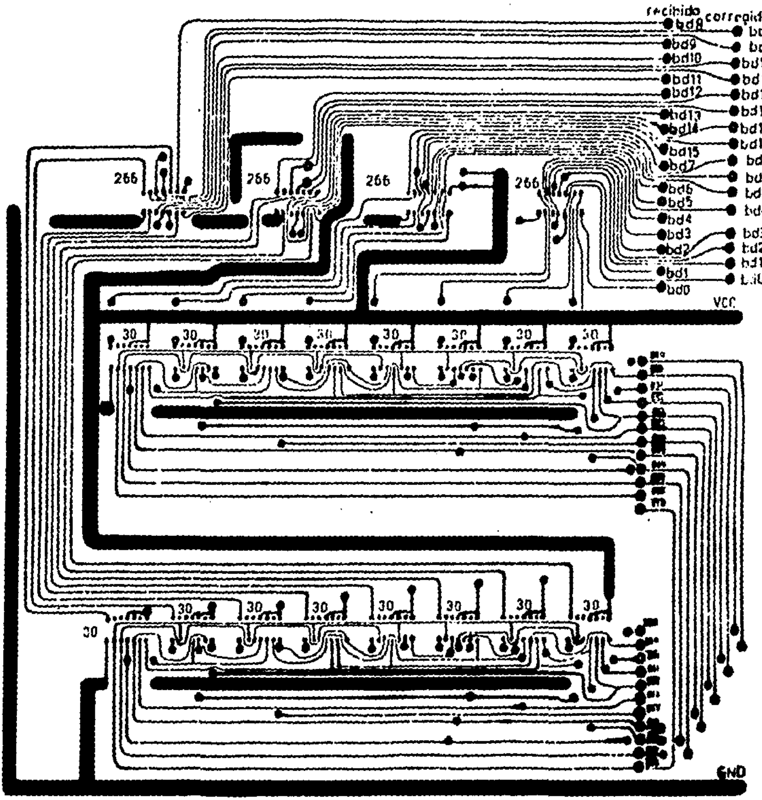
SN74S260



SN74LS266

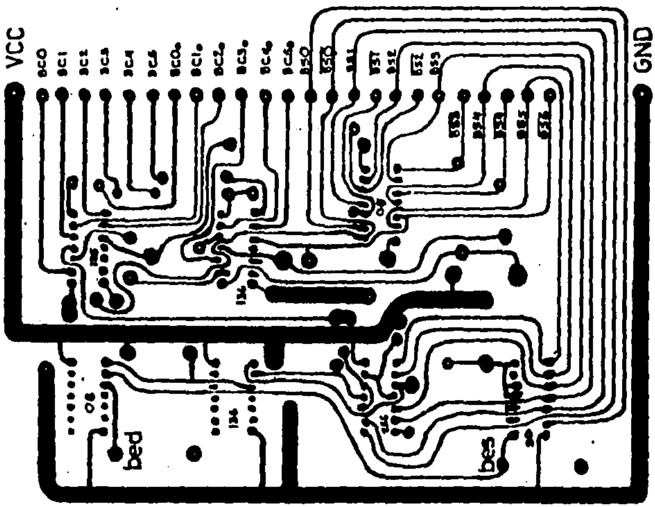


Una vez definidos los circuitos integrados a utilizar, se diseñaron los siguientes circuitos impresos para cada etapa:

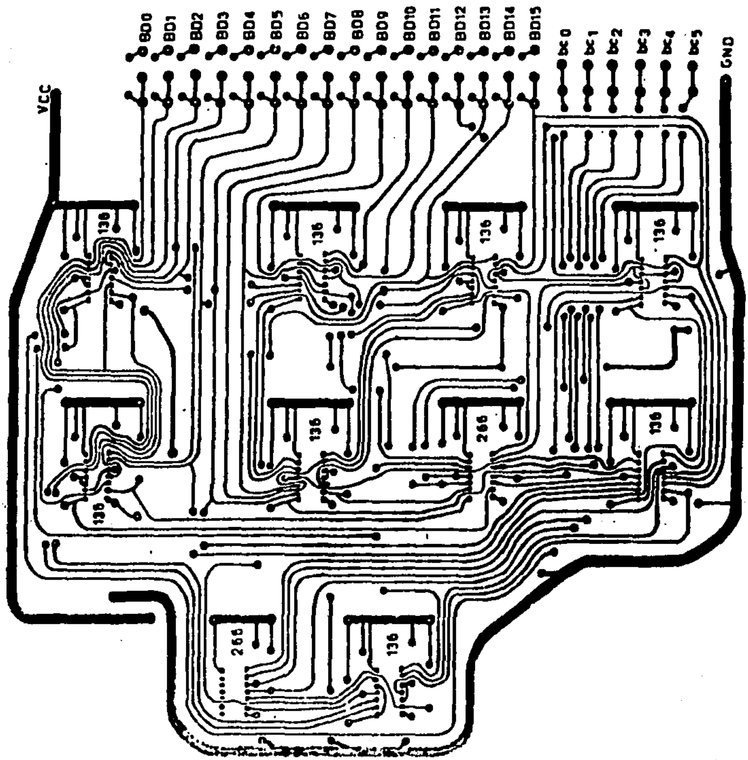


GENERADOR DE SINDROME - DETECTOR Y CORRECTOR DE ERROR

SEÑALIZADOR DE ERROR



CODIFICADOR



CAPITULO V.- PRUEBAS Y RESULTADOS.

Con el fin de mostrar la eficiencia del circuito que hemos diseñado, se hicieron una serie de pruebas y cuyos resultados permitieron corroborar el buen funcionamiento del mismo.

El primer problema con el que nos topamos para probar un circuito de este tipo, es poder generar los errores en el momento deseado, cuestión que en condiciones normales no es posible puesto que la ocurrencia de errores es aleatoria. Por esta razón nos vimos en la necesidad de colocar un interruptor de tres posiciones para cada bit de datos y de chequeo, para tener la posibilidad de modificarlos y simular los errores deseados para las pruebas.

A continuación se presenta una tabla que resume los resultados de las pruebas y con los cuales queda demostrado el buen funcionamiento del circuito:

T A B L A   D E   R E S U L T A D O S

Bits de dato Modificados BDm	Bits de Síndrome 0	1	2	3	4	5	BES	BED	Corrige?	Clave de texto
$BD_{trans}=BD_{rec}$	1	1	1	1	1	1	0	0	NO	@
BD0	0	0	1	0	1	1	1	0	SI	*
BD1	0	1	0	0	1	1	1	0	SI	*
BD2	1	0	0	0	1	1	1	0	SI	*
BD3	0	0	1	1	0	1	1	0	SI	*
BD4	0	1	0	1	0	1	1	0	SI	*
BD5	1	0	0	1	0	1	1	0	SI	*
BD6	1	0	1	0	0	1	1	0	SI	*
BD7	1	1	0	0	0	1	1	0	SI	*
BD8	0	0	1	1	1	0	1	0	SI	*
BD9	0	1	0	1	1	0	1	0	SI	*
BD10	0	1	1	0	1	0	1	0	SI	*
BD11	1	0	1	0	1	0	1	0	SI	*
BD12	1	1	0	0	1	0	1	0	SI	*
BD13	0	1	1	1	0	0	1	0	SI	*
BD14	1	0	1	1	0	0	1	0	SI	*
BD15	1	1	0	1	0	0	1	0	SI	*
BC0	0	1	1	1	1	1	1	0	NO	V
BC1	1	0	1	1	1	1	1	0	NO	V
BC2	1	1	0	1	1	1	1	0	NO	V
BC3	1	1	1	0	1	1	1	0	NO	V
BC4	1	1	1	1	0	1	1	0	NO	V
BC5	1	1	1	1	1	0	1	0	NO	V

CONTINUACION 1

Bits de dato Modificados BDm(s)	Bits de Síndrome 0	1	2	3	4	5	BES	BED	Corrige?	Clave de texto
0-1	1	0	0	1	1	1	1	1	NO	"
0-2	0	1	0	1	1	1	1	1	NO	"
0-3	1	1	1	0	0	1	1	1	NO	"
0-4	1	0	0	0	0	1	1	1	NO	"
0-5	0	1	0	0	0	1	1	1	NO	"
0-6	0	1	1	1	0	1	1	1	NO	"
0-7	0	0	0	1	0	1	1	1	NO	"
0-8	1	1	1	0	1	0	1	1	NO	"
0-9	1	0	0	0	1	0	1	1	NO	"
0-10	1	0	1	1	1	0	1	1	NO	"
0-11	0	1	1	1	1	0	1	1	NO	"
0-12	0	0	0	1	1	0	1	1	NO	"
0-13	1	0	1	0	0	0	1	1	NO	"
0-14	0	1	1	0	0	0	1	1	NO	"
0-15	0	0	0	0	0	0	1	1	NO	"
1-2	0	0	1	1	1	1	1	1	NO	"
1-5	0	0	1	0	0	1	1	1	NO	"
1-8	1	0	0	0	1	0	1	1	NO	"
1-11	0	0	0	1	1	0	1	1	NO	"
1-14	0	0	0	0	0	0	1	1	NO	"
2-3	0	1	0	0	0	1	1	1	NO	"
2-6	1	1	0	1	0	1	1	1	NO	"

CONTINUACION 2

Bits de dato Modificado BDm(s)	Bits de Síndrome 0	1	2	3	4	5	BES	BED	Corrige?	Clave de texto
2-9	0	0	1	0	1	0	1	1	NO	"
2-12	1	0	1	1	1	0	1	1	NO	"
2-15	1	0	1	0	0	0	1	1	NO	"
3-4	1	0	0	1	1	1	1	1	NO	"
3-7	0	0	0	0	1	1	1	1	NO	"
3-10	1	0	1	0	0	0	1	1	NO	"
3-13	1	0	1	1	1	0	1	1	NO	"
4-5	0	0	1	1	1	1	1	1	NO	"
4-8	1	0	0	1	0	0	1	1	NO	"
4-11	0	0	0	0	0	0	1	1	NO	"
4-14	0	0	0	1	1	0	1	1	NO	"
5-6	1	1	0	0	1	1	1	1	NO	"
5-9	0	0	1	1	0	0	1	1	NO	"
5-12	1	0	1	0	0	0	1	1	NO	"
5-15	1	0	1	1	1	0	1	1	NO	"
6-7	1	0	0	1	1	1	1	1	NO	"
6-10	0	0	1	1	0	0	1	1	NO	"
6-13	0	0	1	0	1	0	1	1	NO	"
7-8	0	0	0	0	0	0	1	1	NO	"
7-11	1	0	0	1	0	0	1	1	NO	"
7-14	1	0	0	0	1	0	1	1	NO	"
8-9	1	0	0	1	1	1	1	1	NO	"
8-12	0	0	0	0	1	1	1	1	NO	"

CONTINUACION 3

Bits de dato Modificado BDm(s)	Bits de Sindrome 0	1	2	3	4	5	BES	BED	Corrige?	Clave de texto
8-15	0	0	0	1	0	1	1	1	NO	"
9-10	1	1	0	0	1	1	1	1	NO	"
9-13	1	1	0	1	0	1	1	1	NO	"
10-11	0	0	1	1	1	1	1	1	NO	"
10-14	0	0	1	0	0	1	1	1	NO	"
11-12	1	0	0	1	1	1	1	1	NO	"
11-15	1	0	0	0	0	1	1	1	NO	"
12-13	0	1	0	0	0	1	1	1	NO	"
13-14	0	0	1	1	1	1	1	1	NO	"
14-15	1	0	0	1	1	1	1	1	NO	"

CLAVES DE TEXTO:

- **@ --- Funciona para BD=0 ó BD=1. No hay errores en la transmisión.**
- *** --- Enciende BES y corrige el error.**
- **V --- Enciende BES pero no modifica BD ni BC ya que se necesitaría una matriz de corrección para BC -Bits de Chequeo-.**
- **" --- Enciende SEF y BED. No corrige los bits erróneos.**

C O N C L U S I O N E S

Se ha demostrado que la información digital puede ser protegida fácilmente contra el ruido en un canal de transmisión y que dicha protección es tanto más costosa cuanto la seguridad requerida para la transmisión es mayor.

La información puede ser codificada tanto cuando se transmite en forma de palabras de longitud fija, como cuando es una secuencia continua de digitos sin estructura alguna.

Para proteger la información, es preferible utilizar "paridad impar" ya que si se tiene una pèrdida total de la información, con este tipo de paridad es posible detectarla mientras que con "paridad par" no lo es.

La utilización de este tipo de circuito puede rendir muy buenos resultados en aplicaciones con microprocesadores que se comuniquen con grandes computadoras a travès de un canal telefónico, por ser èste un canal ruidoso por excelencia y por el creciente número de operaciones que pueden realizarse de esta forma.

Mientras màs pequeña sea la palabra que se desea proteger, màs caro resulta protegerla, o sea, la relación entre los bits de redundancia y los bits de información es mayor.

El tiempo de retransmisión se ve dràsticamente disminuido con el uso de este tipo de circuito, ya que todos los errores sencillos se corrigen en el lado receptor y no causan retransmisión como ocurre con la mayoria de los sistemas existentes.

Finalmente, consideramos haber cumplido con un fin didàctico al elaborar este trabajo y nos sentiremos satisfechos si alguna persona se beneficia con èl.

B I B L I O G R A F I A

- ERROR DETECTION AND CORRECTION CODES. Peterson and Weldon. Ed Prentice-Hall. 1978.

- MICROPROCESSOR APPLICATIONS HANDBOOK. Scott, David F. Ed. McGraw-Hill. 1982.

- MICROCOMPUTER-BASED DESIGN. Peatman, John B. Ed. McGraw-Hill. 1977.

- DATA COMMUNICATIONS, Facilities, Networks and Systems Design. Doll, Dixon R. Ed. Wiley-Interscience. 1978.

- TELECOMMUNICATION SYSTEMS ENGINEERING. Lindsey, William C. and Simon, Marvin K. Ed. Prentice-Hall. 1973.

- DIGITAL ELECTRONICS FOR SCIENTISTS. Malmstadt-Enke. Ed. W. A. Benjamin. 1969.

- ALGEBRAIC CODING THEORY. Berlekamp, Elwyn. Ed. McGraw-Hill. 1968.

- INFORMATION AND CODING THEORY. Ingels, Franklin. Ed. Intext Educational Publishers. 1971.

- FAIRCHILD SEMICONDUCTOR The TTL Applications Handbook. Error detection/correction. August 1973.

- ENCYCLOPEDIA OF COMPUTER SCIENCE. Edited by Ralston, Anthony. 1976.

- TELECOMMUNICATIONS AND THE COMPUTER. Martin, James. Ed. Prentice-Hall. 1969.

- TELECOMMUNICATION SYSTEM ENGINEERING. Analog and Digital Network Design. Freeman, Robert L. CCITT. 1978.

- BASICS OF DATA COMMUNICATION Electronics Book Series. Karp, Harry R. 1968.

- COMPUTER DESIGN REVIEW. Penn Well Publication. Dec. 1981 to Feb. 1982.

- B6800 REFERENCE MANUAL. Burroughs Corp.

- SDK-85 USER'S MANUAL. Intel. July 1977.

- TRANSMISION DE DATOS. ESIME. Robledo, Cornelio S. 1977.

- TRANSMISION DE DATOS. AMICEE - ESIME. Curso de extensión Profesional sobre transmisión de datos. 1977.

- APUNTES DE CURSO DE ACTUALIZACION U.N.A.M. Centro de Educaciòn Continua. La Electrónica en las comunicaciones: Señales Digitales en presencia de ruido. Kuhlmann, Federico R. 1977.

- PRINCIPLES OF COMMUNICATION SYSTEMS. Taub and Schilling. Ed. International Student. 1971.

- MANUAL DE LA FAMILIA TTL. Texas Instruments.