

117  
Zejem



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

FACULTAD DE INGENIERÍA

MULTIPLEXAJE DIGITAL CON  
RELLENADO DE BITS

TESIS PROFESIONAL

Que para obtener el Título de:  
INGENIERO MECANICO ELECTRICISTA

Presenta:  
DOMINGO ABEL LOPEZ VARGAS

Director de tesis:  
ING. MARIO A. IBARRA PEREYRA

MEXICO, D. F.

1995

FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**Dedico este trabajo a todas aquellas personas  
que fueron afectadas por la realización de esta tesis: en especial a mis padres,  
a mis hijos Edgar Abel y Emanuel y otras personas que no soportaron...**

**A Cronos que nos presiona, pero que nos ubica.**

**Agradezco de manera infinita la valiosa ayuda de mi director y de todos mis  
compañeros que me apoyaron.**

## **INDICE**

| Capítulo   | Pag. |
|--|------|
| Capítulo I.- Introducción.                                   | 1    |
| Capítulo II.- Teoría del multiplexaje.                       | 5    |
| Capítulo III.- Planteamiento del problema.                   | 14   |
| Capítulo IV.- Diseño del Sistema de Almacenamiento Elástico. | 17   |
| 1.- Diseño del generador de palabras pseudoaleatorias.       | 17   |
| 2.- Diseño del Sistema Transmisor.                           | 21   |
| 3.- Diseño del Sistema Receptor.                             | 30   |
| Capítulo V.- Construcción y Pruebas.                         | 40   |
| 1.- Construcción.  | 40   |
| 2.- Relación de pruebas.                                     | 44   |

|  |           |
|--|-----------|
| <b>Capítulo VI.- Evaluación y Conclusiones.</b>  | <b>52</b> |
| <b>Apéndice.- Generadores de Señal de Prueba para los<br/>Sistemas de Transmisión Digital.</b> | <b>54</b> |
| <b>Bibliografía.-</b>  | <b>59</b> |

## **CAPITULO I. - INTRODUCCION**

La existencia de cualquier ser vivo es de constantes esfuerzos por satisfacer necesidades. Los seres microscópicos y los unicelulares solo tienen dos: alimentarse y reproducirse, pero entre más complejo es el organismo, más diversos son sus problemas y mayores sus esfuerzos para resolverlos; y ya sabemos a donde va a parar este razonamiento: al ser humano que requiere alimentarse, vestirse, reproducirse, defenderse y encontrar satisfactores para un sinnúmero de necesidades que se ha inventado, muchas de ellas ilícitas; pero qué le vamos a hacer; el hecho es que hay que resolver problemas.

Lo anterior desemboca en esbozar la metodología que se puede usar para resolver los mencionados problemas.

Pensemos en que el problema es de tal naturaleza que se puede poner por escrito en un papel. Lo primero que se puede hacer para resolverlo es trabajar de atrás para adelante; esto es, imaginar una solución y ver que pasa si la aplicamos al problema. Si la solución propuesta "no encaja" entonces buscamos otra y otra, hasta que "le atinamos".

Otra forma de resolver el problema es dividirlo en muchos pedacitos o mejor dicho, en problemas más simples, cuya solución evidentemente se encuentre a nuestro alcance.

Una vez que aparentemente hemos encontrado la solución en el papel, procedemos a simular su implementación, extrapolando las posibles consecuencias de su aplicación. Si no encontramos consecuencias funestas, entonces procedemos a realizar físicamente la solución al problema.

Una de las necesidades que se reconocen que tienen todos los seres vivos que se mueven, es la de la comunicación en diversos grados de complejidad. Vease por ejemplo una larga fila de hormigas que van y vienen entre el hormiguero y algún insecto muerto; si nos fijamos con atención veremos que cada vez que dos de ellas se encuentran tocan sus antenas pero no lo hacen con las que van en el mismo sentido. Otro caso interesante es el canto de las aves, que generalmente se considera

como un signo de alegría, pero que en realidad es un canto bélico para indicar a otras aves que no se atrevan a invadir sus dominios.

De esta forma, el ser humano, es el ser vivo que tiene más necesidad de comunicarse y el que tiene más problemas para lograrlo, dada la inmensa variedad de mensajes y la gran cantidad de "receptores" a los que se desea entregar la información.

Para resolver el gran problema de la comunicación, el ser humano se ha tardado ya varios miles de años, ya que curiosamente el problema crece y se diversifica con el paso del tiempo y en vez de irnos aproximando a la solución definitiva, tenemos que hacer cada vez mayores esfuerzos para "sacar la cabeza fuera del agua", esto es, para desarrollar técnicas nuevas que satisfagan las crecientes necesidades.

Hay detalles que complican mucho el problema de la comunicación: los mensajes deben llegar íntegros, lo más pronto posible, con el mínimo costo, deben llegar solo a un receptor o deben difundirse, deben recorrer cortas o largas distancias, etc. etc.

Acorde con la metodología establecida párrafos atrás, se ha fraccionado el gran problema en problemas menores; así, se han desarrollado las comunicaciones analógicas y las digitales y dentro de estas últimas, se han implementado procedimientos para digitalizar textos, sonidos e imágenes. Así mismo, se cuenta con procedimientos para enviar varios mensajes digitales simultáneamente por la misma vía sin que se revuelvan (esto también se hace con señales analógicas).

Esta tesis trata de uno de tantos pequeños problemas que se han tenido que resolver para que se pueda realizar la comunicación; tiene que ver con el procedimiento de juntar varios mensajes digitales en una sola línea y separarlos en el otro extremo sin que los bits se revuelvan; técnica conocida como multiplexaje por división de tiempo.

El proceso de multiplexaje por división de tiempo se inicia cuando diversos usuarios entregan sus señales (analógicas o textos) al equipo que las digitaliza y acomoda los bits en un orden adecuado para la transmisión, intercalando bits de identificación para saber a quien corresponde cada paquete de datos.

Si todo el proceso descrito anteriormente es controlado por un solo reloj, todos los dígitos binarios tendrán la misma duración y el receptor o demultiplexor no tendrá problemas graves de sincronía.

Los dolores de cabeza empiezan cuando los usuarios ejecutan por su cuenta el proceso de digitalización usando sus propios relojes y entregan al sistema multiplexor sus señales ya digitalizadas; ya que aunque se exige que todos los relojes tengan la misma frecuencia, nunca podrán cumplir esta condición, con la consecuencia de que algunas señales binarias llegarán al multiplexor con un exceso de bits/seg y otras con menos bits/seg de los requeridos y podría darse el caso de que una misma señal binaria presentara alternativamente las dos condiciones debido a la inestabilidad de su reloj.

para solucionar estos inconvenientes se ha desarrollado el concepto de "almacenamiento elástico", que se explicará a continuación.

Un almacén elástico o "Buffer elástico" es una memoria en la que durante un periodo de tiempo se guardan los bits que van llegando, provenientes de un digitalizador. Si al final de ese periodo, el número de bits almacenados no alcanza un mínimo preestablecido, el sistema agregará los bits "de relleno" necesarios para lograr tal cantidad; por otra parte, si el número de bits almacenados rebasa el valor establecido, el sistema guardará los bits sobrantes para transmitirlos en la siguiente trama. En todos los casos el transmisor informa al receptor cuantos bits ha quitado o agregado, este procedimiento se repite indefinidamente

El problema de desarrollar un buffer elástico se puede partir en dos problemas menores: cuando las señales binarias tienen menos bits/seg que el valor establecido y cuando las señales tienen más bits/seg de lo estipulado.

En el primer caso, solo hay que rellenar con ceros ( o con unos) los lugares faltantes y avisar al receptor para que los retire y vuelva a "estirar" los bits de información a su anchura original. Esto se llama relleno positivo.

El segundo caso, conocido como relleno negativo, es muy desfavorable, ya que cuando el buffer se llene con los bits sobrantes, si no se detiene la entrada de datos, el sistema empezará a perder información.

En esta tesis, se va a manejar el problema de relleno positivo, que es generalmente usado en equipos comerciales, para lo cual se van a diseñar tres circuitos: un generador de señal binaria pseudoaleatoria, un buffer elástico y un circuito que retire los bits de relleno y reconstruya la señal binaria tal como era originalmente.

Además del diseño, se construirán los circuitos, consignando en este trabajo, los diagramas lógicos y eléctricos, las pruebas, los resultados obtenidos y las conclusiones.

El asunto de multiplexar varias señales rellenas solo se va a tratar teóricamente, ya que su implementación no presenta algún atractivo desde el punto de vista de principios fundamentales y requiere de mucha labor manual repetitiva.

## **CAPITULO II**

### **TEORIA DEL MULTIPLEXAJE.**

El multiplexaje es un proceso que permite la transmisión de varias señales por la misma vía sin que se interfieran unas a otras.

Esta técnica se utiliza ampliamente en telefonía y redes digitales, ya que nos permite cobrar a muchos usuarios por el uso de un único canal de comunicación.

Además de los beneficios económicos directos, el multiplexaje nos permite realizar grandes expansiones en la capacidad de los sistemas para ofrecer servicios a un número cada vez mayor de usuarios.

Este proceso, que en español se conoce como multicanalización, se puede llevar a cabo en tres diferentes formas:

**Multiplexaje por división de espacio.-** Cuando las señales ocupan el mismo rango de frecuencias y son emitidas al mismo tiempo, pero viajan por diferentes conductos.

**Multiplexaje por división de frecuencia.-** Cuando las señales son emitidas al mismo tiempo y viajan por el mismo conducto, pero ocupan diferentes rangos de frecuencia.

**Multiplexaje por división de tiempo.-** Cuando las señales ocupan el mismo rango de frecuencias y viajan por el mismo conducto, pero son emitidas en diferentes tiempos.

La técnica que vamos a utilizar en el desarrollo, de este trabajo es la última de las mencionadas anteriormente, por lo que solo de esta se dará la explicación correspondiente.

El multiplexaje por división de tiempo requiere de un proceso de muestreo de las señales que se van a transmitir y de la intercalación de las muestras como se ve en la figura II.1.

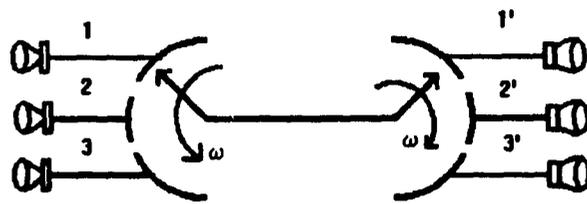


Fig. II.1.- Diagrama conceptual de un sistema T.D.M.

Para que el sistema esquematizado en la figura anterior funcione satisfactoriamente, es necesario que las llaves giratorias tengan una velocidad angular determinada por el teorema del muestreo; de modo que si las señales provienen de aparatos telefónicos, su frecuencia máxima será de 3400 Hz. y las llaves deben girar a más de 6800 rev/seg. En los sistemas comerciales la frecuencia de muestreo es de 8000 muestras/seg. En la figura II.2 se puede ver tres señales analógicas, muestreadas, multiplexadas y listas para transmitirse

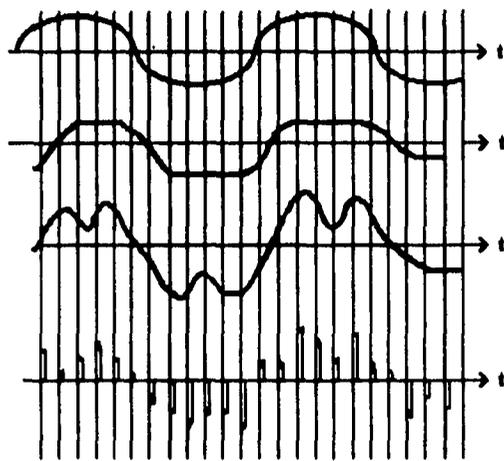


Fig. II.2.- Tres señales analógicas multiplexadas.

Es evidente que un sistema mecánico jamás podrá girar a 8000 revoluciones por segundo, por lo que los sistemas comerciales usan circuitos electrónicos de alta velocidad, quedando el sistema como se muestra en la figura II.3.

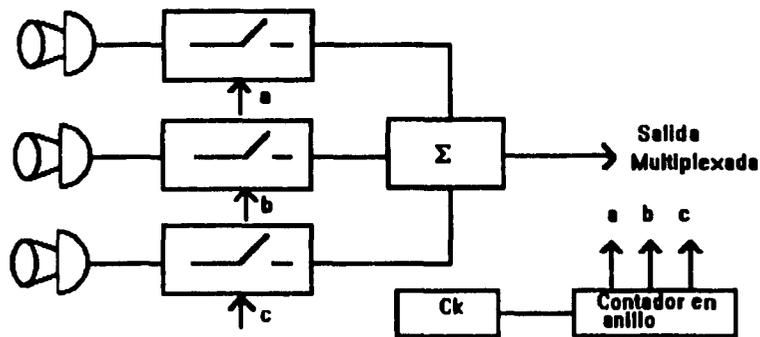
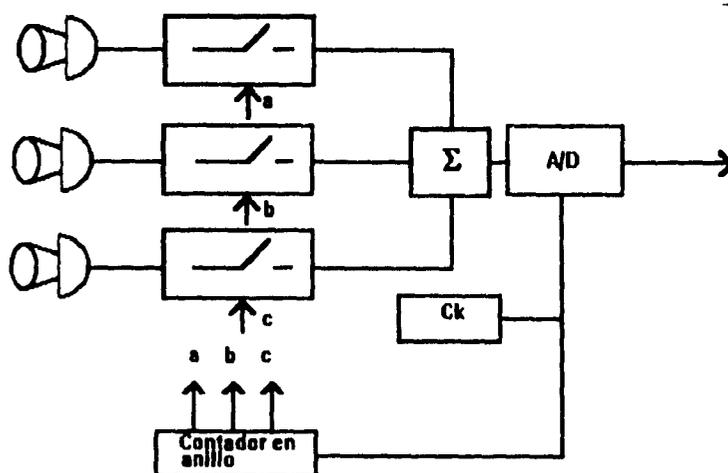


Fig. II.3.- Sistema electrónico de multiplexaje por división de tiempo.

Como se puede apreciar en la figura, los interruptores electrónicos se van cerrando secuencialmente, controlados por los pulsos que salen del contador anular, que son 8000 pulsos/seg. en cada salida, de manera que si vamos a procesar  $n$  señales analógicas, la frecuencia del reloj principal será  $8000n$  ciclos/seg; mismo valor que el número de muestras/seg que salen del sistema.

Las muestras obtenidas del sistema de multiplexaje que estamos analizando son fuertemente afectadas por el ruido; de manera que para darles inmunidad, tenemos que convertirlas a código binario. De esta manera el receptor solo tendrá que decidir si lo que llega es un uno o un cero, tarea mucho más sencilla que la de determinar la altura de la muestra que puede tener un número infinito de valores.

Existen dos posibles posiciones del multiplexor en el sistema que estamos analizando: En un caso, primero se multiplexa y después se hace la conversión a binario y en otro caso, se convierten las señales a binario y luego se multiplexan. En la figura II.4 se ilustra esto.



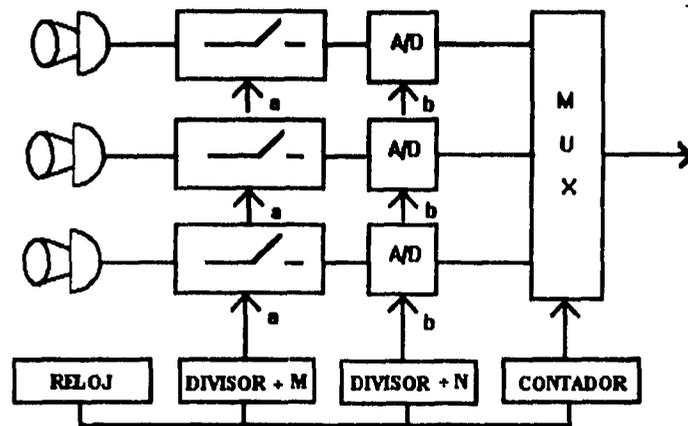


Fig. II.4.- Posibles posiciones del multiplexor en el sistema.

Si lo que se va a transmitir no son señales analógicas, sino textos codificados en binario, el panorama es completamente el mismo ya que si se usa un reloj para todos los codificadores no hay problema y si se usan relojes independientes para los codificadores, tendremos las dificultades ya mencionadas.

El triste panorama presentado anteriormente se puede componer de dos formas:

Primera.- Haciendo que los usuarios del multiplexor sincronicen sus relojes con un reloj maestro que tendría que ser proporcionado por la autoridad en telecomunicaciones de la localidad, o por una agrupación de usuarios.

Segunda.- Implementando un sistema de compensación de velocidades antes de que las diversas señales entren al multiplexor. Este sistema es el que se conoce como almacenamiento elástico o "buffer elástico" y evidentemente es el que vamos a tratar en el resto de este trabajo.

El diagrama de la figura II.5 nos muestra la colocación de los mencionados dispositivos dentro de un sistema de multiplexaje.

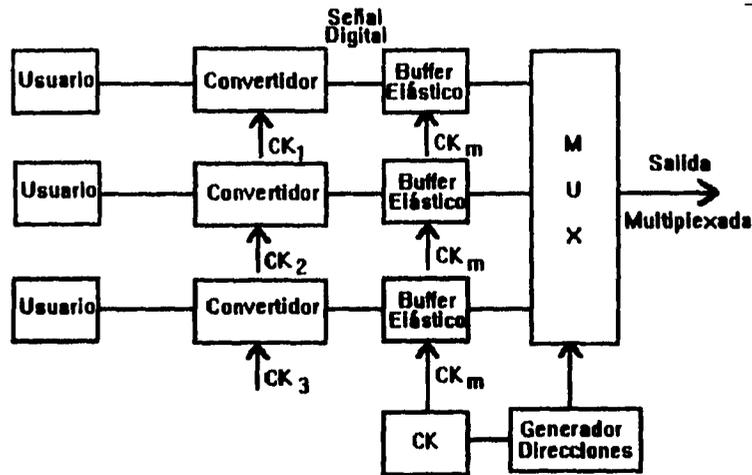


Fig. II.5.- Multiplexor con relleno de bits.

El proceso de relleno, que también se conoce como "Justificación" y que en inglés se dice "bit stuffing" se realiza de la siguiente forma:

Se debe disponer de una unidad de memoria que vaya recibiendo y almacenando los pulsos binarios conforme vayan llegando.

Cada cierto periodo, se cuentan los bits almacenados en la memoria. Si el reloj que los controla tiene la frecuencia nominal, el número de pulsos será óptimo; esto es, que no faltan ni sobran bits, y en este caso no se requieren bits de relleno. Si el reloj trabaja a una frecuencia mayor que la nominal, se recibirá un excedente de bits; en este caso se transmite la cantidad nominal de bits, y los que sobran se guardan para ser transmitidos posteriormente. Si el reloj

trabaja a una frecuencia menor que la nominal, La memoria contendrá al final del periodo menos bits de los esperados; por lo cual será necesario agregar algunos bits para completar el valor correcto y realizar la transmisión.

En todos los casos mencionados en el párrafo anterior, es necesario agregar algunos bits para informar al receptor si ha habido relleno positivo o relleno negativo, cuantos bits se han utilizado para este fin o si no ha habido relleno.

En la figura II.6 se ilustra la secuencia de bits de una porción de un mensaje relleno. La parte (a) ilustra el relleno positivo y la parte (b) el negativo.

En un caso real la frecuencia del reloj que controla a la señal binaria tiene un error muy pequeño con respecto al valor nominal, por lo que se espera que haya un bit de relleno esporádicamente. Por ejemplo una variación de +0.01% en la frecuencia de un reloj de 1200 Hz. Implica que van a llegar 1200.12 bits/seg. o sea que habrá 12 bits excedentes cada 100 segundos, o sea que en cualquiera de los dos casos, el sistema tendrá que usar un bit de relleno cada 8.33 segundos, lo cual no es incómodo; ya que cada paquete llevará n bits de mensaje, Un bit de relleno y dos bits para indicar si hay relleno positivo, negativo o no hay relleno.

En el caso de que, sistemáticamente el mensaje llegue al multiplexor con un exceso de bits. La memoria tendrá que estar almacenando continuamente estos bits sobrantes, por lo que llegará un instante en que la memoria quedará completamente llena y evidentemente no podrá seguir almacenando bits; debido a esto se pueden presentar varias posibilidades

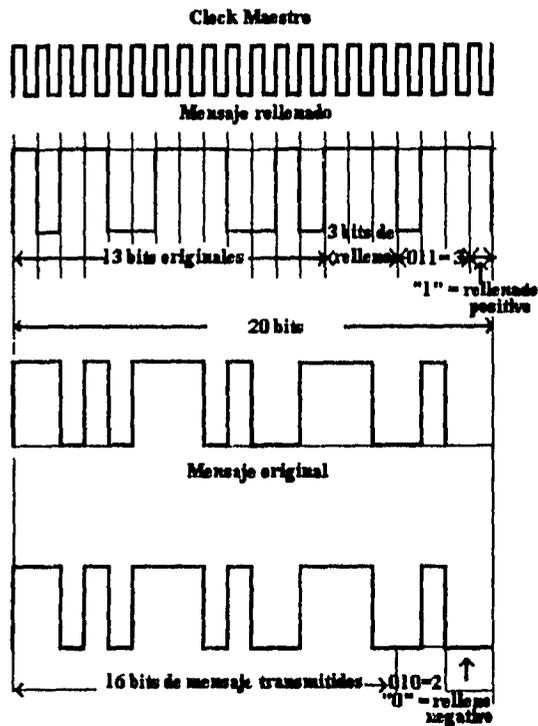


Fig.II.6.- Secuencias binarias con relleno positivo y negativo.

La situación más favorable es cuando el sistema de multiplexaje puede indicarle a la fuente de la señal binaria que ya no siga mandando información, para que le dé tiempo a la memoria de desalojar todos los bits que tiene guardados. Esta situación no es usual en sistemas reales, ya que requiere un canal de ida y otro de retorno entre la señal y el multiplexor.

Otra situación que se puede presentar, es que al llenarse la memoria, simplemente se desechan los bits excentes y se continúa con el proceso de transmisión. Si las señales que estamos multiplexando son señales analógicas,

esta pérdida de bits, implica un "clic" en el audífono y la conversación puede continuar sin graves problemas. Si las señales que estamos multiplexando son textos codificados en binario, los bits desechados implican la pérdida de un símbolo alfanumérico, lo cual es absolutamente inadmisibile.

Por la razón expuesta en el párrafo anterior, el relleno negativo suele resultar inoperante y generalmente no se utiliza en sistemas comerciales.

¿Que podemos hacer para garantizar que nuestro sistema de multiplexaje nunca tenga que efectuar relleno negativo?. La respuesta es simple; basta con asegurarnos de que la frecuencia de reloj del multiplexor siempre sea superior a  $n$  veces la frecuencia nominal de cualquiera de los relojes de las señales que se van a multiplexar; de este modo siempre habrá relleno positivo.

Como se puede ver de las explicaciones anteriores, el proceso de relleno, soluciona los inconvenientes resultantes de recibir señales a velocidades diferentes; sin embargo, por este beneficio hay que pagar el precio, consistente en que la velocidad de salida del multiplexor es mayor que la suma de las velocidades de entrada.

## **CAPITULO III.**

### **PLANTEAMIENTO DEL PROBLEMA.**

En este capítulo se trata de establecer los parámetros básicos que se deben de proporcionar al diseñador de circuitos, para que este pueda especificar los elementos eléctricos del buffer elástico, que deseamos construir.

Así mismo, se pretende esbozar algunas de las posibles soluciones que pueda tener el problema.

En primer lugar, es necesario establecer que el objetivo del proyecto es múltiple. Inicialmente se trata del objetivo de toda tesis, que es el de mostrar la capacidad del autor para resolver problemas de su especialidad. En seguida, se trata de obtener un aparato con propósitos didácticos, o sea, que permita demostrar un principio fundamental determinado en forma académica.

En tercer lugar, anotaremos que no queremos obtener un aparato comercializable; esto implica que no es necesario cumplir con normas técnicas, ya que no se pretende interconectarlos con equipos o redes comerciales.

En concreto: El aparato que se pretende implementar, será utilizado en las prácticas del Laboratorio de Comunicaciones Digitales; primero para mostrarles a los alumnos cual es el principio fundamental y cual es la instrumentación para analizar el circuito y en seguida para comparar la complejidad de este diseño con la de los realizados por los alumnos.

Todo lo anterior nos lleva a que los parámetros de diseño se pueden seleccionar con un amplio grado de libertad, lo cual en cierto modo, facilita un poco las cosas, ya que por ejemplo, no tendremos que enfrentarnos con los problemas tecnológicos derivados del uso de altas velocidades de transmisión y también podremos evitar el uso de circuitos impresos multicapa, requeridos cuando es obligatorio compactar el sistema en un mínimo de volumen.

El problema impuesto por el título de la tesis, se puede dividir inicialmente en dos partes: Transmisor y Receptor.

Siguiendo con la misma filosofía, el diseño del transmisor se puede descomponer en tres problemas menores: en primer lugar, generar varias señales digitales, controladas con relojes ligeramente diferentes; en segundo lugar, es necesario rellenar las señales digitales, para obtener velocidades idénticas y en tercer lugar, hay que multiplexar las señales rellenas para sacarlas por una línea única.

El diseño del receptor requiere realizar los procesos contrarios a los del transmisor; es decir demultiplexar, eliminar el relleno y entregar las señales a los usuarios finales.

Teniendo en cuenta que la etapa de multiplexaje complica innecesariamente la construcción del sistema, se va a prescindir de esta parte del proceso, para centrar toda la atención en el diseño y construcción del buffer elástico, dejando como una práctica de laboratorio, la implementación del multiplexor y el demultiplexor digitales.

En el capítulo anterior, se estableció que existe el relleno positivo y el relleno negativo y se mencionó que se pueden arreglar las cosas para que nunca se presente la segunda posibilidad; esto es, la de que sobren bits. De esta forma procederemos en el planteamiento, especificando que el buffer elástico debe trabajar exclusivamente con relleno positivo.

Un parámetro importante es la velocidad de transmisión del mensaje relleno. Esta la debemos establecer, en un valor tal que, los pulsos sean fácilmente visibles en un osciloscopio estandar, sin que se presente el parpadeo de las bajas frecuencias ni las distorsiones e interferencias usuales en los circuitos de alta velocidad.

Por, lo anterior, se considera que una velocidad final de 25 kbits/seg puede dar resultados satisfactorios, para los fines que se persiguen.

El mensaje original, se supone que debe de provenir de una computadora o de un convertidor analógico digital, pero estos dispositivos, producen trenes de bits completamente aleatorios y por lo tanto, muy difíciles de visualizar en forma estática en un osciloscopio. Para solucionar este inconveniente, se puede obtener la información digital de un generador de palabras pseudoaleatorias, que es lo que se propone como una de las partes del proyecto.

Establecida ya la necesidad de diseñar y construir un generador de palabras binarias pseudoaleatorias, quedan dos parámetros por establecer, que son la longitud de las palabras y la velocidad de transmisión del mencionado generador.

La longitud de las palabras, se deberá seleccionar, para que sea menor de 15 bits, que es lo que más se puede visualizar en un osciloscopio y la velocidad de transmisión se escogerá de tal forma que sea menor de 25 kbits/seg, para permitir bits de relleno y bits de control, suficientes para la operación del receptor.

La forma de enlazar el transmisor y el receptor, será por medio de tres hilos conductores, que corresponderán a la señal rellena, al reloj maestro y a los pulsos de carga.

La tecnología a usar para la implementación de los circuitos queda a criterio del diseñador, ya que se prevee la posibilidad de usar componentes TTL y componentes MOS. Únicamente se excluye la posibilidad de usar transistores discretos.

El montaje de los circuitos, se deja también al criterio del diseñador, excluyéndose la posibilidad de usar el montaje en las tabletas experimentales protoboard.

Finalmente, la alimentación de energía (corriente directa) se realizará con las fuentes de que se dispone, eludiendo la necesidad de diseñar y construir tales equipos.

## CAPITULO IV

### DISEÑO DEL SISTEMA DE ALMACENAMIENTO ELASTICO.

En este capítulo se presentarán los diagramas de bloques, diagramas lógicos y diagramas de circuitos del sistema que se pretende desarrollar; así como un reducido número de cálculos, intercalando las explicaciones necesarias y los comentarios que se consideren pertinentes de manera que no resulte una simple y fría colección de figuras y cálculos.

#### 1.- Diseño del generador de palabras pseudoaleatorias.

Una palabra pseudoaleatoria es una cadena de bits que aparentemente no tienen una secuencia matemáticamente definida y que se repite periódicamente, mientras el circuito esté energizado.

Una palabra pseudoaleatoria típica es la siguiente:

1 1 1 1 0 0 0 1 0 0 1 1 0 1 0

Como se puede ver, aparentemente no hay una ley que permita deducir, qué bit va en cada una de las posiciones. En realidad sí la hay, pero solo una persona con un C.I. de 200 o alguien que sepa como se generó la palabra puede predecir el valor de cada bit.

El prefijo "pseudo" se debe entonces a dos razones: A que sí hay una ley de variación y a que la misma palabra se repite continuamente.

Se conocen tres formas de generar palabras pseudoaleatorias:

Primera.- Grabar la palabra en una memoria digital (RAM, ROM, PROM, etc.) y con un contador darle las direcciones para leerla en forma secuencial, como se ve en la figura IV.1. Con esta técnica se puede lograr cualquier longitud de palabra.

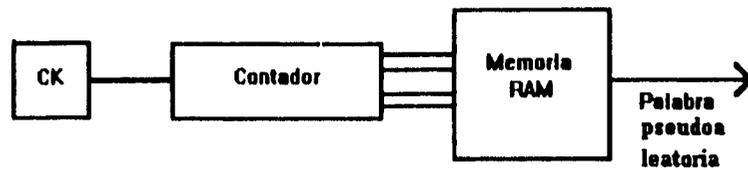


Fig. IV.1.- Generador de palabras pseudoaleatorias.

Segunda.- Poner los bits de la palabra como niveles de voltaje fijos a la entrada de un multiplexor tal como el 74150 y por medio de un contador irle dando las direcciones para que los niveles lógicos de entrada vayan presentándose consecutivamente a la salida. Esto se puede ver en la figura IV.2. Con esta técnica se puede cambiar la palabra con mayor facilidad que si estuviera grabada en una memoria.

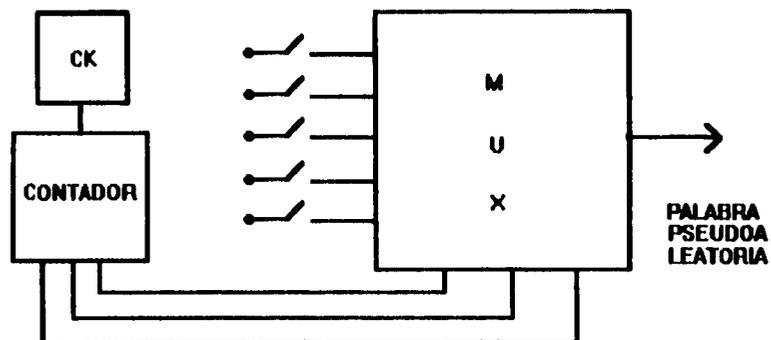


Fig. IV.2.- Otro generador de palabras pseudoaleatorias.

Tercera.- Se puede usar un registro de corrimiento de n secciones retroalimentado con una compuerta X-OR, como se ve en la figura IV.3.

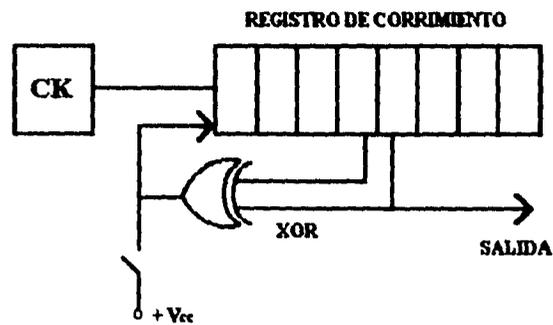


Fig. IV.3.- Otro generador de palabras pseudoaleatorias

Este procedimiento genera palabras cuya longitud es  $2^n-1$ ; por ejemplo, un registro de cuatro secciones genera la palabra de 15 bits que se anotó al principio de este capítulo.

La secuencia de la palabra cambiará si conectamos la compuerta X-OR a diferentes salidas del registro de corrimiento.

Con este circuito lógico no se puede generar cualquier palabra que se nos ocurra, ya que la secuencia de bits resultante está determinada por una fórmula en función de la longitud del registro y de los puntos de conexión de la compuerta X-OR. En el apéndice de esta tesis se anotan algunas de las secuencias pseudoaleatorias que se pueden obtener y sus propiedades lógicas.

Es evidente que la velocidad de salida de este generador de palabras en bits/seg. es igual a la frecuencia del reloj que lo controla.

Teniendo en cuenta las características lógicas de el generador de la figura IV.3, se procederá a diseñar un circuito basado en este principio fundamental.

Como generador de reloj se utilizará el elemento en el que todos piensan cuando se proponen construir un oscilador de onda cuadrada de baja frecuencia. Nos referimos al omnipresente LM-555.

El siguiente componente que vamos a seleccionar es un registro de corrimiento de cuatro secciones, entrada serie salida paralelo, por lo que se escoge el 74164, que es de 8 secciones. En el catálogo hay registros de cuatro secciones pero el 74164 es más fácil de usar.

La compuerta X-OR no es difícil de seleccionar, ya que hay pocas opciones, por lo que se escoge el 7486.

El último elemento que se incluye en el circuito es un interruptor para darle un pulso de arranque al circuito de manera que empiece a generar sus palabras en el momento que lo necesitemos, ya que el dispositivo no arranca solo al momento de energizarlo.

Los valores de los componentes R y C del reloj se calcularán posteriormente, quedando el circuito del generador de palabras pseudoaleatorias como se ve en la figura IV.4.

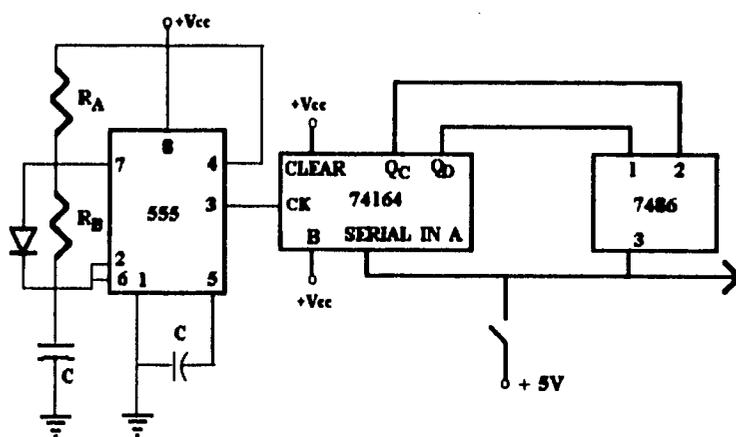


Fig. IV.4.- Generador de palabras pseudoaleatorias.

## 2.- Diseño del sistema transmisor.

El siguiente elemento que debemos definir es la estructura de las palabras rellenas que vamos a transmitir. En la figura IV.5 podemos ver que la palabra rellena es de  $K$  bits, de los cuales  $M$  son de mensaje,  $N$  son de relleno y  $P$  son de control; estos últimos nos van a indicar cuantos bits de relleno se agregaron.

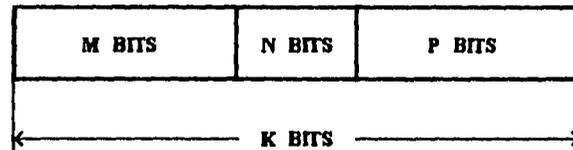


Fig. IV.5.- Estructura de la palabra rellena.

Para poder formar esta palabra, es necesario recibir los  $M$  bits de mensaje y almacenarlos temporalmente en un dispositivo de memoria en tanto se calculan y se generan los bits de relleno y de control. Resulta pues evidente que todo este conjunto de unos y ceros, estará disponible en un arreglo en paralelo, que habrá que convertir a serie para su posterior transmisión.

El circuito lógico que puede hacer la conversión de paralelo a serie es un multiplexor. Cabe hacer notar que un multiplexor se puede construir de cualquier número de entradas, usando compuertas lógicas; sin embargo este no es el objetivo de la tesis, por lo cual utilizaremos un multiplexor comercial, tal como el 74150 que es de 16 entradas y una salida. En función de lo anterior, nuestra palabra rellena tendrá  $K = 16$  bits en total.

Como siguiente paso, se propone que se cuente con cuatro bits de control, de los cuales el último será siempre cero para propósitos de sincronización y ajuste de tiempos. De esta manera, los 12 bits sobrantes llevarán el mensaje y el relleno.

Teniendo en cuenta que de los 4 bits de control solo utilizaremos tres, el número máximo permisible de bits de relleno será de siete y el mínimo de cero.

Según se estableció en el planteamiento del problema, la salida del buffer debe ser de 25 Kbits/seg. por lo que la duración de nuestra palabra de 16 bits es de  $16/25000 = 640 \mu\text{seg}$ ; en este lapso podremos procesar un máximo de 12 bits de mensaje y un mínimo de 5 bits de mensaje; por lo tanto podremos calcular ya las velocidades máxima y mínima de la fuente de datos binarios, como se anota en seguida:

Si entran M bits de mensaje, la velocidad de transmisión es la relación del número de bits al periodo de la palabra; esto es:  $V.T. = M/640 \times 10^{-6}$  bits/seg.

Para el caso en que entren 12 bits de mensaje:

$$V.T_{\max} = 12/640 \times 10^{-6} = 18750 \text{ bits/seg.}$$

Para el caso en que entren 5 bits de mensaje:

$$V.T_{\min} = 5/640 \times 10^{-6} = 7812 \text{ bits/seg.}$$

Con la estructura de la palabra que hemos propuesto, es posible rellenar señales binarias que varíen entre las dos velocidades calculadas, lo que le da al sistema una enorme elasticidad. Para nuestro caso, se propone que el generador de palabras pseudoaleatorias opere a una velocidad cercana a los 15000 bits/seg. y se dejará que la inestabilidad inherente del LM555 nos proporcione pequeñas variaciones de velocidad de transmisión del generador de palabras pseudoaleatorias.

Para acomodar temporalmente los bits de mensaje que van llegando y los bits de relleno, se propone un registro de corrimiento de 12 bits, entrada serie y salida paralelo y como no hay en el catálogo un circuito de estas características, se hará uso de 2 circuitos 74164 conectados en cascada, usando las 8 salidas del primero y las 4 primeras salidas del segundo. El reloj que controla el flujo de los datos en estos registros es el mismo con el que se genera el mensaje binario.

Como el flujo de datos es ininterrumpido, necesitamos almacenar los bits recibidos, así como los bits de relleno y de sincronía, en tanto que el

multiplexor los puede procesar y para esto utilizaremos circuitos biestables de los conocidos como "latch", tales como el 74116.

El último elemento que vamos a incluir en la cadena, es un contador descendente, controlado por el reloj del mensaje binario y que empiece a contar a partir de 12, de modo que al terminar el ciclo de 640  $\mu$ seg, nos dirá cuantos bits de relleno se van a transmitir.

Todos los elementos que se han mencionado se interconectarán como se muestra en la figura IV.6.

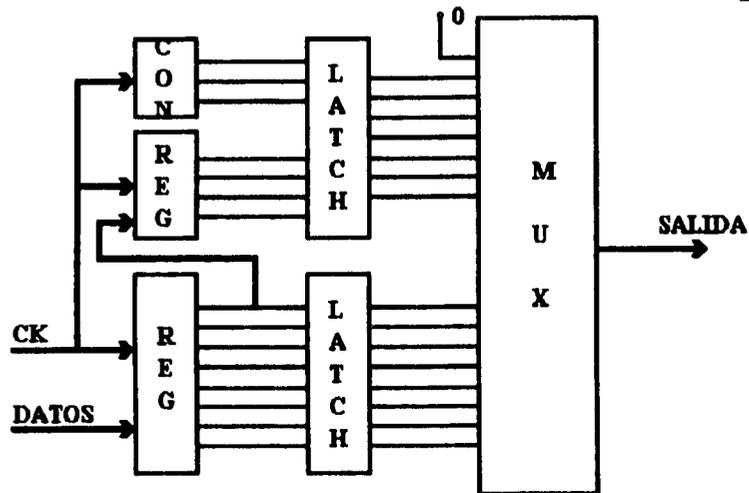


Fig. IV.6.- Componentes básicos del buffer elástico.

Podemos ahora explicar brevemente el funcionamiento del sistema de la figura anterior.

El reloj del generador de palabras pseudoaleatorias controla dos procesos dentro del buffer elástico: Por una parte activa al contador descendente (74193) que nos va a indicar el número de bits de relleno, ya que la cuenta empieza en 12 y se decrementa una unidad por cada bit de mensaje que llega al sistema. Por otra parte, controla el desplazamiento de los bits del

mensaje dentro del registro de corrimiento (74164). Al terminar el periodo de 640  $\mu$ seg. tendremos ya estructurado el byte conteniendo M bits de mensaje, 12-M bits de relleno, 3 bits de control y un cero al final. Durante este último cero, el byte es transferido a los latches (74116) y en seguida se restablece a ceros el registro de corrimiento y el contador se regresa a su posición original de 12, para iniciar la formación de un nuevo byte. Mientras esto ocurre, los bits almacenados en los latches en el ciclo anterior son transferidos a la línea por el multiplexor (74150).

El siguiente problema que tenemos que resolver es el de la generación de toda una serie de pulsos de control para iniciar la operación de los dispositivos en el momento adecuado. El principal obstáculo que debemos eliminar es la posibilidad de que haya colisión entre un bit de mensaje y una orden de restablecimiento, ya que si ambos pulsos llegan simultáneamente a un circuito, este obedecerá la orden de restablecimiento y perderá el bit de mensaje.

En la figura IV.7 se presenta el diagrama de bloques del generador de pulsos de control para el buffer elástico.

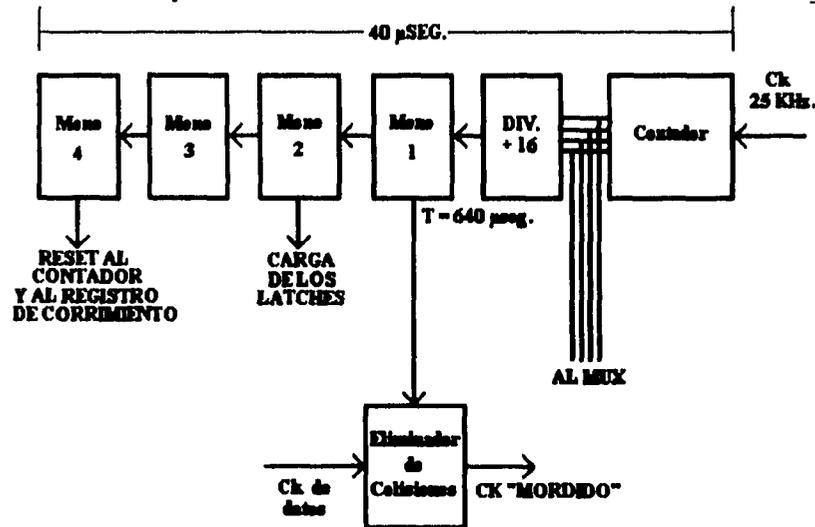


Fig. IV.7.-Generador de pulsos de control.

Contra la costumbre establecida, este diagrama se lee de derecha a izquierda; entonces vemos que el reloj maestro de 25 KHz. se alimenta al contador 7493; este es de cuatro bits; por lo tanto, contará de cero a quince en binario. Las cuatro salidas de este contador alimentan directamente al multiplexor 74150, para que este ponga en serie los 16 bits que tiene en sus entradas; esas mismas cuatro salidas del contador son aprovechadas por el divisor entre 16, que genera un pulso cada vez que el contador llega a 15; este pulso tiene una duración de  $1/25000 \text{ seg.} = 40 \mu\text{seg.}$  y con él se termina el ciclo de rellenado, ya que en este momento, el multiplexor está sacando en bit número 16 que es siempre un cero.

La salida del divisor entre 16, es alimentada a una cascada de cuatro monoestables, con objeto de generar pulsos de una anchura definida y sucesivamente retardados. El retardo total ocasionado por estos cuatro monoestables, deberá ser menor que el tiempo que le lleve al multiplexor transferir un bit; esto es  $40 \mu\text{seg.}$

El primer monoestable produce pulsos angostos cada  $640 \mu\text{seg.}$  que son llevados al circuito de eliminación de colisiones junto con los pulsos del reloj de datos. Si el flanco ascendente del reloj de datos ocurre al mismo tiempo que algún pulso de salida del monoestable, el circuito de eliminación de colisiones "morderá" una pequeña parte del pulso del reloj de datos, con la finalidad de que no se acepte dato alguno cuando los registros de corrimiento están siendo restablecidos, ya que de ocurrir esta circunstancia, se perdería el dato.

La salida del segundo monoestable permite que los bits que han sido acomodados en los registros de corrimiento y en el contador descendente sean transferidos a los latches.

El tercer monoestable, produce un pequeño retardo para que la salida del cuarto monoestable restablezca los registros de corrimiento a ceros y al contador descendente a 12. Este retardo es necesario para que la carga de los latches y la puesta a ceros de los registros no ocurran simultáneamente, porque entonces los latches recibirían datos falsos.

Para el diseño del eliminador de colisiones, vamos a referirnos al cronograma de la figura IV.8.

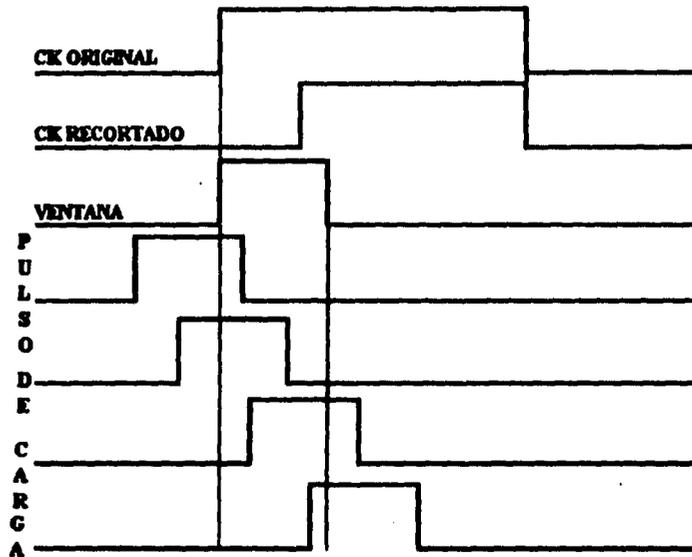


Fig. IV.8.- Cronograma del circuito de eliminación de colisiones.

En la primera línea tenemos un pulso obtenido directamente del reloj de datos. En la segunda línea se ha recortado una pequeña porción del inicio del pulso antes mencionado. La tercera línea nos muestra un pulso que denominamos "ventana", que se inicia al mismo tiempo que el pulso del reloj original. En las siguientes líneas tenemos varias posibles posiciones del pulso de salida del primer monoestable, en las que se puede ver que algunas ocurren al mismo tiempo que la ventana; en caso de que se presente esta circunstancia, será necesario "morder" nuevamente el inicio del pulso del reloj de datos. Con este reloj mordido, se garantiza que los datos nunca van a estar entrando al registro de corrimiento cuando este sea restablecido a ceros.

En la figura IV.9 tenemos el circuito lógico de eliminación de colisiones.

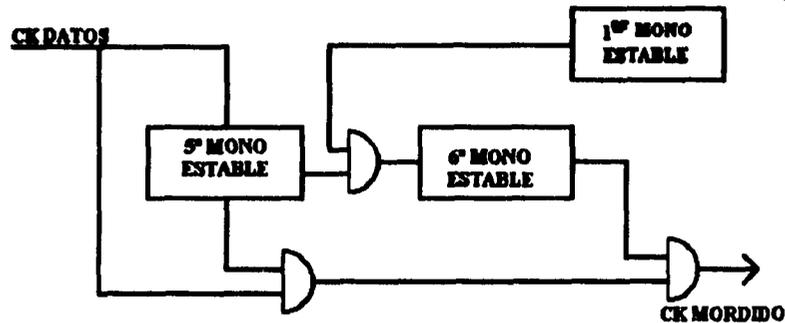


Fig. IV.9.-Circuito lógico del eliminador de colisiones.

Calcularemos ahora los valores de los componentes pasivos que determinan la frecuencia de los relojes.

Las ecuaciones de diseño para el reloj, utilizando el LM555 son:

$$t_{\text{alta}} = 0.695R_A C$$

$$t_{\text{baja}} = 0.695R_B C$$

$$T = 0.695(R_A + R_B) C$$

Para hacer el ciclo de trabajo igual a 0.5, se requiere conectar un diodo entre las patas 6 y 7 del LM 555 y hacer iguales las dos resistencias, con lo cual las tres ecuaciones se convierten en una sola, que se puede escribir como:

$$T = 0.695(2R_A)C.$$

Hagamos primero el cálculo para el reloj de 15 KHz.

Como se tienen dos incógnitas y una sola ecuación, acudimos al viejo truco de seleccionar el valor del capacitor, que será de 0.047  $\mu\text{fd}$ ; con lo cual, el valor de  $R_A$  será aproximadamente de 1000 ohms.

Los valores de los otros componentes del reloj son recomendados por el fabricante, de modo que el circuito queda como se ve en la figura IV.10, donde también se ha dibujado el resto del diagrama del generador de palabras.

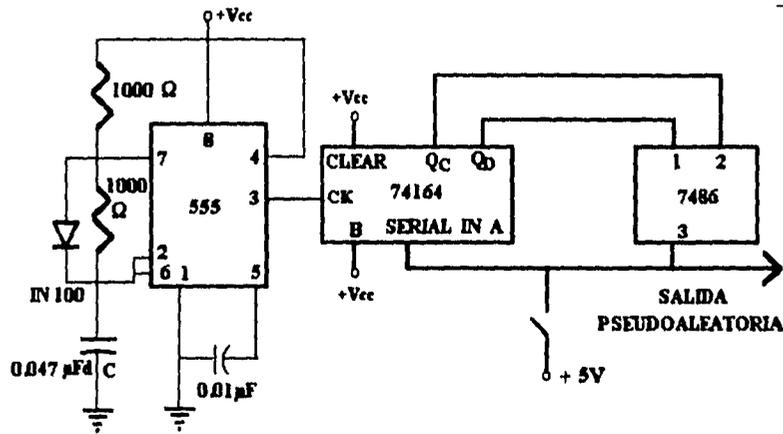


Fig.IV.10.- Generador de reloj y mensaje binario.

Una vez armado el circuito y energizado, es necesario pulsar una vez el interruptor para iniciar su funcionamiento.

Diseño del reloj maestro.- Es similar al diseño descrito anteriormente; la diferencia estriba en que la velocidad es de 25 KHz en lugar de 15 KHz

Escogiendo un capacitor de 0.033  $\mu$ f, obtenemos un valor de resistencia de 872 ohms. Debido a que no hay un valor comercial similar a este, y aunque lo hubiese, la tolerancia de fabricación del 10 % lo haría inadmisibles para nuestros propósitos, se opta por instalar potenciómetros de 1000 ohms en vez de las dos resistencias de 872 ohms y hacer el ajuste fino con el auxilio de un frecuencímetro, para lograr la frecuencia de 25 KHz. con la mayor exactitud posible.

### Cálculo de los componentes de los monoestables.

Para los primeros cuatro monoestables:

Recordando que la suma de la duración de los cuatro pulsos no debe de rebasar los 40  $\mu\text{seg}$ , asignaremos el valor de 8  $\mu\text{seg}$  para la anchura total de los cuatro pulsos; es ahora necesario determinar el valor de R y el valor de C, con la fórmula que se anota a continuación:

$$T = 0.28 R C [1 + 0.7/R]$$

Teniendo en cuenta que para cada monoestable T es igual a 2  $\mu\text{seg}$  y asignando arbitrariamente un valor de 2200 ohms a la resistencia, podemos obtener el valor del capacitor:

$$C = 2000 / [(0.28)(2.2) (1 + 0.7 / 2.2)] = 2463 \text{ pfd.}$$

El valor más próximo comercial es de 2500 pfd.

### Cálculo de los componentes del quinto monoestable:

Teniendo en cuenta que un valor de T de 4  $\mu\text{seg}$  es adecuado y asignando arbitrariamente un valor de 1500 ohms a la resistencia, podemos obtener el valor del capacitor:

$$C = 4000 / [(0.28)(1.5) (1 + 0.7 / 1.5)] = 6493.5 \text{ pfd.}$$

El valor más próximo comercial del capacitor es de 6800 pfd; recalculando la duración de los pulsos, tenemos que:

$$T = (0.28)(1.5)(6800)(1 + 0.7 / 1.5) = 4188 \text{ nseg.}$$

Este valor de duración del pulso del monoestable está dentro de un rango de tolerancia adecuado, por lo que se usarán los valores comerciales obtenidos.

Los valores de los componentes, para el sexto monoestable son iguales a los de los primeros cuatro, debido a que se le va a dar una anchura de pulsos de 2  $\mu\text{seg}$ .

### 3.- Diseño del sistema del receptor.

Los elementos centrales del sistema receptor son:

Un registro de corrimiento para poner en paralelo los datos que van llegando en serie. (74164)

Un conjunto de latches para almacenar temporalmente, los bits acomodados en el registro de corrimiento. (74116).

Un convertidor paralelo-serie o multiplexor para sacar los bits de mensaje, descartando los de relleno y los de control. (74150).

Así mismo, se debe contar con un reloj que pueda generar desde cinco hasta doce pulsos en el lapso de 640  $\mu$ seg, de manera que se pueda reconstruir la anchura de los datos originales. Esto habrá que diseñarlo con técnicas de síntesis de frecuencia a base de PLL (4046).

Este reloj deberá cambiar su frecuencia en función del número de bits de relleno indicados por los tres bits de control, como se explica en la siguiente tablita:

| Bits de control | Pulsos de relleno | Pulsos de reloj cada 640 $\mu$ seg. |
|-----------------|-------------------|-------------------------------------|
| 000             | 0                 | 12                                  |
| 001             | 1                 | 11                                  |
| 010             | 2                 | 10                                  |
| 011             | 3                 | 9                                   |
| 100             | 4                 | 8                                   |
| 101             | 5                 | 7                                   |
| 110             | 6                 | 6                                   |
| 111             | 7                 | 5                                   |

Toda la circuitería anterior deberá ser controlada por el reloj del sistema transmisor y por una lógica que en función de los bits de control, programe la

frecuencia del reloj que va a ensanchar los pulsos del mensaje. Todo ello se puede ver en la figura IV.11.

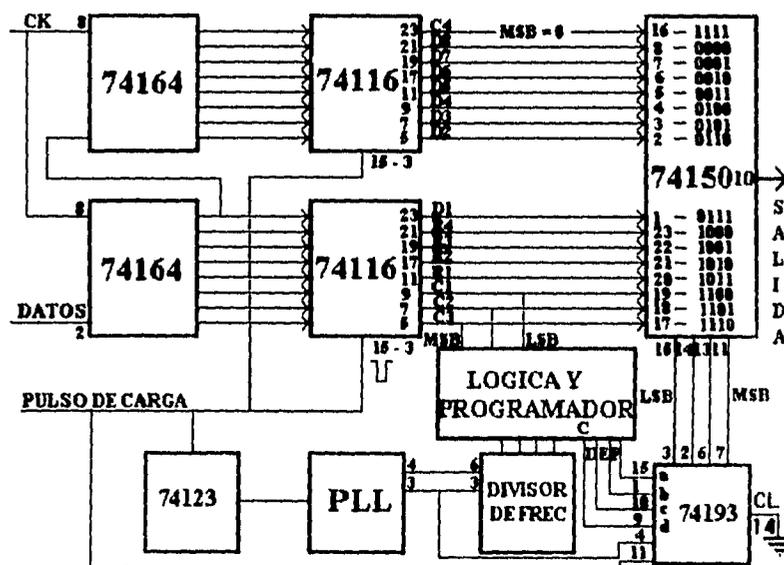


Fig. IV.11.- Diagrama de bloques de la unidad receptora.

En el diagrama de la figura IV.11, se puede ver una caja rotulada "Lógica y programador" que recibe a su entrada los tres bits de control y en función de ellos produce dos salidas: la primera es para programar al reloj PLL a la frecuencia requerida y la segunda es para programar al contador (74193) para que solo los bits de datos pasen de las entradas del multiplexor (74150) a su salida, omitiendo la transferencia de los bits de control y los bits de relleno.

El primer paso para diseñar la lógica de que hablamos antes, es determinar como están acomodados los 16 bits a la salida de los latches del receptor; para esto empezamos por ver como están acomodados en los latches del transmisor.

Según podemos ver en la figura IV.12, en la entrada 16 del mux del transmisor se tiene el cuarto bit de control, que siempre es cero; en las entradas 13, 14 y 15 se tienen los tres bits de control; más abajo tenemos los N bits de relleno y hasta abajo tenemos los M bits de mensaje.

Por el orden en que quedaron ubicados los 16 bits en las entradas del multiplexor, el primero en salir a la línea es el cuarto bit de control; en seguida salen los bits del mensaje empezando por el último; luego los bits de relleno y finalmente los tres bits de control.

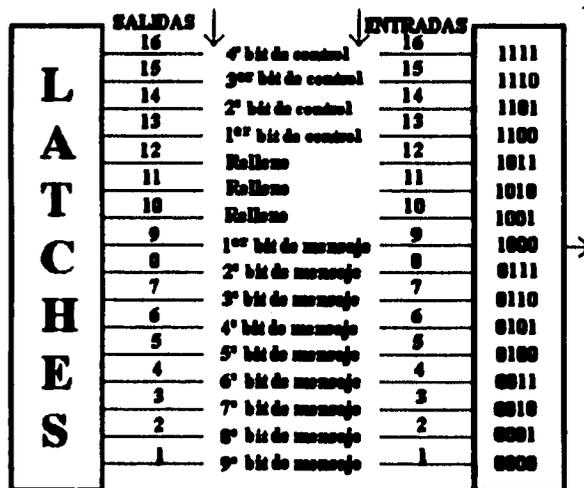


Fig. IV.12.- Disposición de los bits en el mux del transmisor.

En la figura IV.13 podemos ver como han quedado acomodados los 16 bits en los latches del receptor.

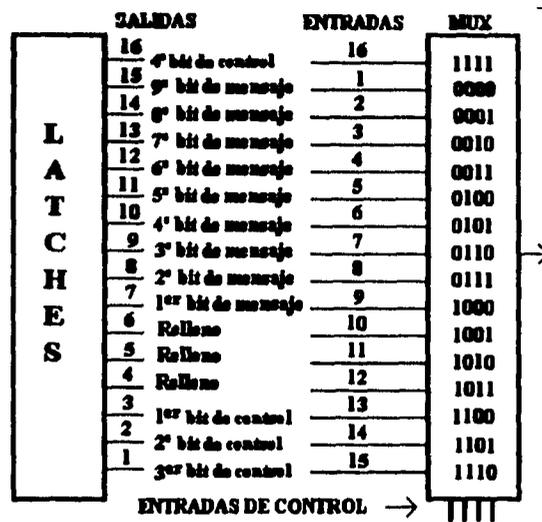


Fig. IV.13.- Disposición de los bits en el mux del receptor.

Analizando la figura IV.12 se nota que las salidas de los latches del transmisor corresponden directamente con las entradas del mux.

En cambio, en la figura IV.13 correspondiente al receptor, vemos que ha sido necesario modificar la correspondencia entre las salidas del latch y las entradas del mux. Esto se ha hecho para lograr que los bits de mensaje puedan salir en serie en el orden correcto. También pueden conectarse entradas con salidas en correspondencia pero esto no simplifica la circuitería.

En la primera columna de la siguiente tabla se tienen las ocho posibles combinaciones de los tres bits de control. En la segunda columna se anotan los bits de relleno que llegan con el mensaje. La tercera columna nos indica en que número debe empezar a trabajar el contador descendente que controla al mux para que solo los bits de datos se transfieran a la salida del mux.

| Bits de control | bits de relleno | inicio de la cuenta | al divisor |
|-----------------|-----------------|---------------------|------------|
| 0000            | 0               | 1011                | 1100       |
| 0001            | 1               | 1010                | 1011       |
| 0010            | 2               | 1001                | 1010       |
| 0011            | 3               | 1000                | 1001       |
| 0100            | 4               | 0111                | 1000       |
| 0101            | 5               | 0110                | 0111       |
| 0110            | 6               | 0101                | 0110       |
| 0111            | 7               | 0100                | 0101       |

La cuarta columna nos dice los bits que se deben enviar al divisor de frecuencia del PLL para generar el reloj adecuado.

Es necesario ahora implementar circuitos lógicos que generen los bits de la tercera y cuarta columnas a partir de la primera columna; esto es fácil ya que la tercera columna es el complemento a 11 de la primera y de la cuarta columna es el complemento a 12 de la primera; por lo tanto, se puede lograr el resultado deseado con restadores lógicos del tipo 7483.

Las operaciones mencionadas en el párrafo anterior se logran con el diagrama de la figura IV.14. Se requiere un restador para el PLL y otro para el contador del mux.

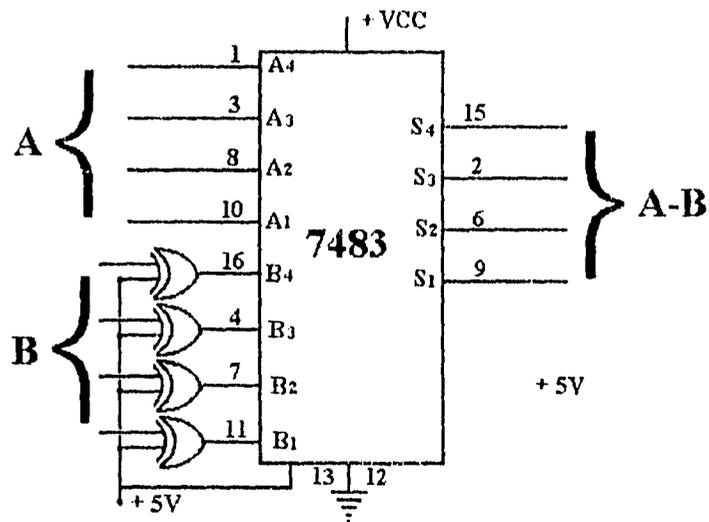


Fig. IV.14.- Restador lógico.

En las entradas A se pone el número 1100 ó 1011 según sea el caso y en las entradas B se alimentan los bits de control de modo que las salidas van al PLL o al controlador del mux.

Para producir el reloj en el receptor, se toma como referencia básica a los pulsos de carga, que ocurren cada 600  $\mu$ seg. o sea cada 16 bits de señal transmitida. Estos pulsos de carga los tomamos directamente del transmisor y entran al sintetizador de frecuencias del PLL para que este pueda generar desde 5 hasta 12 pulsos en el intervalo mencionado. El diagrama de este generador se ve en la figura IV.15

En los circuitos PLL, la señal producida por el OCV se reajusta automáticamente para que su frecuencia sea igual a la de la señal de referencia que alimentamos al comparador; pero se puede lograr que el OCV opere a otra frecuencia si abrimos el lazo entre el OCV y el comparador e intercalamos un divisor de frecuencia programable; de esta forma podemos hacer que el OCV produzca una frecuencia  $n$  veces mayor que la de referencia, ya que el divisor la divide entre  $n$  y el comparador detecta que sus dos entradas tienen la misma frecuencia, con lo cual el lazo queda cerrado.

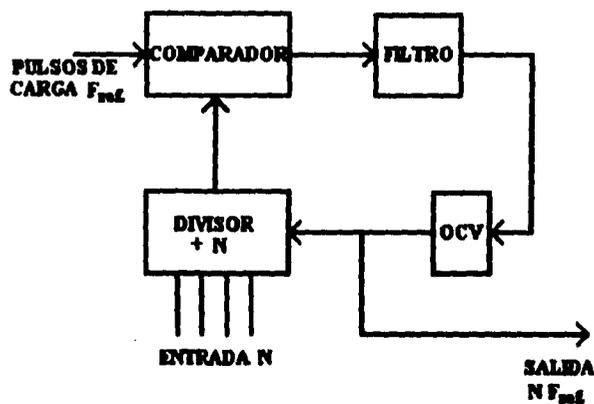


Fig. IV.15.- Generador de reloj de datos.

Cálculo de los componentes del sintetizador de frecuencias.

Como ya se mencionó párrafos atrás, el PLL debe recibir un pulso cada  $640 \mu\text{seg.}$  y en ese lapso debe de producir de cinco a doce pulsos de reloj, dependiendo de la orden que le envíe la lógica. Lo que hay que calcular en este caso son los componentes  $R$  y  $C$  del oscilador controlado por voltaje.

Las fórmulas de diseño del PLL son las que se anotan a continuación:

En los circuitos PLL, la señal producida por el OCV se reajusta automáticamente para que su frecuencia sea igual a la de la señal de referencia que alimentamos al comparador; pero se puede lograr que el OCV opere a otra frecuencia si abrimos el lazo entre el OCV y el comparador e intercalamos un divisor de frecuencia programable; de esta forma podemos hacer que el OCV produzca una frecuencia  $n$  veces mayor que la de referencia, ya que el divisor la divide entre  $n$  y el comparador detecta que sus dos entradas tienen la misma frecuencia, con lo cual el lazo queda cerrado.

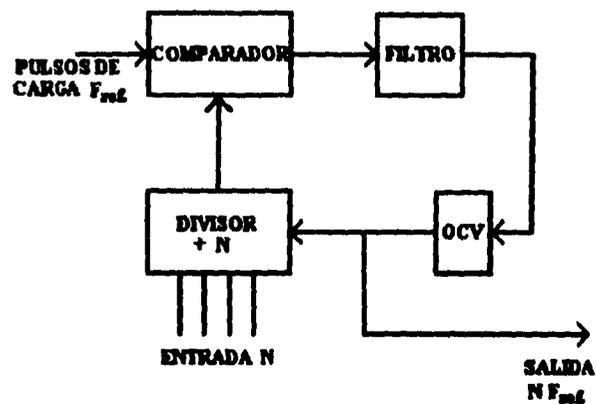


Fig. IV.15.- Generador de reloj de datos.

Cálculo de los componentes del sintetizador de frecuencias.

Como ya se mencionó párrafos atrás, el PLL debe recibir un pulso cada  $640 \mu\text{seg}$ . y en ese lapso debe producir de cinco a doce pulsos de reloj, dependiendo de la orden que le envíe la lógica. Lo que hay que calcular en este caso son los componentes R y C del oscilador controlado por voltaje.

Las fórmulas de diseño del PLL son las que se anotan a continuación:

$$f_{\min} = 1 / R_2 (C_1 + 32 \text{ pf}) \quad V_{\text{co input}} = V_{\text{ss}}$$

$$f_{\max} = 1 / R_1 (C_1 + 32 \text{ pf}) + f_{\min} \quad (V_{\text{co input}} = V_{\text{DD}})$$

Donde:  $10\text{K} < R_1 < 1\text{M}$   
 $10\text{K} < R_2 < 1\text{M}$   
 $100\text{pF} < C_1 < 0.01\mu\text{F}$

Las igualdades anteriores delimitan el rango de frecuencia dentro del cual va a operar el OCV. En nuestro caso, queremos que el PLL se encadene con una señal cuyo periodo es de 640  $\mu\text{seg}$ , correspondiente a una frecuencia de 1562.5 Hz. para que produzca las ocho frecuencias mltiplos que se van a calcular en seguida:

$$f_1 = \frac{12}{640 \times 10^{-6}} = 18750$$

$$f_2 = \frac{11}{640 \times 10^{-6}} = 17187.5$$

$$f_3 = \frac{10}{640 \times 10^{-6}} = 15625$$

$$f_4 = \frac{9}{640 \times 10^{-6}} = 14062.5$$

$$f_5 = \frac{8}{640 \times 10^{-6}} = 12500$$

$$f_6 = \frac{7}{640 \times 10^{-6}} = 10937.5$$

$$f_7 = \frac{6}{640 \times 10^{-6}} = 9375$$

$$f_8 = \frac{5}{640 \times 10^{-6}} = 7812.5$$

De la lista anterior vemos que el OCV debe operar entre 7812 y 18750 Hz, que van a ser nuestras frecuencias extremas de diseño.

Entonces con un capacitor determinado en forma empírica calcularemos los valores de las resistencias, como se ve a continuación.

$$f_{\min} = \frac{1}{R_2(C_1 + 32_{\text{pf}})} \quad (V_{\text{CO input}} = V_{\text{SS}})$$

$$f_{\max} = \frac{1}{R_1(C_1 + 32_{\text{pf}})} + f_{\min} \quad (V_{\text{CO input}} = V_{\text{DD}})$$

$$\begin{aligned} \text{Donde: } 10\text{K} < R_1 < 1\text{M} \\ 10\text{K} < R_2 < 1\text{M} \\ 100_{\text{pf}} < C_1 < 0.1_{\text{pf}} \end{aligned}$$

$$\text{tomando } C_1 = 0.025_{\text{pf}} \quad f_{\min} = 7812.5\text{Hz} \quad f_{\max} = 18750\text{Hz}$$

$$R_2 = \frac{1}{f_{\min}(C_1 + 32_{\text{pf}})} \quad R_2 = \frac{1}{7812.5(25032) \times 10^{-12}} = 5113.45 \Omega$$

$$R_1 = \frac{1}{(f_{\max} - f_{\min})(C_1 + 32_{\text{pf}})} = \frac{1}{(18750 - 7812.5)(25032) \times 10^{-12}}$$

$$R_1 = 3652.46 \Omega$$

Tomaremos los valores comerciales más próximos

$$R_2 = 5000 \Omega \quad R_1 = 3900 \Omega$$

El último juego de valores que debemos calcular son la resistencia y el capacitor del filtro paso bajas del PLL con la siguiente fórmula:

$$f_c = \frac{1}{2\pi RC}$$

Tomando la frecuencia de corte igual a 100 Hz y un capacitor de 1  $\mu$ fd.

$$R = \frac{1}{2\pi f_c C} = \frac{1}{2\pi(100) \times 10^{-6}} = 1591.54 \Omega$$

Tomando el valor comercial más próximo  $R = 1500 \Omega$  Al final del presente trabajo se puede ver el diagrama general del transmisor y el receptor.

## **CAPITULO V**

### **CONSTRUCCION Y PRUEBAS**

#### **1.- Construcción.**

Comentaremos en este capítulo las diversas etapas del proceso que se realizó para poner en operación el sistema de almacenamiento elástico así como las técnicas utilizadas para probarlo y los resultados obtenidos.

El primer paso consiste en realizar un diagrama en el que se muestre la interconexión de los pines o patas de los circuitos integrados entre sí y con los componentes pasivos.

Las figuras V.1 y V.2, representan la interconexión de los elementos del transmisor y del receptor respectivamente.

El siguiente paso del proceso requiere escoger la técnica de montaje que sea sencilla de realizar, segura, que permita la reparación, de buen aspecto y bajo costo y que permita modificaciones posteriores.

Se cuenta entonces con varias opciones para solucionar este problema:

- a).- Montaje en protoboard
- b).- Montaje con alambrado wire wrap
- c).- Montaje en circuito impreso.

La técnica (a) es la más sencilla de realizar, la que permite reparaciones con mayor facilidad, pero no es segura, ya que los componentes se sueltan fácilmente y tiene mal aspecto; además es la que permite más fáciles modificaciones.

La técnica (b) es fácil de realizar y segura, permite fáciles reparaciones y tiene mal aspecto, pero permite modificaciones.

La técnica (c) es difícil de realizar, es segura, es muy difícil de reparar y tiene muy buen aspecto, pero no permite modificaciones.

Las tres técnicas de montaje tienen un costo similar.

Por lo anotado anteriormente se puede ver que la técnica de alambrado con wire wrap es la que más se adapta a nuestras posibilidades.

En las figuras V.1 y V.2 tenemos los diagramas de interconexión de los componentes del generador de palabras y de las unidades transmisora y receptora del sistema de rellenado de bits, tal como van a quedar alambradas en las tarjetas.

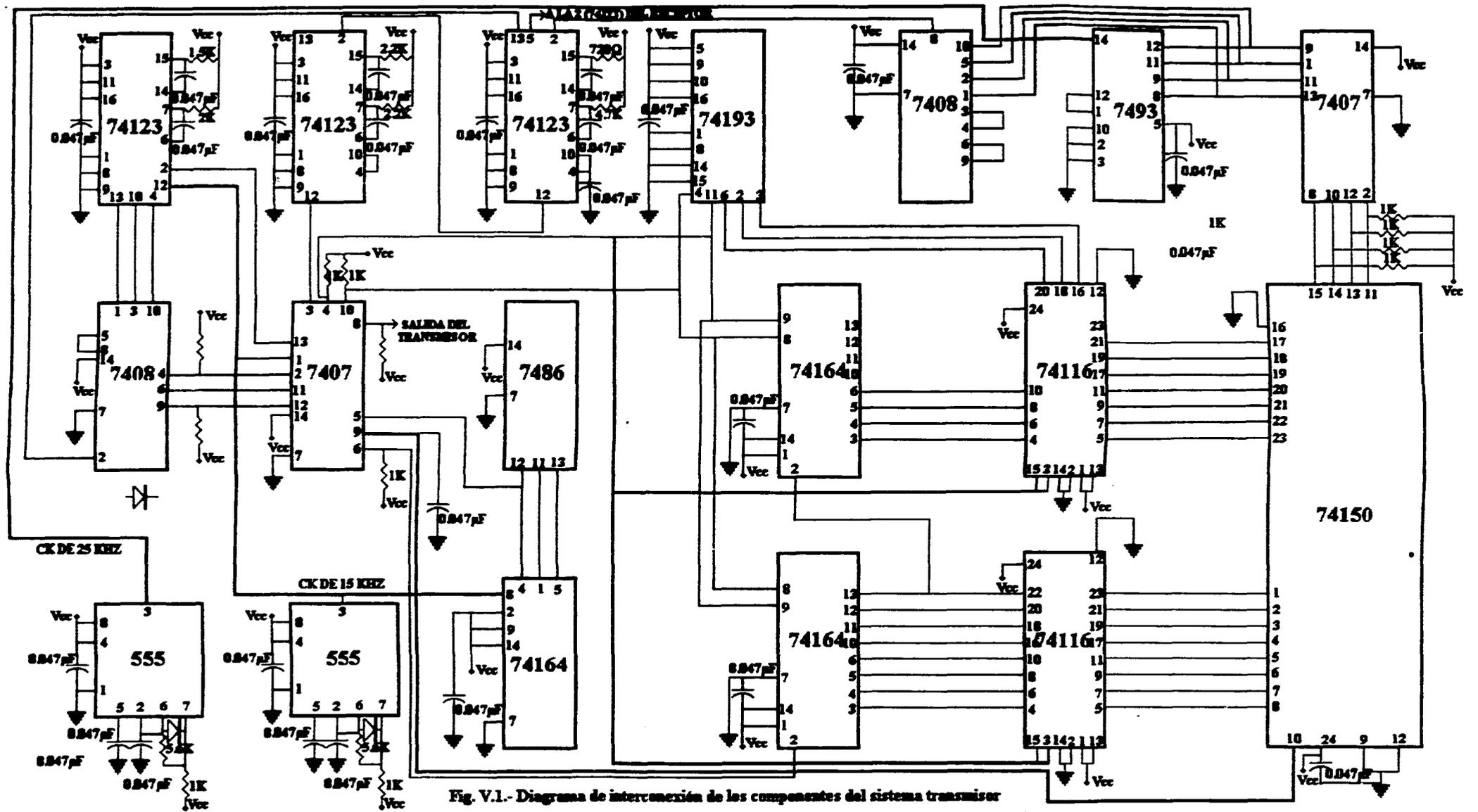
En las figuras mencionadas en el párrafo anterior, aparecen componentes resistivos y capacitivos que no corresponden al diseño original; mismos que se justificarán a continuación.

La técnica empleada para el montaje, aunada a la reducida experiencia del autor en este tipo de trabajos, da como resultado la aparición de imperfecciones en la forma de los pulsos; estos defectos se aprecian como pequeñas arrugas en la parte superior de los pulsos. Estas arrugas pueden parecer inofensivas cuando se ven en la pantalla del osciloscopio; pero si en un momento dado el nivel de voltaje cae abajo del umbral, el circuito que recibe tales señales las va a considerar como ceros, lo cual introduce errores en el procesamiento de la señal.

Para eliminar tales problemas, se utilizan dos técnicas:

La primera consiste en conectar un pequeño capacitor a la salida del circuito que presenta el defecto mencionado. El valor del capacitor debe ser tal que no deforme excesivamente el flanco ascendente de los pulsos y que elimine la arruga que nos causa problemas. El valor del capacitor empleado en cada caso se determina en forma experimental por lo que no se proporcionará cálculo alguno en estos casos. También ayuda la conexión de un pequeño capacitor a tierra directamente en la entrada de alimentación del circuito integrado que presenta el problema.

La segunda técnica consiste en emplear circuitos integrados cuya salida es de las conocidas como "open collector", que nos permiten drenar una corriente considerablemente mayor, lo cual hace que los pequeños voltajes de ruido no afecten demasiado a la señal de salida de los circuitos lógicos. Las resistencias de carga a la salida de los componentes también se determinan experimentalmente.



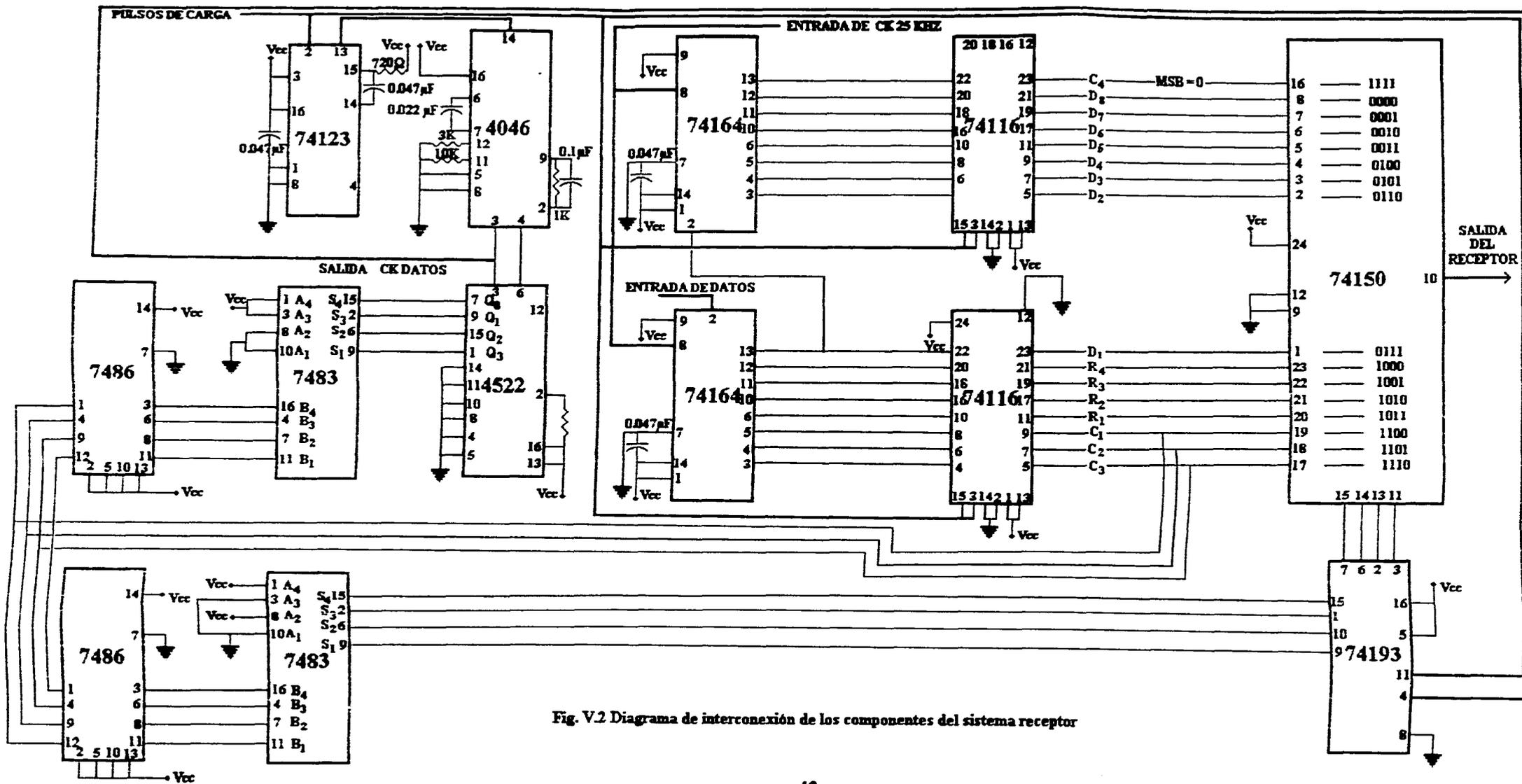


Fig. V.2 Diagrama de interconexión de los componentes del sistema receptor

Se procederá entonces a construir dos tarjetas; en la primera se incluirá el generador de palabras pseudoaleatorias y la unidad transmisora del buffer y en la segunda evidentemente quedará la unidad receptora. La construcción de las fuentes de alimentación de corriente directa no se considera como parte del proyecto.

Las tarjetas van provistas de lenguetas de conexión para ser enchufadas en conectores hembra de 22 patas lo que permite instalar el buffer completo en un bastidor junto con otras tarjetas que ejecuten algún otro proceso de comunicación.

## **2.- Relación de pruebas.**

Las pruebas consisten en medir la frecuencia del reloj de datos así como la frecuencia del reloj del buffer, en obtener oscilogramas de la palabra pseudoaleatoria, de la palabra rellena y de la palabra recuperada a la salida del receptor, estos diagramas se obtienen con la ayuda de un analizador de estados lógicos que nos permiten graficar hasta 16 señales simultáneamente.

En la figura V.3 se presentan los cronogramas de la señal de datos y su reloj. Entre los cursores R y S hay 10 ciclos de reloj y en el encabezado de la gráfica se lee  $R-S = 705 \mu\text{seg}$ ; lo cual corresponde a una frecuencia de 14.184 KHz. Este valor está en concordancia con la especificación de 15 KHz, que se estableció como parámetro de diseño.

En la figura V.4 se presentan los cronogramas del reloj de 25 KHz, de las cuatro salidas del contador que direcciona al multiplexor y de los pulsos de salida de los monoestables num. 1, 2, 3 y 4. Entre los cursores R y S hay 16 ciclos de reloj y en el encabezado de la gráfica se lee  $R-S = 639 \mu\text{seg}$ ; lo cual corresponde aproximadamente a una frecuencia de 25 KHz, en concordancia con las especificaciones originales.

En la figura V.5 se puede ver que los cuatro pulsos de los monoestables están entre los cursores R y X y viendo el encabezado de la figura, podemos calcular que la diferencia entre estos últimos es de 6800 nseg; tiempo menor a los 40  $\mu\text{seg}$ . estipulados para la duración total de los cuatro pulsos.

La figura V.6 muestra los cronogramas del reloj de datos original, de los pulsos de carga y del reloj "mordido"; pudiendo apreciarse que cuando el pulso de carga coincide con el inicio de un pulso de reloj, a este último se le recorta o se le muerde una pequeña parte de su porción inicial.

La figura V.7 nos permite analizar los cronogramas del reloj maestro de 25 KHz, de los pulsos de carga y de la palabra rellenada; pudiéndose notar en este último las tres partes de la palabra; esto es: bits de mensaje, bits de relleno y bits de control.

En la figura V.8 podemos ver el mensaje original y el mensaje recuperado en el receptor, pudiendo notar el defasamiento entre las dos señales binarias.

Con el análisis de las figuras consignadas en este capítulo, podemos concluir que el circuito implementado funciona satisfactoriamente.

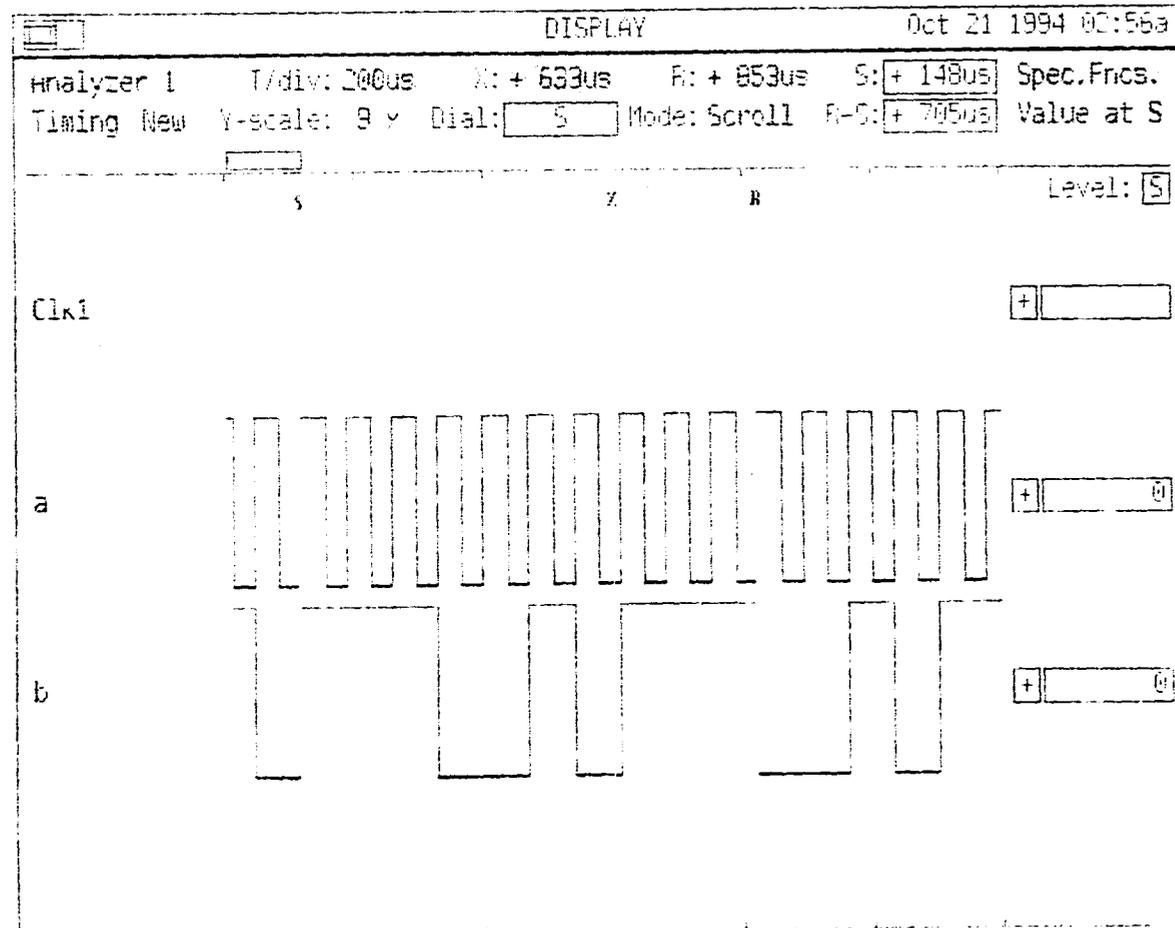


FIG. V.3

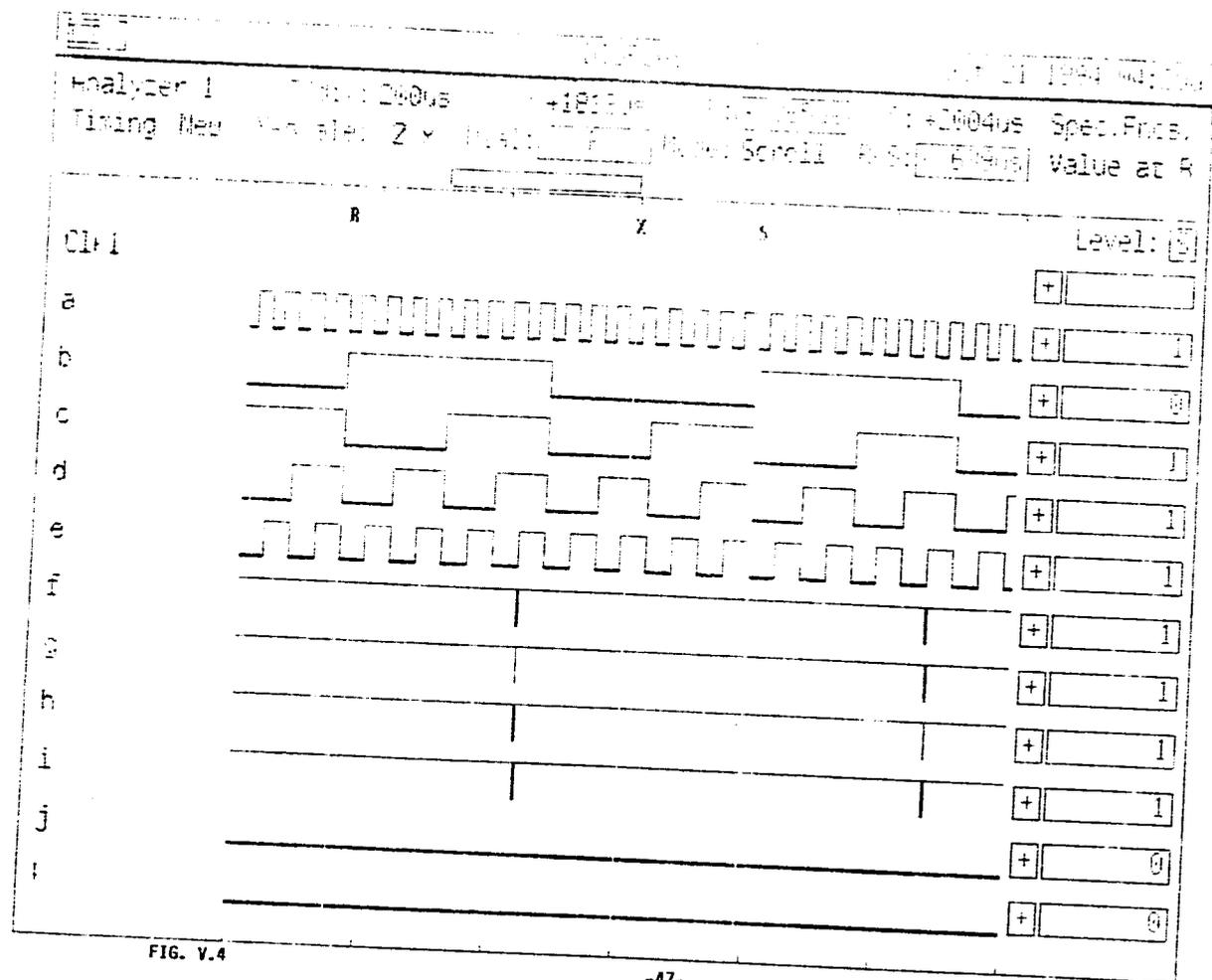


FIG. V.4

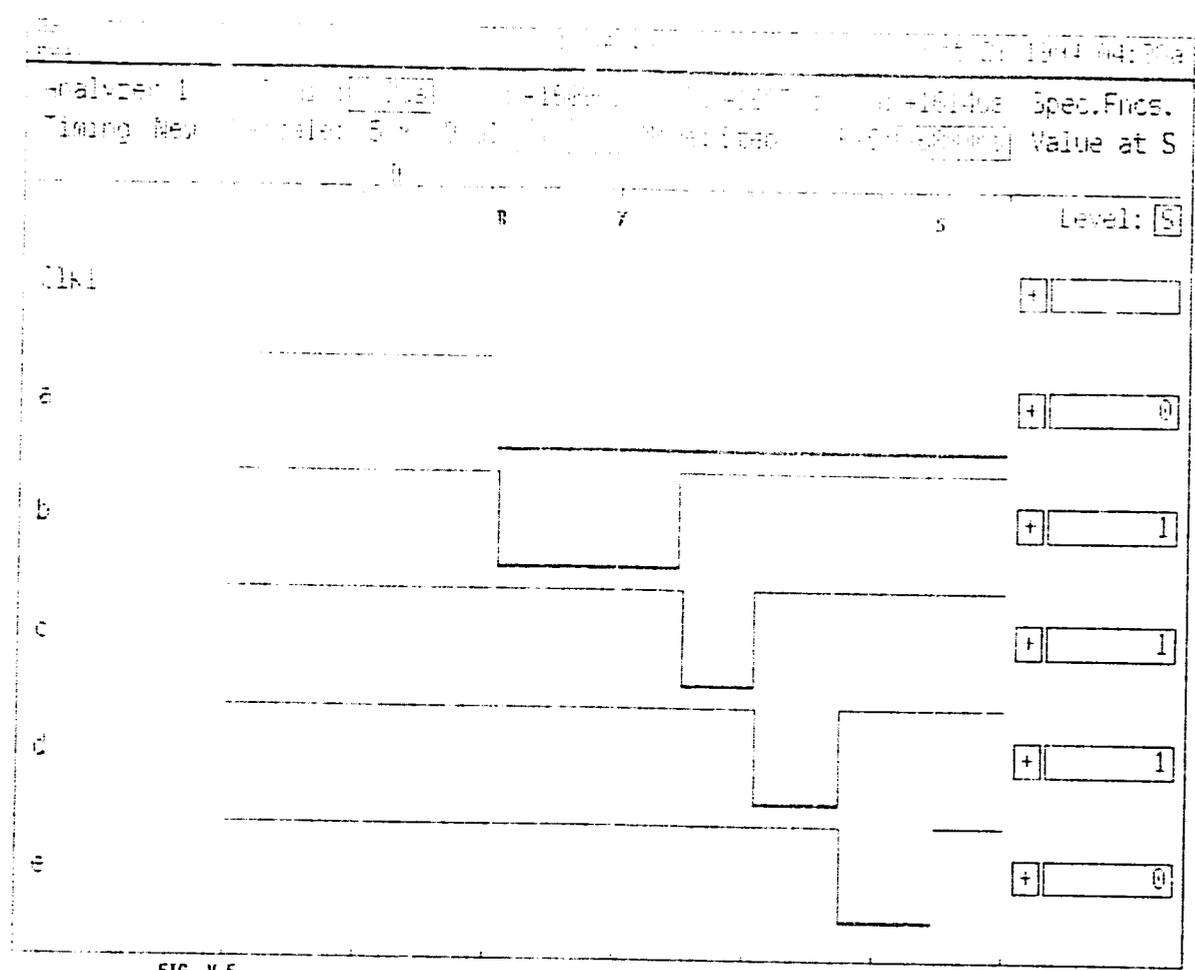


FIG. V.5

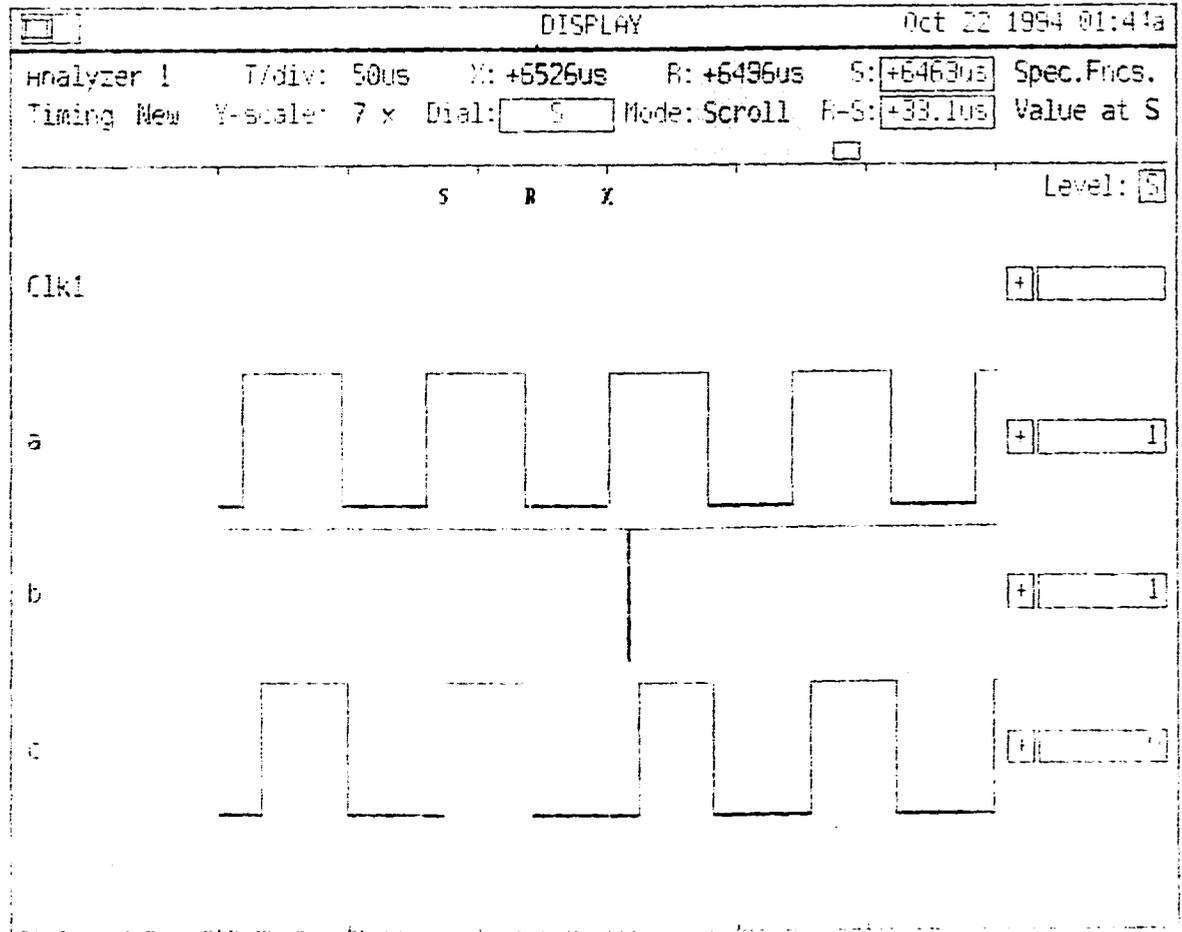


FIG. V.6

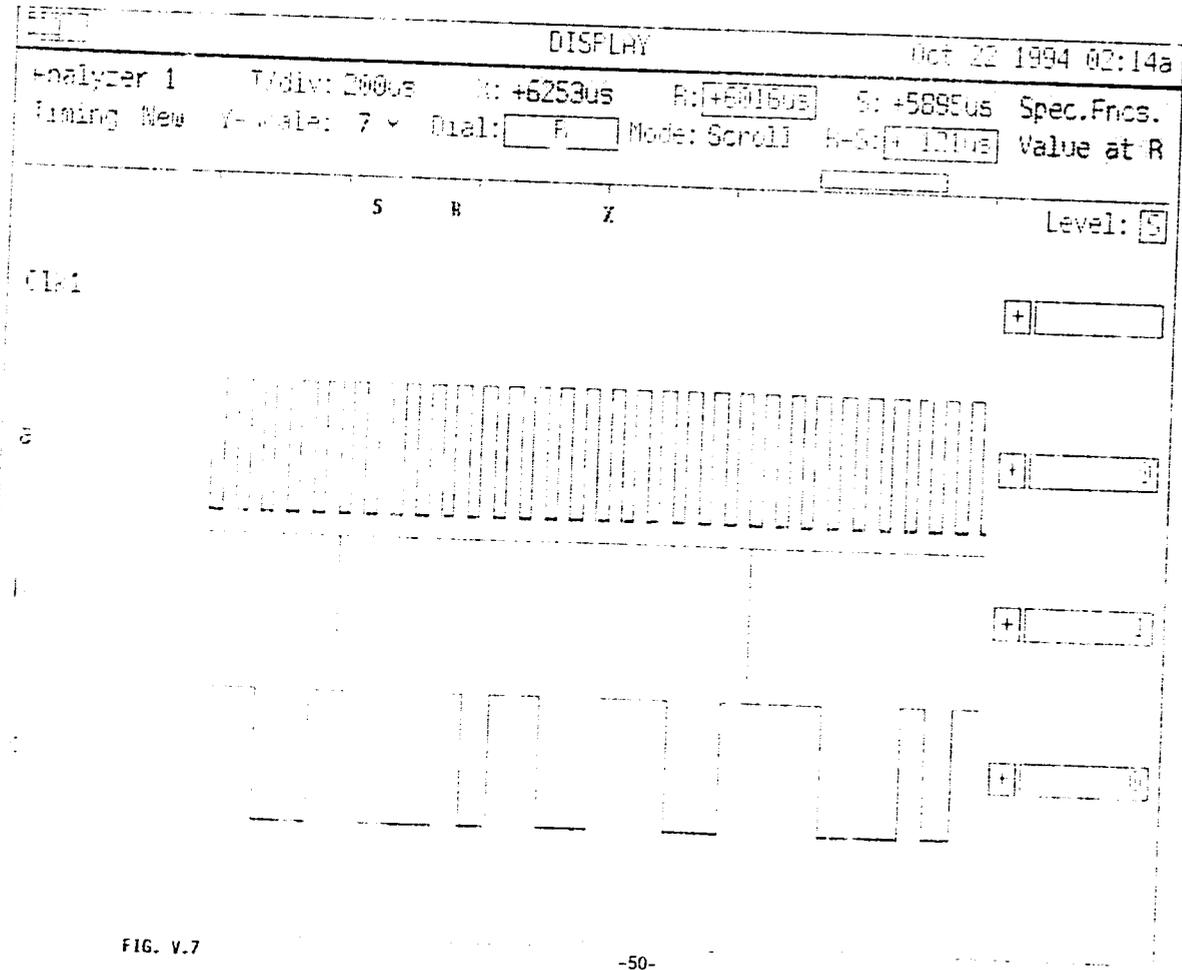


FIG. V.7

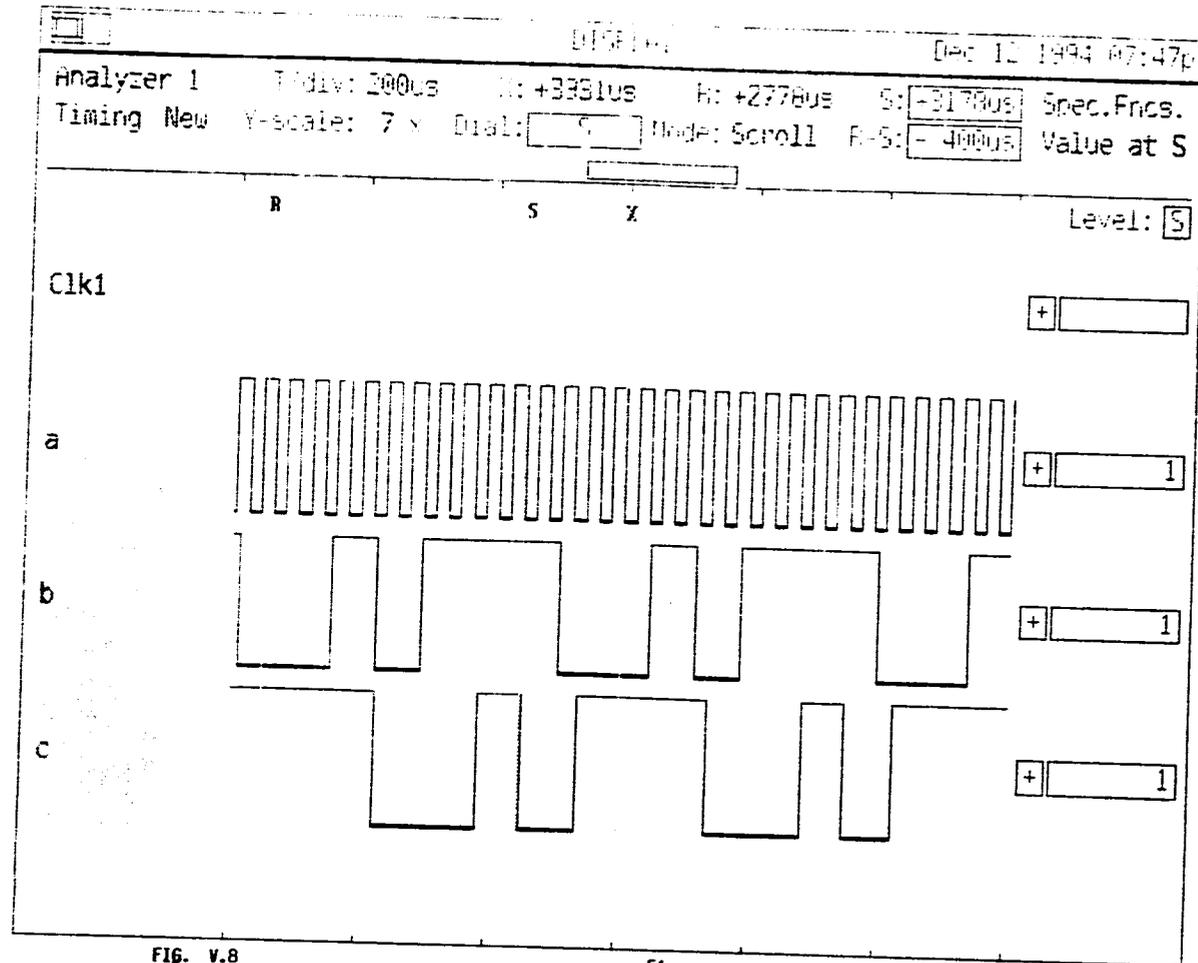


FIG. V.8

## CAPITULO VI

### EVALUACION Y CONCLUSIONES

En este momento se cuenta con un dispositivo que resuelve el problema planteado.

El desarrollo de la tesis tuvo varias etapas:

En la primera, cuando se hicieron los tanteos preliminares, solo se contaba con osciloscopios analógicos de 15 MHz y doble trazo. Se pudo contar con un despliegue de cuatro trazos que no ayudó mucho, debido a la mala calidad del TRC.

En la segunda etapa, tuvimos acceso a un osciloscopio de doble trazo, 60 MHz. y memoria digital. Eso ayudó bastante.

Para la tercera etapa, tuvimos la felicidad de trabajar con un analizador de estados lógicos que nos permitió visualizar los bits "ocultos" en las arrugas de los pulsos con lo cual pudimos proceder a su eliminación, además, los 16 canales simultáneos que pueden verse en ese aparato permitieron encontrar muchos problemas no visibles con el equipo convencional. Fué una gran fortuna contar con ese analizador.

Es necesario ahora mencionar algo acerca del dispositivo implementado. No está optimizado.- Desde el planteamiento se le dió demasiada capacidad de relleno, tolerando hasta 7 bits de relleno por cada 5 bits de mensaje, cuando en los sistemas comerciales se espera cuando más un bit en cada trama de 256 bits. Esto, trae como consecuencia un número excesivo de bits adicionales lo cual aumenta la complejidad del sistema.

Así mismo, la lógica tiene algunos detalles que se notaron gracias a las críticas de observadores externos al proyecto, por ejemplo, el divisor entre 16 no es necesario, ya que la cuarta salida del contador tiene un flanco de subida cada 16 ciclos de reloj; de modo que este pulso puede ser usado para activar directamente al primer monoestable.

Por último se hicieron algunas concesiones simplificadoras al proyecto como la de permitir que el receptor se controle directamente con el reloj del transmisor, siendo que usualmente se debe tener un recuperador de reloj en el receptor a partir de la señal rellenada que recibe.

Una omisión grave en este proyecto fué la de no considerar la posibilidad de un desarrollo de software, o sea, programar el buffer elástico en una computadora o un microcontrolador cosa que está muy de moda.

De todas formas se considera que el dispositivo satisface los requerimientos iniciales y podemos decir "misión cumplida".

## APENDICE

### GENERADORES DE SEÑAL DE PRUEBA PARA LOS SISTEMAS DE TRANSMISION DIGITAL.

Las señales de prueba que se usan para evaluar la transmisión digital deben consistir en secuencias binarias con propiedades estadísticas similares a las de las señales digitales que se transmiten al funcionar el equipo normalmente. La excepción a esta regla son las secuencias especiales que se emplean para verificar el funcionamiento de ciertos circuitos, como son los circuitos de recuperación de reloj; estas señales se eligen normalmente para que representen el peor caso o condiciones marginales.

Estas señales de prueba especiales son generalmente secuencias de información repetitivas y cortas, por ejemplo 0001000...,1110111...,000111000111...,etc, y se generan en circuitos de desplazamiento cuyas longitudes (cantidad de celdas) sean iguales a la cantidad de bits de la secuencia. Para secuencias más largas se pueden usar memorias de propósito general para almacenar la secuencia de prueba que se desee.

Las secuencias de prueba que hemos presentado no sirven para verificar el desempeño total de los sistemas de transmisión, para lograr esto deben ser secuencias aleatorias largas, parecidas a la información.

Existen varios métodos para generar las secuencias aleatorias. Uno de los métodos consiste en usar fuentes verdaderamente aleatorias, como podrían ser una resistencia que genere ruido térmico, un diodo que genere ruido de granalla, o una fuente radioactiva. Desgraciadamente estos métodos carecen de repetibilidad, y las secuencias que se generan pueden no ser verdaderamente aleatorias en caso que los equipos no esten correctamente regulados.

Otro método es programar una computadora para que genere números aleatorios. Este método es repetible, pero en muchas aplicaciones la computadora puede resultar lenta y costosa.

El método más común es utilizar registros de corrimiento con realimentación, generalmente lineal. De esta manera se obtienen secuencias repetibles con propiedades estadísticas satisfactorias.



En la figura A.1 se muestra el ejemplo de un generador lineal de secuencias pseudoaleatorias, que utiliza un registro de corrimiento de cinco etapas con la realimentación hecha mediante una compuerta OR-EXCLUSIVA.

La palabra pseudoaleatoria producida en el registro de la fig. A1 es:

1111100110100100001010111011000.

La secuencia contiene un total de 31 bits ( $2^n-1$ , donde  $n$  es la cantidad de etapas del registro de corrimiento), 16 unos y 15 ceros; hay ocho series de longitud uno (cuatro series de ceros y cuatro de unos), 4 series de longitud dos, 2 de longitud tres, una de longitud cuatro y una de longitud cinco. Además si observamos a través de una "ventana" de cinco bits de ancho, la que podamos desplazar a lo largo de la secuencia de bit en bit, observamos que existen todas las 31 (ó  $2^n-1$ ) palabras no nulas de cinco bits, sin que ninguna se repita. La única palabra faltante es la totalmente nula, pues el registro de corrimiento permanecería en el estado totalmente nulo en caso de aparecer esta palabra.

Estas propiedades, delineadas para una configuración particular, se cumplen para todo registro de corrimiento con realimentación lineal, si la disposición del lazo de realimentación es tal que genera secuencias de longitud máxima. Las secuencias de longitud máxima son las secuencias de  $2^n-1$  bits, generadas en un registro de desplazamiento realimentado de  $n$  etapas, la secuencia se genera periódicamente, repitiéndose cada  $2^n-1$  bits.

Para cada valor dado de  $n$  existen muchas configuraciones posibles de realimentación que producen secuencias de longitud máxima. Cada una de estas configuraciones produce una secuencia diferente, pero todas las secuencias tienen las propiedades estadísticas que se han ilustrado para la secuencia de 31 bits. A continuación se enumeran las propiedades en forma explícita.

1.- La cantidad de ceros es igual a la cantidad de unos, menos uno.

2.- La cantidad de series (de unos o de ceros consecutivos) de longitud  $k$  es aproximadamente el doble de la cantidad de series de longitud  $k+1$ , es decir, la mitad de las series son de longitud 1, la cuarta parte son de longitud 2, la octava parte es de longitud 3, etc.

3.- La secuencia contiene todas las combinaciones posibles de palabras de  $n$  bits, exceptuando la combinación completamente nula. Cada combinación aparece exactamente una sola vez.

Las secuencias de longitud máxima poseen otras propiedades que las hacen objeto de estudios matemáticos muy interesantes, pero que también provocan diversas consecuencias. A continuación se enumeran varias de estas propiedades:

1.- Si se suma en módulo 2 una secuencia de longitud máxima con una réplica de sí misma pero retardada, la suma resulta ser la misma secuencia, retardada en una cantidad distinta de bits.

2.- Supongamos que se muestrea una secuencia de longitud máxima cada bit  $k$ -ésimo. Si  $k$  es par, la secuencia que resulta es una versión desplazada de la versión original; si  $k$  es primo el resultado será una nueva secuencia de longitud máxima con el mismo período.

3.- Autocorrelación de las secuencias de longitud máxima. La autocorrelación de una secuencia es un número igual a la diferencia entre la cantidad de coincidencias y la cantidad de discrepancias de los bits de la secuencia original cuando se le desplaza sobre sí misma. Se acostumbra normalizar los números que resultan, dividiéndolos por la cantidad de bits comparados. Por ello, si el desplazamiento es cero, la autocorrelación es exactamente igual al número de bits  $2^n - 1$  pues todos los bits coinciden; dividiéndolo por el mismo número obtenemos el resultado final de 1. Esta propiedad se cumple en cualquier secuencia, pero al calcular las autocorrelaciones para secuencias de longitud máxima y para desplazamientos cualesquiera que no sean nulos, se encuentra que la diferencia entre la cantidad de coincidencias y discrepancias es exactamente 1, por lo que el valor de la autocorrelación disminuye a  $1/(2^n - 1)$ , donde  $n$  es el número de etapas del registro de corrimiento generador.

De las tres propiedades mencionadas anteriormente, la más importante es la referente a la autocorrelación. Su importancia reside en el hecho de que resulta muy sencillo verificar si dos secuencias de longitud máxima están exactamente alineadas, debido al pico nítido que presenta la función autocorrelación: todo desplazamiento de solamente un bit hace disminuir el

valor de la función autocorrelación de la unidad a  $1/(2^n-1)$ , que es un valor muy pequeño aún para registros de desplazamiento bastante cortos. Por ejemplo, en un registro de desplazamiento de diez bits este valor es  $1/1023$  que es aproximadamente 0.1%.

En la siguiente tabla se muestran las secuencias de las palabras pseudoaleatorias que se pueden obtener con registros de corrimiento de diferentes longitudes.

| Salidas | Secuencias obtenidas                         |
|---------|--|
| 1 y 2   | 110  |
| 1 y 3   | 1110100                                      |
| 1 y 4   | 111101011001000                              |
| 1 y 5   | 111110101001100010000                        |
| 2 y 3   | 1110010                                      |
| 2 y 4   | 111100                                       |
| 2 y 5   | 1111100110100100001010111011000              |
| 3 y 4   | 111100010011010 Palabra usada en el proyecto |
| 3 y 5   | 11111100011011101010000100101100             |
| 4 y 5   | 111110000100011001010                        |

### **Bibliografía:**

**\*Information Transmission, Modulation, And Noise.**  
Mischa Schwartz.  
Editorial Mc Graw Hill Publishing Company.

**\*"Multiplexing of Asynchronous Digital Signals Using Pulse Stuffing with Added-bit Signaling"**  
V.I. Johannes and R.H. McCullough.  
IEEE Trans. Commun. Tech., vol. COM-14, no. 5, pp. 562-568, October 1966.

**\*Circuitos Integrados Lineales y Amplificadores Operacionales.**  
Robert F. Coughlin / Frederick F. Driscoll.  
Editorial Prentice Hall.

**\*Sistemas de Comunicación.**  
Ferrel G. Stremier.  
Editorial Fondo Educativo Interamericano.

**\*Introducción a la Teoría y Sistemas de Comunicación.**  
B.P. Lathi.  
Editorial Limusa.

**\*Sistemas de Comunicación.**  
B.P. Lathi.  
Editorial Interamericana.

**\*The TTL Data Book for Design Engineers.**  
Texas Instruments Incorporated.

**\*Sistemas de Comunicación.**  
A: Bruce Carlson.  
Editorial Mc Graw Hill.

**\*Curso DIGICOM teoría y práctica moderna.**  
de las comunicaciones digitales.  
DEGEM<sup>R</sup> Systems.

**\*Lógica Digital y Diseño de Computadores.**  
Morris Mano.  
Editorial Prentice Hall.

**\*CMOS Logic Data.**  
Motorola Inc.