

**UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA**



**DISEÑO E IMPLEMENTACION DE UN ANALIZADOR LOGICO
EMPLEADO COMO INTERFAZ PARA UN COMPUTADOR PERSONAL**

TESIS

PARA OBTAR EL TITULO PROFESIONAL DE

INGENIERO ELECTRONICO

JOSE MANUEL RIVERA VIGO

PROMOCION 1991-II

LIMA - PERU

1996

Dedico la presente
tesis a mis padres
y hermanos.

SUMARIO

En el presente proyecto se diseña e implementa una interfaz para ser conectada a un computador personal (IBM PC o compatible) así como el software correspondiente. De manera tal, que el computador se comporte como un analizador lógico de tiempos.

El hardware de la interfaz ha sido diseñado íntegramente usando los procedimientos de diseño de la lógica combinacional y secuencial. Para el diseño del software, se ha utilizado el lenguaje de programación Turbo Pascal, específicamente la herramienta de software Turbo Vision.

Al término del proyecto, hemos logrado un analizador lógico de tiempos de 8 canales, aunque por las consideraciones del diseño estos pueden ser fácilmente ampliados; una frecuencia de muestreo de 20Mhz y tres modos de disparo. Es un instrumento de fácil uso, debido a que el software es completamente sencillo de manejar; ya que cuenta con rutinas de manipulación de información así como de ayuda, de mucha utilidad para el operador.

**DISEÑO E IMPLEMENTACION DE UN ANALIZADOR LOGICO
EMPLEADO COMO INTERFAZ PARA UN COMPUTADOR PERSONAL**

**UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA**

TITULO

**«DISEÑO E IMPLEMENTACION DE UN ANALIZADOR LOGICO
EMPLEADO COMO INTERFAZ PARA UN COMPUTADOR PERSONAL»**

AUTOR

JOSE MANUEL RIVERA VIGO

**PROYECTO PARA OBTAR EL TITULO PROFESIONAL DE
INGENIERO ELECTRONICO**

LIMA - 1995

EXTRACTO

La tendencia al cambio de muchas aplicaciones electrónicas convencionalmente analógicas, hacia las técnicas digitales y el diseño de circuitos basados en microprocesadores produjo el diseño y desarrollo de los analizadores lógicos. Estos son instrumentos que nos permiten analizar señales digitales para el diseño de sistemas, depuración de errores, etc.

El presente proyecto, desarrolla un analizador lógico de tiempos como circuito de

interfaz para un computador personal. Esta se controla mediante un software y es capaz de muestrear datos a razón de 20Mhz.

La cantidad de canales del Analizador Lógico es de ocho, con una profundidad por canal de 8 Kbits, con pre-disparo programable. El disparo o trigger se realiza por palabra de disparo u opcionalmente de manera manual, haciendo uso del teclado.

En cuanto al software, se ha diseñado de manera tal que la comunicación con el usuario sea sencilla, esto es posible ya que se basa en menús y ventanas de diálogo.

Las ventajas de usar un computador personal radican en que se puede utilizar la potencia de su hardware, como sus unidades de disco para poder almacenar en ellos los datos adquiridos, así como el poder imprimirlos. Es decir con el presente proyecto de tesis, se precisa contar con un instrumento de poca difusión debido su costo pero que es de gran utilidad.

TABLA DE CONTENIDOS

PROLOGO	1
CAPITULO I. GENERALIDADES DE LOS ANALIZADORES LOGICOS	3
CAPITULO II. ESTUDIO DEL SISTEMA	10
2.1.- La Comunicación con el Computador Personal.	11
2.2.- El Banco de Memoria.	14
2.3.- La Temporización del Sistema.	16
2.4.- La Comparación para el Disparo.	17
2.5.- La Lógica del Pre-Disparo.	18
2.6.- La Lógica de Control.	20
2.7.- Los Modos de Disparo.	21
2.7.1.- El Modo Manual.	21
2.7.2.- El Modo Automático.	23
2.7.3.- El Modo Con Pre-Disparo.	24
2.8.- La Transferencia de Datos hacia el Computador.	27
CAPITULO III. DISEÑO DEL HARDWARE DEL SISTEMA	30
3.1.- El Circuito de Comunicación con el Computador.	30
3.1.1.- Escritura a un Puerto.	33

VIII

3.1.2.-	Lectura desde un Puerto.	35
3.2.-	El Circuito de Temporización.	38
3.3.-	El Circuito de Control.	42
3.3.1.-	Modo Manual	46
3.3.2.-	Modo Automático	48
3.3.3.-	Modo Con Pre-Disparo	50
3.4.-	El Circuito de Pre-Disparo.	54
3.5.-	El Circuito de Comparación.	57
3.6.-	El Circuito del Banco de Memoria.	59
3.6.1.-	Escritura de datos en las memorias.	61
3.6.2.-	Lectura de datos de las memorias.	69
3.7.-	Características Eléctricas de los Circuitos de Entrada.	70

CAPITULO IV. SOFTWARE DEL SISTEMA 72

4.1.-	Estructura del Software del Sistema	73
4.1.1.-	El Programa de Gestión Principal	73
4.1.2.-	El Programa de Gestión de Menús	75
4.1.3.-	El Programa de Gestión de Ordenes	75
4.2.-	Los Menús	78
4.2.1.-	El menú Principal	78
4.2.2.-	El menú (Alt+barra espaciadora)	79
4.2.3.-	El menú Archivo (Alt+A)	81
4.2.4.-	El menú Buscar (Alt+B)	90
4.2.5.-	El menú Asignar (Alt+S)	92
4.2.6.-	El menú Ejecución (Alt+E)	95
4.2.7.-	El menú Resultados (Alt+R)	97
4.2.8.-	El menú Ventana (Alt+V)	101
4.3.-	El Manejador de Interrupciones	105

CAPITULO V. OPERACION DEL ANALIZADOR LOGICO Y RESULTADOS EXPERIMENTALES	107
5.1.- Instalación del Interfaz	107
5.2.- Ejecución Básica del Software	107
5.3.- Instalación de las Sondas de la Tarjeta	109
5.4.- Condiciones Iniciales	109
5.5.- Disparo del Analizador	110
5.6.- Análisis de la Información Adquirida	111
5.7.- Análisis de un Sistema Basado en el μ P Z80	112
5.7.1.- Prueba 1	113
5.7.2.- Prueba 2	116
5.7.3.- Prueba 3	117
CONCLUSIONES	123
BIBLIOGRAFIA	125
APENDICE A COSTOS.	
APENDICE B LISTADO DEL PROGRAMA FUENTE PARA EL COMPUTADOR PERSONAL.	
APENDICE C RESULTADOS EXPERIMENTALES DE LA PRUEBA 3 PARA EL SISTEMA BASADO EN EL μP Z80.	
APENDICE D DIAGRAMA DEL CIRCUITO COMPLETO DEL INTERFAZ.	
APENDICE E HOJAS DE DATOS DE LOS CIRCUITOS INTEGRADOS USADOS EN EL PROYECTO.	

PROLOGO

Cuando se fué haciendo amplia la utilización de microprocesadores de bajo costo, se fueron abriendo también nuevos campos de aplicación para el diseño electrónico, pero al mismo tiempo se encontraron nuevos problemas de la combinación hardware-software.

En el pasado, el diseñador electrónico contaba con un instrumento del dominio del tiempo, es decir, un osciloscopio, para poder encontrar desperfectos y/o mejorar su diseño. Cuando los dispositivos digitales progresaron hacia la tecnología de bus, los diseñadores entendieron que el osciloscopio no era el instrumento más adecuado, pues cualquier avería en su sistema se convertía en una investigación larga y frustrante intentando encontrar el lugar exacto en el cual sincronizar, para poder observar la perturbación, analizar el problema y por último corregirlo.

Así, se produjo una variación en la forma de realizar el análisis electrónico y con la aparición de direcciones, palabras, buses y señales similares, se ha pasado del dominio clásico del tiempo al dominio de datos. Así los fabricantes de instrumentos responden a ésta variación presentando instrumentación de dominio de datos, comúnmente conocidos como Analizadores Lógicos.

El propósito del presente proyecto es la de obtener en base a un computador personal, tan usado hoy en día, un Analizador Lógico de Tiempos; de esta manera podemos contar con un instrumento relativamente barato y así solucionar problemas de diseño y detección de fallas. Ayudando de esta manera al diseñador de circuitos digitales y de sistemas basados en microprocesador.

La presente tesis está dividida en cinco capítulos. En el primer capítulo, se realiza una breve descripción general de los analizadores lógicos, sus tipos y características. En el segundo capítulo, se describe el sistema, analizando al hardware en bloques; así también se cita las consideraciones que se han hecho en el programa que se ejecuta en el computador personal y que hace posible que el hardware del interfaz funcione adecuadamente.

El tercer capítulo, se centra en el diseño en sí del interfaz; allí se muestra cada uno de las partes constitutivas del circuito, haciendo un análisis mediante diagramas de tiempos de cada parte circuital en cada modo de trabajo a la que es sometida el interfaz, consolidando la validación del funcionamiento del hardware del sistema.

En el cuarto capítulo, se describe el comportamiento de software del sistema; se presenta cada una de las ventanas de ingreso de datos, así como la manera de invocarlas. Además, se describen las características que posee el software para el manejo de la información tomada y de si misma.

En el capítulo quinto, se enseña el manejo tanto del hardware como del software; desde la instalación del interfaz hasta la generación de reportes (impresión). Adicionalmente se somete al analizador a tres pruebas experimentales donde el circuito bajo prueba es un sistema basado en el μP Z80.

Quiero aprovechar este espacio para agradecer a mi familia por su apoyo en lo que duró la realización del presente proyecto de tesis. Así también resaltar el apoyo de mi asesor, el Ing^o Aurelio Morales Villanueva, quien revisó con mucho esmero el presente texto aportando valiosas sugerencias.

CAPITULO I GENERALIDADES SOBRE LOS ANALIZADORES LOGICOS

Años atrás, el diseñador de circuitos digitales hubiera utilizado un osciloscopio en el caso de un fallo en el sistema. Obviamente, era el mejor medio para estudiar las características de los impulsos y conseguir la mayor resolución en el dominio del tiempo. Sin embargo, se encontró que habían varias medidas en las que las limitaciones de los osciloscopios eran más evidentes. Estas eran, principalmente, el número inadecuado de canales de medida (usualmente dos o cuatro), la limitación de un osciloscopio para detectar fenómenos no repetitivos en varios canales y la imposibilidad de disparar a partir de una combinación de bits, así como presentar los acontecimientos anteriores a esa combinación (pre disparo).

Así, se creó un instrumento llamado analizador lógico con el propósito de resolver con un sólo aparato el mayor porcentaje posible de los casos de análisis y detección de anomalías que pueden presentarse en la práctica, intentando evitar la necesidad de tener que disponer de varios instrumentos conectados simultáneamente al equipo bajo prueba. De esta manera, no sólo se persigue la identificación de los fallos de funcionamiento, sino también el origen de los mismos.

Cualquier analizador lógico, se compone de cuatro secciones básicas: adquisición de datos de medida, memoria o almacén de datos, sección de control y presentación sobre pantalla. La sección de adquisición o toma de datos los recibe en paralelo y simultáneamente del equipo bajo prueba, mediante una serie de sondas especiales, situadas en los puntos adecuados. El circuito de entrada y las sondas deben de tener unas características

tales que no interfieran con el funcionamiento normal del equipo, ya que de lo contrario se obtendrían conclusiones falsas.

Existen dos procedimientos de adquisición de datos: síncrono y asíncrono. El primero se emplea normalmente para efectuar el análisis de estados lógicos, representándose en la pantalla en sistema binario, octal o hexadecimal a voluntad del usuario, en este caso la secuencia de adquisición de datos del analizador depende de la frecuencia del reloj del equipo bajo prueba. Los analizadores que se basan en este procedimiento se denominan analizadores lógicos de estados. Este instrumento puede comprobar el desarrollo del programa y las transacciones de datos del sistema bajo prueba, representando los datos binarios adquiridos en forma de palabras respecto del tiempo.

El segundo procedimiento se utiliza para la representación de las formas de onda o diagramas de tiempos y en este caso la frecuencia a la que el instrumento toma los datos está determinada por su propio oscilador interno de reloj. El valor de la frecuencia es muy importante cuando se desean “capturar” fenómenos transitorios o ruidos espúreos que causan alteraciones al equipo; ello exige periodos de muestreo muy bajos, lo que implica el empleo de altas frecuencias de muestreo.

La representación del diagrama de tiempos no suele ser la denominada de tiempo real, que es la correspondiente a las formas de onda que realmente han aparecido en los buses, sino que presentan unas reconstrucciones de las mismas a base de los datos de niveles obtenidos durante el muestreo de las diferentes vías. Si el periodo de muestreo es bajo, se obtendrá una imagen muy similar a la real, pero en otras ocasiones pueden pasarse por alto la presentación de impulsos muy breves o de algunos ruidos o fenómenos transitorios también conocidos como “glitches”. Los analizadores que se basan en este procedimiento se denominan analizadores lógicos de tiempos.

Los analizadores lógicos de la primera época, consistían básicamente en analizadores de estados lógicos y disponían únicamente de un display o visualizador alfanumérico, en el que se representaban en el sistema hexadecimal, una serie de estados sucesivos tomados de los “buses” de datos o de direcciones del sistema digital sometido a prueba, y que habían sido previamente almacenados en la memoria interna del sistema.

Los analizadores lógicos de tiempos surgen para poder superar el bajo número de canales que ofrecían los osciloscopios. El analizador lógico de tiempos utiliza las técnicas de muestreo para presentar en un tubo de rayos catódicos las curvas de tensión respecto del tiempo de todos los canales digitales.

En esencia, los analizadores lógicos han permitido unir el espacio que existe entre el hardware y el software, permitiendo al programador ver como se ejecuta su software. Los primeros analizadores lógicos de estados tenían nada más que un display en binario, pero después podían presentar la información en binario, octal, decimal y hexadecimal. También podían reconocer un patrón determinado, así como un retardador digital para capturar los estados después de un determinado número de impulsos de reloj.

Los analizadores lógicos que hemos descrito hasta aquí han sido de uso general y pueden ser utilizados para el diseño y la localización de fallas de todos los sistemas digitales, desde el sistema digital más simple hasta los más complejos como aquellos basados en microprocesador. Adicionalmente a los analizadores lógicos de uso general existen también los analizadores de estado lógicos especiales, los cuales permiten escoger un microprocesador particular como por ejemplo el Z80, 8080, etc.; con la ventaja de poder presentar los códigos mnemotécnicos alfanuméricos del conjunto de instrucciones del microprocesador en particular que se este analizando. Esto permite examinar los bloques de datos rápidamente para determinar si su código se ejecuta adecuadamente. Inclusive existen analizadores lógicos especiales que

pueden influir sobre el microprocesador bajo prueba, con lo que se puede realizar la ejecución de programas paso a paso e inclusive para la ejecución del programa.

Otros analizadores para reducir el número de controles implementan el concepto de “menú”. De manera que operador puede observar en la pantalla los parámetros de medida que pueden seleccionarse. Así bajo el concepto de “menú” permite la selección de muchas funciones con un número reducido de teclas.

El empleo de cada tipo de analizador lógico, responde a la aplicación que se esté desarrollando, es decir depende del circuito bajo prueba. En ciertas aplicaciones se podría utilizar un analizador lógico de estados para disparar a un analizador lógico de tiempos o un osciloscopio para poder magnificar el área alrededor de la falla y así poder estudiar las señales deseadas.

El número de canales disponibles en un analizador lógico va ligado a la frecuencia de muestreo y a la profundidad de memoria por canal. En varios analizadores lógicos el número máximo de canales están sólo disponibles a una frecuencia de muestreo menor que la máxima y con una capacidad de memoria por canal limitada, es decir se sujeta a un compromiso entre las tres características. Esta limitación se hace más evidente en los analizadores lógicos de tiempos, puesto que implica velocidades de adquisición de datos más elevadas.

Existe una posibilidad que resuelve el compromiso número de canales vs. frecuencia, la cual permite que cuando se reduce el número de canales para aumentar la frecuencia, se pueda escoger cuáles de ellos van a quedar activos y cuáles no, evitando tener que reconectar las sondas.

La manera de administrar los recursos disponibles de los tres parámetros consiste en demultiplexar los canales de entrada que queden, repartiendo las muestras entre las memorias que queden libres de los canales no utilizados, ello permite, aumentar la frecuencia sin tener que usar memorias más rápidas, y al mismo tiempo mantener el intervalo de tiempo durante el cual se están adquiriendo los datos.

Todas estas consideraciones son válidas para los analizadores lógicos de tiempos, puesto que en de estados la frecuencia de muestreo es síncrona con la del circuito bajo prueba, y por lo tanto fija. En este caso, sólo se puede jugar con el número de canales y la profundidad de memoria, repartiendo la capacidad total de memoria entre los canales activos.

Cuando en el análisis de tiempos se utiliza una frecuencia de muestreo muy elevada, puede suceder que se capturen muy pocas variaciones de estados y se llene la memoria con datos repetitivos; para solucionar este impase, ciertos analizadores utilizan una técnica de almacenamiento diferente, llamada transicional, que consiste en almacenar en memoria el estado de cada canal solamente cuando se detecta un cambio de estado en alguno de ellos, este método implica almacenar el valor de un contador, el cual contiene el número de muestras que corresponden a cada estado.

En el analizador lógicos de tiempos o asíncrono, la frecuencia de muestreo define directamente la **resolución temporal** del instrumento, calculada como el intervalo de muestreo. La precisión depende de la descompensación entre canales (skew), imprecisión de umbral, inestabilidad y la dispersión del reloj de adquisición. La dispersión entre canales se produce debido al retardo que ocurre al pasar las señales por la sonda de adquisición, a la respuesta distinta de cada canal a los flancos ascendentes o descendentes y a la intermodulación y diafonía que existe entre los canales.

El **disparo** en un analizador lógico determina cuando los datos serán tomados. En primer lugar el disparo en analizador lógico sincroniza todos los canales, permitiendo una comparación del estado de cada uno de ellos en cada instante. El punto de disparo de un analizador lógico puede significar el punto de inicio o fin de la toma de datos. La potencia de disparo del analizador se determina por la secuencia de disparo, la cual define las condiciones exactas que deben cumplirse antes de que ocurra el disparo. La lógica de disparo compara la información de entrada con las palabras de disparo especificadas por el operario con unas condiciones determinadas. El número de comparadores definen la potencia del analizador. La manera como se efectúa el hardware de la sección del disparo, afecta a la velocidad de respuesta del analizador.

Muchos equipos permiten almacenar junto a los datos, la configuración con la que han sido tomados, para lo cual cuentan con una memoria RAM alimentada de una batería en forma constante, o también utilizan discos flexibles o duros. De esta manera se pueden almacenar gran cantidad de datos para su tratamiento posterior análisis con software específico. También es común que los discos usados sean compatibles con PC's, con lo que se podría analizar en cualquier computador personal.

Ya que hemos descrito de manera global a los analizadores lógicos, debemos de presentar a nuestro proyecto como un analizador lógico de tiempos que usa un computador personal como parte del equipo, dotándolo de características propias de los sistemas que manejan datos. En el siguiente capítulo desarrollaremos el estudio de nuestro sistema.

CAPITULO II ESTUDIO DEL SISTEMA

Ahora que estamos familiarizados con los tipos de analizadores lógicos y sus características, mostraremos en este capítulo la solución al problema planteado así como los criterios de diseño tanto de hardware como de software a nivel de sistema, para que así en los Capítulos III y IV se estudie el hardware y el software de manera particular.

Describiremos al sistema estudiando al hardware y al mismo tiempo iremos señalando los aspectos de software que acompañan y hacen posible que el hardware funcione. El sistema hardware se compone de 6 partes:

La Comunicación con el Computador Personal.

El Banco de Memoria.

La Temporización del Sistema.

La Comparación para el Disparo.

La Lógica del Pre-Disparo.

La Lógica de Control.

La Figura 1, muestra el diagrama de bloques del hardware del sistema. A continuación describiremos cada parte del hardware.

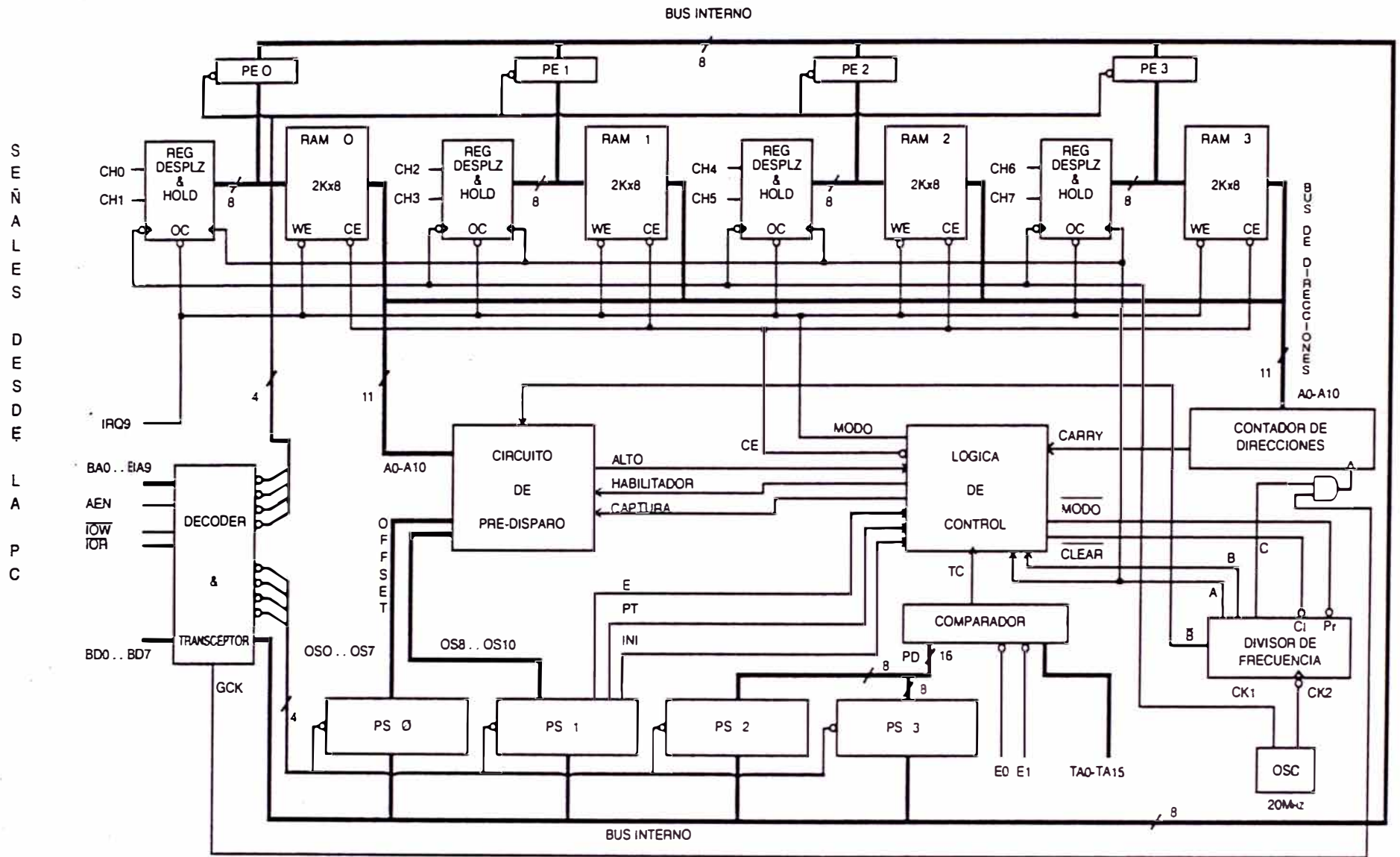


Figura 1. Diagrama de bloques del hardware del sistema

2.1.- La Comunicación con el Computador Personal

Esta parte del circuito nos permitirá la comunicación desde el computador personal con el interfaz de manera que podamos mediante el software de control, que se ejecuta en el computador, colocar a los circuitos internos del interfaz en el modo adecuado seleccionado según el operador.

Esta parte del interfaz también nos permite ingresar los datos adquiridos en la memoria de la tarjeta luego de haber ocurrido una de las tres formas de disparar al analizador, las cuales se detallarán posteriormente.

Se compone de un circuito decodificador, un transceptor, cuatro puertos de salida y cuatro puertos de entrada. El DECODER sirve para direccionar a 4 puertos de salida (ellos en la Figura 1 son: PS 0, PS 1, PS 2 y PS 3) y a los 4 puertos de entrada (PE 0, PE 1, PE 2 y PE 3); tanto los puertos de entrada como de salida son de ocho bits cada uno. El TRANSCEPTOR une el bus de datos del computador personal (sólo los 8 bits menos significativos BD0..BD7) con el bus de datos interno del interfaz. En este BUS INTERNO se desplazan los datos que vienen del computador personal así como también los datos que se obtienen de los bancos de memoria y se dirigen al computador personal pasando por el TRANSCEPTOR.

Las señales del Computador Personal que intervienen en la decodificación son BA0 - BA9, AEN, IOW y IOR. De las señales del bus de datos del computador, solamente se toman las ocho menos significativas, es decir BD0 - BD7. Otra de las señales del computador utilizadas en el interfaz es la señal de interrupción IRQ9, el circuito que activa esta señal se estudiará cuando se analice la Lógica de Control.

Desde el bloque decodificador se entrega una señal etiquetada como GCK (ver

Figura 1), la cual entrega pulsos positivos cada vez que se decodifica un puerto de entrada. Estos pulsos sirven de señal de reloj para el contador de direcciones, el cual se describirá en la sección 2.3.

El interfaz será diseñado para ser colocado en un slot del computador personal cuya arquitectura del bus puede ser ISA, EISA e ISA-VESA.

A continuación se describe para que se emplean cada uno de los puertos de salida.

El Puerto de Salida: PS 0.- Por este puerto salen 8 bits (OS0-OS7) que junto con los 3 primeros bits (OS8-OS10) del puerto PS 1, forman una palabra de 11 bits OS0..OS10 que en la Figura 1 se denomina OFFSET. El OFFSET sirve para indicar al interfaz la cantidad de datos de Pre-Disparo.

El Puerto de Salida: PS 1.- Básicamente es un puerto de control, por allí se le dice al interfaz el modo (de los tres existentes) de disparar al analizador.

Los pines del puerto están definidos así:

- 0, 1 y 2 : Bits que son usados junto a los datos del puerto PS 0.
- 3 y 4 : No son usados.
- 5 : Denominado E (Enable).
- 6 : Denominado PT (Pre - Trigger).
- 7 : Denominado INI (Inicio).

Los bits 5, 6 y 7 definen el Modo de disparo de acuerdo a la Tabla N° 1.

Modo de Disparo	Previo			Arranque		
	INI	PT	E	INI	PT	E
Con Pre-Disparo	0	1	1	1	1	1
Automático	0	0	0	1	0	0
Manual	0	0	1	1	0	1

Tabla N°1. Modos de Disparo del Analizador Lógico

El Puerto de Salida: PS 2.- Sirve para escribir los 8 bits más significativos de la palabra de disparo.

El Puerto de Salida: PS 3.- Sirve para escribir los 8 bits menos significativos de la palabra de disparo.

A continuación se describe para que se emplea cada uno de los puertos de entrada, denotados como PE 0, PE 1, PE 2 y PE 3.

El Puerto de Entrada: PE 0.- Por aquí ingresan los datos de la RAM 0 que almacena la información de los canales 0 y 1 del interfaz.

El Puerto de Entrada: PE 1.- Por aquí ingresan los datos de la RAM 1 que almacena la información de los canales 2 y 3 del interfaz.

El Puerto de Entrada: PE 2.- Por aquí ingresan los datos de la RAM 2 que almacena la información de los canales 4 y 5 del interfaz.

El Puerto de Entrada: PE 3.- Por aquí ingresan los datos de la RAM 3 que almacena la información de los canales 6 y 7 del interfaz.

2.2.- El Banco de Memoria

Esta parte del circuito sirve para el ingreso de datos, provenientes de las sondas (CH0,CH1, CH2, CH3,CH4, CH5, CH6, CH7, ver Figura 1), y su almacenaje en las memorias. Básicamente esta parte consta de un circuito que se repite cada dos canales, en la Figura 2 se muestra el diagrama de bloques para dos canales, el cual considera 2 registros de desplazamiento de entrada serie y salida paralelo de 4 bits (SHIFT), un registro buffer de carga paralelo sincrónico de 8 bits (HOLD) y una memoria RAM de 2K x 8.

En la memoria RAM de 2K x 8, se almacena 2 canales. Cada canal usa 4 bits del bus de datos de la memoria, esto significa que por canal tendremos una profundidad de 2K x 4 bits u 8K bits, o lo que es equivalente a 8192 bits muestreados y almacenados como máximo.

Para que un dato de un canal ingrese hacia la memoria, este primero deberá ingresar a un registro de desplazamiento de 4 bits (SHIFT), el cual usa una señal de reloj de 20Mhz (CK1), es decir cada 50ns los datos son muestreados. Luego una vez que han transcurrido 4 pulsos de reloj, 4 datos están en su salida paralela, estos bits junto a otros 4 bits provenientes del otro registro de desplazamiento (instalado en las mismas condiciones) del otro canal forman un dato de 8 bits; los cuales son conducidos al registro HOLD de 8 bits. Este registro HOLD toma datos a una frecuencia de 5Mhz, es decir 4 veces más lenta que la señal de muestreo en los registros de desplazamiento; esto significa que este registro siempre que adquiera datos desde los dos registros de desplazamientos, captura de éstos siempre información "nueva". Los 8 bits de este registro van hacia el bus de datos de la memoria para grabarse en ella. El registro mantiene estos 8 bits durante un tiempo de 200ns, tiempo suficiente para que la memoria pueda escribirlos dentro de ella.

Resumiendo, cada dirección de memoria contiene un dato de 8 bits, de los cuales 4 bits menos significativos (D0-D3) pertenecen a un canal (CH0) y los 4 más significativos (D4-D7) al otro canal (CH1).

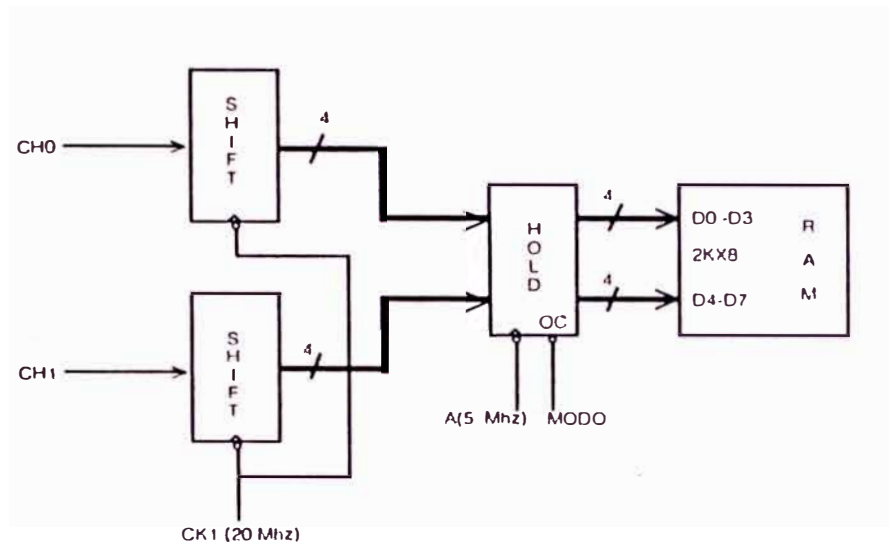


Figura 2. Diagrama de Bloques del Circuito de Muestreo y Almacenaje de Datos en Memoria

El registro HOLD y los registros SHIFT juegan un papel importante en cuanto a la velocidad de muestreo, en el sentido en que a la memoria se le está otorgando un tiempo de ciclo de escritura de $4 \times 50\text{ns}$, es decir 200ns , así se evita de tener una memoria sumamente rápida (que tenga como mínimo un tiempo de ciclo de escritura de 50ns).

Ya que este módulo sólo sirve para almacenar y leer datos de dos canales, entonces el circuito total del banco de memoria considera cuatro de estos circuitos, para tener la capacidad de almacenar datos de ocho canales.

Todas las memorias hacen uso del mismo bus de direcciones el cual proviene de un contador de direcciones (ver Figura 1). Los datos una vez escritos en la memoria, para poder leerlos desde el computador, se realiza la decodificación de los puertos de entrada: PE 0, PE 1, PE 2 y PE 3.

2.3.- La Temporización del Sistema

Las señales de tiempo que hacen posible la generación ordenada de los eventos que soporta el proyecto, nace de dos señales de reloj, CK1 de 20Mhz con 50% de ciclo de servicio (duty cycle), y la otra con idénticas características, CK2 que no es más que la inversión de CK1 (ver Figura 1).

Estas dos señales de reloj controlan la temporización cuando el interfaz está trabajando en adquisición de datos. Cuando ya se escribieron datos en las memorias y se está transmitiendo hacia la memoria del computador, es este último quien controla la temporización.

La señal CK1 será destinada para tomar las muestras en los ocho registros de desplazamiento; esto se realiza en el borde negativo de CK1. La señal CK2 ingresa a un **Divisor de Frecuencia** (ver Figura 1), el cual es sensible al flanco de bajada de CK2. El divisor de frecuencia entrega 4 señales A, B, B y C; las cuales tienen una frecuencia equivalente a la cuarta parte del reloj principal. En la Tabla 2 se muestra la frecuencia y el ciclo de servicio (duty cycle) de cada una de ellas:

	SEÑALES			
	A	B	\bar{B}	C
Frecuencia:	5Mhz	5Mhz	5Mhz	5Mhz
Duty Cycle:	25%	75%	25%	50%

Tabla N° 2. Características de las señales del divisor de frecuencia

Este divisor de frecuencia es síncrono y sus cuatro señales son empleadas por el resto del circuito. A continuación mostramos para que se emplea cada una de dichas señales. La señal A sirve para tomar los datos en todos los registros HOLD (borde positivo), así como también para habilitar la escritura en las ocho memorias (nivel 0); esto último no lo hace directamente sino a través de CE que es una señal de la lógica de control (ver Figura 1).

La señal B sirve para que con su borde negativo se sincronice la toma (en un latch) de la dirección en la cual ha ocurrido en disparo. Esto se detallará en el Modo de Pre-disparo.

La señal B le dice al circuito de pre-disparo el momento que debe comparar las direcciones A0-A10 con las del OFFSET. La señal C sirve para generar los pulsos (borde positivo) para el contador de direcciones.

También tenemos que considerar en esta parte, al CONTADOR DE DIRECCIONES (ver Figura 1) que es el que realiza el conteo de direcciones que usan las memorias, así como también es usado por el bloque del pre-disparo. La señal de reloj que se emplea para el contador proviene desde una compuerta AND (ver Figura 1), esto se debe a que este contador debe usar dos señales de reloj, uno que es la señal C del divisor de frecuencia, el cual se emplea cuando se escribe datos a las memorias y la otra es una señal que proviene del bloque de comunicación con el computador, conocida como GCK (ver Figura 1), la cual produce un pulso cada vez que el computador ejecuta una instrucción de ingreso de datos por un puerto.

2.4.- La Comparación para el Disparo

Para los Modos Con Pre-Disparo y Automático, se necesita que se inicie la toma de datos

cuando una palabra de 16 bits provenientes del circuito de prueba (TA0-TA15) se iguale con los 16 bits que vienen de los puertos de salida PS 2 y PS 3 (ver Figura 1), estos almacenan la palabra de disparo de 16 bits colocadas por el software. La salida de este circuito que es la señal TC (ver Figura 1), envía un pulso (flanco de subida). cada vez que se produce la igualdad de los datos.

Adicionalmente este circuito, cuenta con dos señales habilitadoras E0 y E1 (Enables) sensibles al nivel; que tienen que estar en cero lógico para que la comparación se lleve a cabo. Estos habilitadores nos dan la opción de comparar solamente en ciertos instantes, es decir, cuando los 16 bits que se están sensando del circuito bajo prueba, sean estables; y no siempre, porque podría ocurrir disparos espúreos debido a la inestabilidad que podrían presentar en ciertos momentos los 16 bits del circuito bajo prueba, como por ejemplo cuando estas líneas entren en estado de alta impedancia o ciertos desfasajes debido a retardos etc.

2.5.- La Lógica del Pre-Disparo

Para producir la característica de obtener muestras antes de que ocurra el disparo, se debe programar al interfaz de manera que empiece a tomar datos antes de que el disparo ocurra. El arranque de este modo es controlado e iniciado desde el teclado (por medio del software en la PC). Así se asegura de tener siempre datos grabados en memoria antes de que el disparo ocurra.

Dentro de las características de este modo, se considera, que la cantidad de muestras tomadas antes del disparo es programable; significando que por medio del software se puede indicar un número de muestras que van desde 4 hasta 8188 muestras previas al disparo.

Para implementar electrónicamente esta característica, este circuito de pre-disparo consta de un Registro, un Sumador y un Comparador. En la Figura 3, se observa el diagrama de bloques del circuito de pre-disparo.

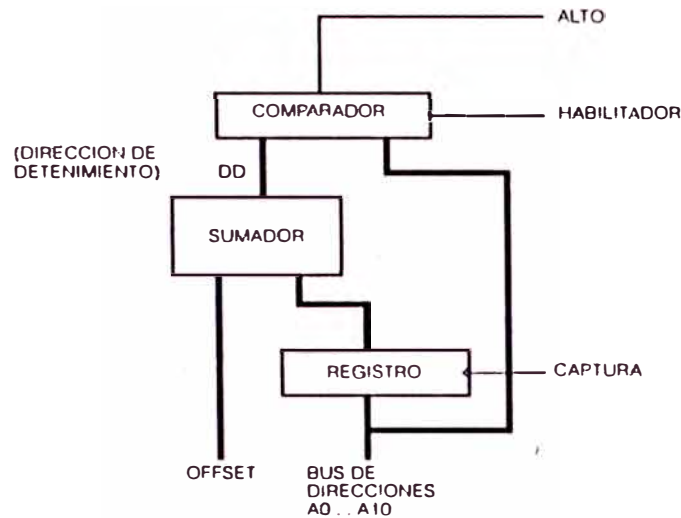


Figura 3. Diagrama de bloques del Circuito de Pre-Disparo

Cuando el disparo ocurre, la señal CAPTURA que proviene de la lógica de control (ver Figura 1), hace que el registro capture la dirección A0..A10 del contador de direcciones. Esta dirección tomada en el registro, ingresa al circuito sumador para que se le agregue el OFFSET, que no es más que la cantidad de direcciones de memoria de post-disparo; el OFFSET es calculado por el computador y se envía al interfaz mediante los 8 bits del puerto de salida PS 0 y los 3 bits menos significativos del puerto de salida PS 1. El circuito sumador produce en su salida la Dirección de Detenimiento DD (ver Figura 3); esta dirección es “mayor” que la que se tomó en el registro, por lo que el contador de direcciones deberá producir una cantidad de direcciones (igual a OFFSET) para que se iguale con DD; cuando esto ocurre, el comparador envía un pulso (en su salida ALTO) que va hacia el circuito de control (ver Figura 1), el cual detiene la secuencia de grabación de datos. Así las direcciones de la memoria quedan divididas en dos grupos, las que fueron tomadas después del disparo y los que fueron tomados antes del disparo.

El comparador es controlado por las señales HABILITADOR y \bar{B} para realizar la comparación, estas señales serán detalladas en el capítulo III.

El OFFSET, es decir la cantidad de direcciones de post-disparo, lo calcula el computador haciendo una diferencia entre el número total de direcciones que soportan las memorias, es decir $800H = 2048$; y el número de muestras que se desea de pre-disparo dividido por 4; como sigue:

$$\text{OFFSET} = 800H - (\text{N}^\circ \text{ de Muestras})/4$$

Para que el OFFSET sea siempre un número entero, es requisito que el **Nº de Muestras** (que lo ingresa el usuario de acuerdo a sus necesidades) sea un múltiplo de 4. Si el usuario ingresa un número de muestras que no es múltiplo de 4, entonces el software lo modifica asignando un múltiplo de 4 basado en el que se ingresó.

2.6.- La Lógica de Control

La lógica de control maneja los modos de operación del analizador lógico. Este circuito de control se alimenta de las siguientes señales eléctricas (ver Figura 1):

3 señales de control que vienen del puerto PS 1, es decir INI, PT y E.

La señal TC que proviene del comparador, indicando que ha sucedido un disparo.

La señal ALTO del circuito de pre-disparo.

La señal CARRY del contador de direcciones.

Por último las señales A y B del divisor de frecuencia.

El circuito de control produce las siguientes señales de control las cuales son (ver Figura 1):

CLEAR y MODO que van respectivamente a Clear y Preset del Divisor de Frecuencia.

La señal CAPTURA que va al circuito de Pre-Disparo para capturar la dirección que estaba transcurriendo cuando se produce un disparo.

La señal HABILITADOR que va al circuito de Pre-Disparo para habilitar o deshabilitar la intervención de dicho circuito.

La señal de interrupción (IRQ9), que es la misma etiquetada como bit de MODO, la que maneja la lectura y escritura de las memorias RAM, así como también el controlador de alta impedancia (OC) de cada uno de los registros HOLD.

Por último la señal CE, que controla los habilitadores de todas las memorias.

El Disparo o Trigger se realiza en cualquiera de los 3 modos mencionados anteriormente.

A continuación presentamos estos modos de disparo.

2.7.- Los Modos de Disparo

En esta sección describiremos como se efectúan los modos de disparo con los que cuenta el analizador lógico motivo de esta tesis.

2.7.1.- El Modo Manual

Este modo es controlado desde teclado, desde allí se dará la señal para que el interfaz empiece a capturar los datos. En primer lugar el usuario invoca al modo manual, eso significa que debe escribir los bits de control en los siguientes valores: E = "1", PT = "0" e INI = "0" (ver Tabla N° 1). Estos valores hacen que el circuito de control establezca sus circuitos internos adecuadamente para este modo. Así, el divisor de frecuencia es limpiado, las memorias son colocadas en modo de escritura

y son habilitados los registros HOLD; de manera que los datos, provenientes de los registros de desplazamiento, ingresen a las memorias. También mediante la señal HABILITADOR se asegura que el circuito de pre-disparo no intervenga durante el modo manual.

Inmediatamente después, desde el computador se cambia el bit INI a “1”, (los otros bits de control PT y E permanecen igual). Esto genera que el divisor por 4 empiece a trabajar, en especial la señal C que es quien hace indirectamente de reloj para el contador de direcciones; es a partir de este instante en que se comienza a tomar datos. La señal CE comienza a oscilar (tal y conforme lo hace la señal A), habilitando y deshabilitando a las memorias luego de cada escritura en ellas. Los primeros datos válidos son escritos en la dirección 000h del contador de direcciones. Cuando el contador llega de nuevo a 000h, él ordena su propia detención enviando la señal CARRY al circuito de control. Este último produce la puesta a “1” de las señales A, B y C del divisor de frecuencia; con esto la señal C se detiene no ofreciendo más pulsos y habilita, mediante la compuerta AND, a que la señal GCK haga las veces de reloj para el contador de direcciones (ver Figura 1). También la lógica de control hace que las memorias queden totalmente habilitadas para su lectura, los registros HOLD vayan a alta impedancia (mediante la señal MODO) facilitando así la salida de los datos de las memorias (ver Figura 1 y Figura 2), a la vez que pide la interrupción 9 de hardware (IRQ 9). Es decir, deja todo listo para la transferencia de información entre el interfaz y el computador, esto se explicará posteriormente en este mismo capítulo.

Cuando termina la transferencia de datos, el computador coloca nuevamente $E = "1"$, $PT = "0"$ e $INI = "0"$; por lo tanto el circuito de control establece lo mismo que se indicó anteriormente al inicio de éste modo de disparo.

2.7.2.- El Modo Automático

En este modo, la adquisición de datos empieza cuando se produce la igualdad entre la palabra de 16 bits de los puertos PS 2 y PS 3, y los 16 bits del circuito bajo prueba TA0..TA15 (estos últimos bits son escogidos de acuerdo a las necesidades del operador).

Este modo es similar al modo manual, la diferencia está en que, en lugar de que la señal INI sirva de arranque para la toma de datos, la responsable de esto es la señal TC que proviene del circuito comparador dando inicio a la toma de datos.

El operador mediante el software debe colocar previamente, antes de solicitar este modo, la palabra de disparo (16 bits) en PS 2 y PS 3; con esta palabra se quiere que cuando el circuito bajo prueba entregue por las líneas de entrada TA0..TA15 la misma cantidad, se empiece a tomar datos.

Luego también desde el computador se envía al puerto PS 1, los bits de control como sigue: E="0", PT="0" e INI="0". Estos valores hacen que el circuito de control establezca sus circuitos internos adecuadamente para este modo. Así el divisor de frecuencia es limpiado, las memorias son colocadas en modo de escritura así como también son habilitados los registros HOLD de manera que la información proveniente de los registros de desplazamiento ingresen a las memorias. También mediante la señal HABILITADOR se asegura que el circuito de pre-disparo no intervenga durante el modo Automático.

Inmediatamente después, desde el computador se cambia el bit INI a "1", (los otros bits de control PT y E permanecen igual). A partir de ese instante queda el circuito en espera de que se produzca el disparo, es decir un pulso proveniente del circuito comparador.

Cuando el comparador envía un pulso mediante la señal TC, este realiza lo mismo que hacía la señal INI en el modo manual. Es decir, genera que el divisor por 4 empiece a trabajar de manera que el contador de direcciones comience a presentar las direcciones válidas, comenzando así la adquisición de datos. La señal CE comienza a oscilar habilitando y deshabilitando a las memorias luego de cada escritura de un byte en ellas. Los primeros datos válidos son escritos en la dirección 000h y así sucesivamente hasta la dirección 7FFh y cuando el contador de direcciones llega de nuevo a 000h, envía la señal CARRY al circuito de control, el cual pone en "1" lógico a las señales A, B y C del divisor de frecuencia; finalizando así la escritura en memorias.

Al igual que en el modo manual, se produce el cambio de reloj para el contador de direcciones, la nueva señal de reloj es GCK (ver Figura 1). La lógica de control coloca a las memorias en modo de lectura, la salidas de los registros HOLD en alta impedancia facilitando así la salida de los datos de las memorias (ver Figura 1 y 2) a la vez que pide la interrupción 9 de hardware (IRQ 9). Así queda todo listo para la transferencia de información entre el interfaz y el computador.

Cuando finaliza la transferencia de datos, el computador coloca nuevamente E = "1", PT = "0" e INI = "0"; por lo tanto el circuito de control establece lo mismo que se indicó anteriormente al inicio del modo manual.

2.7.3.- El Modo Con Pre-Disparo

En este modo de disparo a diferencia de los modos precedentes, (los cuales empiezan a tomar datos a raíz de que ocurre el disparo) nos da la posibilidad de poder observar una cantidad de muestras antes de que el disparo ocurra. Aquí el disparo también es hecho por el circuito comparador mediante la señal TC, es decir se logra el trigger por palabra de disparo (programable desde el computador).

Esta característica se basa en que antes que ocurra un disparo, el circuito de interfaz ya está tomando datos, de forma tal que cuando exista un disparo, ya la memoria tiene datos previos; así se podrá tener datos en la memoria divididos en: antes del disparo (pre-disparo) y después del disparo (post-disparo).

Como la memoria tiene una capacidad total por canal de 8192 bits, parte de estos bits serán de pre-disparo y el resto de post-disparo. La cantidad de datos de pre-disparo es programada por el operador haciendo uso del software; estos oscilan desde 4 hasta 8188 datos y deben ser una cantidad múltiplo de 4. Si no se ingresa un número múltiplo de 4 entonces el software lo arregla tomando un número múltiplo de 4 inmediato inferior al número que se ingresó.

La condición de que el número de muestras debe ser un número múltiplo de 4, obedece a que los datos son muestreados bit a bit en el registro de desplazamiento pero son ingresados a la memoria de 4 en 4, por lo tanto de dirección en dirección de memoria se guardan arreglos de bits de 4 en 4 (esto es válido cualquiera que sea el modo de disparo).

Lo dicho anteriormente supone que cuando se esté trabajando en este modo de pre-disparo, y se dé la orden de inicio de este modo, entonces a partir de ese instante las memorias entran a ciclos de escritura tomando información previa; y si en cierto instante ocurre un disparo (originado por el circuito comparador) entonces existirá la incertidumbre de saber exactamente que bit se estaba tomando cuando ocurrió el disparo, esto es así porque ese bit pertenece a una dirección de memoria y en el momento de leer la memoria, no sabremos en cual de los 4 bits que ella posee se produjo el disparo. Lo que si se podrá saber es la dirección que contiene al bit que estaba siendo muestreado cuando se produjo el disparo y eso es lo que se hace, capturar la dirección que estaba transcurriendo cuando se produce un disparo.

Con lo anteriormente dicho, también no se podrá entonces exigir un número de bits cualquiera de pre- disparo debido a que los bits están agrupados de 4 en 4, significando que se podrá pedir 4, 8, 12, 16 etc. es decir múltiplos de cuatro hasta 8188 muestras previas. No obstante, el software deja que el operador ingrese cualquier número, e internamente corrige eligiendo el múltiplo de 4 más cercano inmediato inferior, para los cálculos de pre-disparo. Además debemos indicar que la cantidad múltiplo de 4 ingresada, tendrá una incertidumbre, es decir un error desde 1 hasta +3 muestras, este detalle será aclarado cuando analicemos los diagramas de tiempos de esta parte del circuito, en el Capítulo III.

El operador, mediante el software debe colocar previamente, antes de solicitar este modo, la palabra de disparo (16 bits) en PS 2 y PS 3; con esta palabra se quiere que cuando el circuito bajo prueba entregue por las líneas de entrada TA0..TA15 la misma cantidad, se empiece a tomar datos. El operador también deberá especificar la cantidad de muestras previas, la cual se transforma en el OFFSET que sale por el puerto PS 0 y tres bits de PS 1. Por este último puerto se envía los bits de control con los valores E = "1", PT = "1" e INI = "0". Estos valores (ver Figura 1) hacen que el circuito de control establezca sus circuitos internos adecuadamente para este modo. Así el divisor de frecuencia es limpiado, las memorias son colocadas en modo de escritura, así como también son habilitados los registros HOLD de manera que la información proveniente de los registros de desplazamiento ingresen a las memorias. También mediante la señal HABILITADOR se asegura que el circuito de pre-disparo no intervenga por el momento.

Luego, desde el computador se cambia el bit INI a "1" lo que da inicio a las operaciones colocando al circuito de control de manera tal que el divisor por cuatro empiece a trabajar, esto obliga a que el contador de direcciones también lo haga. Así, el interfaz ingresa a ciclos de escritura de memoria en forma indefinida hasta que se produzca un disparo el cual es determinado por el comparador, dado por la señal TC.

Cuando el comparador envía un pulso por la línea TC le indica a la lógica de control que mediante su señal HABILITADOR habilite al circuito de pre-disparo (ver Figura 1). Al mismo tiempo mediante la señal CAPTURA le indica que tome la dirección (que viene del contador de direcciones) en la cual se está grabando actualmente.

Luego de una cantidad de direcciones igual al OFFSET, el circuito de pre-disparo genera un pulso en su señal ALTO, la cual ingresa al circuito de control y éste produce la puesta a "1" de las señales A, B y C del divisor de frecuencia; con esto la señal C se detiene no ofreciendo más pulsos y habilita, mediante la compuerta AND, a que la señal GCK haga las veces de reloj para el contador de direcciones (ver Figura 1). También la lógica de control hace que las memorias queden totalmente habilitadas para su lectura, los registros HOLD vayan a alta impedancia, facilitando así la salida de los datos de las memorias (ver Figura 1) a la vez que pide la interrupción 9 de hardware (IRQ 9). Es decir deja, como en los otros modos de disparo, todo listo para la transferencia de información entre el interfaz y el computador, esto se explica posteriormente en este mismo capítulo. Cuando termina la transferencia de datos, el computador coloca nuevamente $E = "1"$, $PT = "0"$ e $INI = "0"$; por lo tanto el circuito de control establece lo mismo que se indicó anteriormente al inicio del modo de disparo manual.

2.8.- Transferencia de datos hacia el Computador

A diferencia del momento en que se escriben los datos en memoria (el cual usa una base de tiempos interna del interfaz), en el momento en que se leen éstos, debe haber sincronismo con las instrucciones de entrada que se ejecutan en el computador. Por lo tanto, para poder sincronizar la transferencia de datos entre el interfaz y el computador, y debido a que el único dispositivo que direcciona a las memorias es el contador de direcciones entonces, se ha obtenido por cambiar solamente la señal interna del reloj (que es

la señal C de 5Mhz) por otra que provenga del circuito de interfaz. Esto es posible debido a que cuando en el interfaz se decodifica uno de los puertos de entrada, cualquiera que sea, se produce una señal de hardware. Esta señal es GCK (ver Figura 1), la cual sirve de reloj al contador, de manera que cuando el computador toma un dato del interfaz, también incrementa en uno la dirección en la memoria, dejándolo listo para que cuando se tome otro dato este lo haga desde la dirección siguiente.

El modo de hacer la transferencia de información desde el interfaz hacia el computador es el siguiente (ver Figura 1):

1. Se ejecuta la instrucción de ingreso de un byte mediante el software usando la dirección de puerto de entrada PE 0.
2. Esta instrucción activa al hardware del interfaz, específicamente al decodificador, el cual activa al Puerto PE 0.
3. Así los 8 bits de la memoria seleccionada (RAM 0), ingresan al bus interno de datos del interfaz y de allí hacia el bus de datos del computador. Al finalizar el ciclo de escritura de éstos 8 bits en el computador se produce un pulso en la línea GCK, el cual obliga al contador de direcciones a cambiar en uno.
4. Se vuelve a realizar los pasos 1, 2 y 3 usando la misma dirección de puerto (PE 0), hasta que se lean todos los datos adquiridos en la RAM 0, es decir 2048 veces.
5. Cambiamos la dirección de puerto y volvemos a los pasos 1, 2, 3 y 4.

En la secuencia de eventos mencionada anteriormente se empieza con la dirección de puerto PE 0 luego PE 1; PE 2 y se termina con PE 3. De esta manera todos los datos, es decir

los 8K bytes pasan a la memoria del computador para su selección y ordenamiento en ocho canales independientes así como su representación en la pantalla del computador.

Cuando se dice que se está leyendo a una memoria en realidad se está leyendo a todas las memorias, debido a que todas las memorias tienen el mismo bus de direcciones y a que el circuito de control habilita a todas las memorias y todas ellas están dispuestas en modo de lectura. A pesar de esto, solo una se decodifica y pasa a través de su puerto respectivo al bus interno y de allí a bus de datos del computador.

La transferencia de datos desde el interfaz al computador personal, es la misma para todos los modos de disparo, ya que este proceso lo maneja el mismo manejador de interrupciones; el cual se activa cada vez que ocurre una interrupción la cual ingresa por la línea IRQ9.

La tarea del manejador de interrupciones es la de leer a todas las memorias RAM del interfaz, separar los datos (recordemos que en cada dirección de memoria existen datos de dos canales) y almacenarlos por separado en 8 arreglos (arrays) de datos.

En este capítulo hemos estudiado en bloques las partes constitutivas del hardware del proyecto motivo de esta tesis, así como se ha descrito los modos de disparo y la transferencia de información desde el interfaz al computador personal. En el próximo capítulo se detallará el hardware del sistema, es decir la implementación electrónica del interfaz con sus respectivos diagramas de tiempos.

CAPITULO III DISEÑO DEL HARDWARE DEL SISTEMA

En el capítulo precedente se ha estudiado en bloques el hardware del proyecto y su funcionamiento. En este capítulo se estudiará el circuito, citando los diferentes elementos que han sido empleados, incluyendo los diagramas circuitales de cada parte. Se incidirá básicamente en los diagramas de tiempos, que hacen posible el funcionamiento acertado del hardware.

Debemos mencionar que en el cálculo de los retardos se ha trabajado con los valores típicos proporcionados en los manuales, los mismos que han sido considerados en los diagramas de tiempos, para un mejor entendimiento de los mismos. En el Apéndice E, se encuentran los datos técnicos de los circuitos empleados en el presente proyecto. En el Apéndice D, se encuentra el diagrama del circuito completo del interfaz.

3.1.- El Circuito de Comunicación con el Computador

La Figura 4 muestra el diagrama de este circuito, en la cual podemos observar que la etapa de decodificación de las direcciones del interfaz, se utiliza una compuerta NAND de 4 entradas 74F20 (U1), un decodificador en línea 74LS138 (U2) y un decodificador 74F139 (U7A y U7B).

Cada vez que ocurra la combinación de bits del bus de direcciones del Computador como sigue:

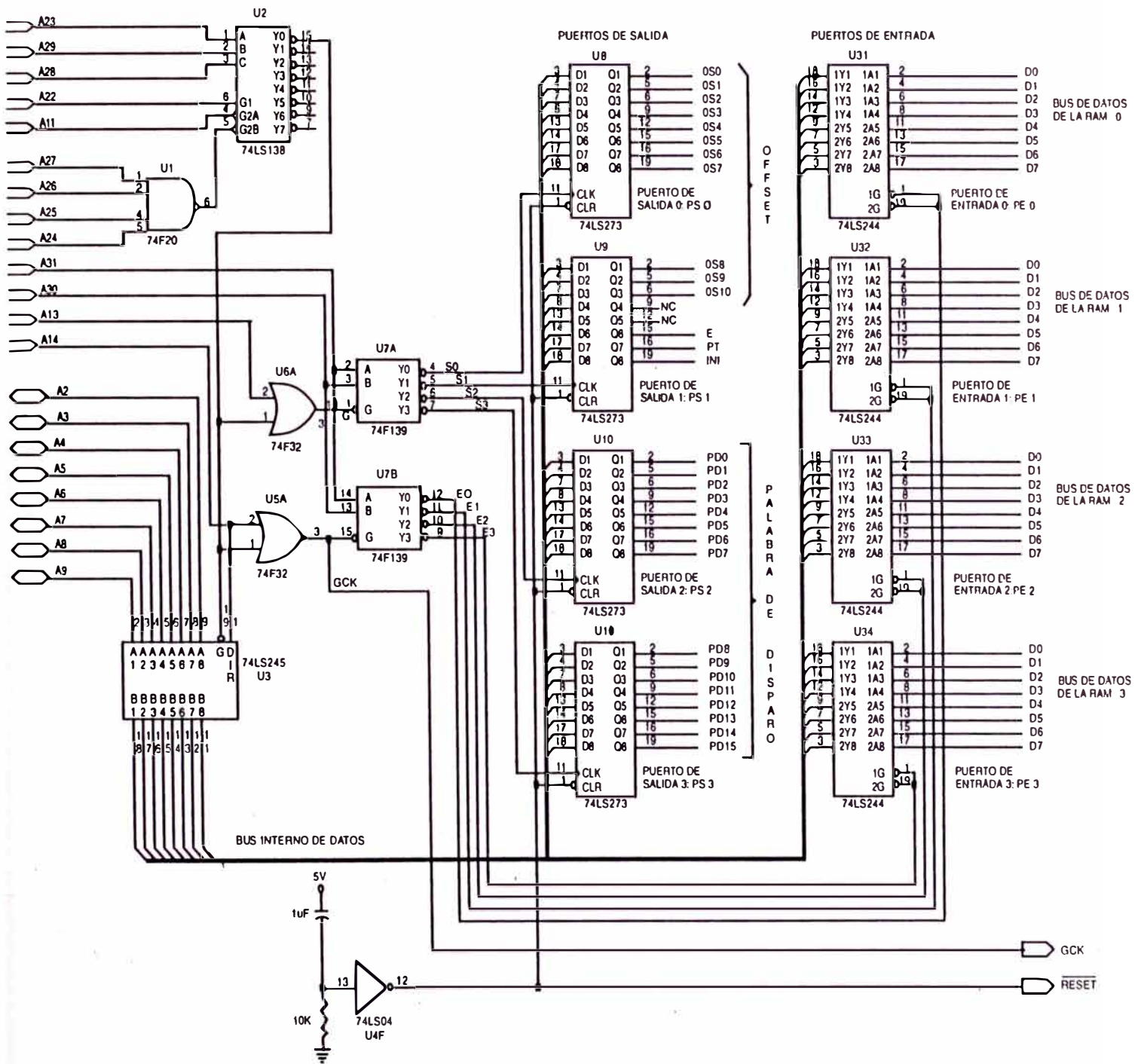


Figura 4. Diagrama del Circuito de Comunicación con el Computador Personal.

AEN BA9 BA8 BA7 BA6 BA5 BA4 BA3 BA2 BA1 BA0
 0 1 0 1 1 1 1 0 0 X X

se decodifica una de las siguientes direcciones de puerto 2F0h, 2F1h, 2F2h y 2F3h. Cada una de estas direcciones se usa para un puerto de salida y un puerto de entrada, como se muestra a continuación:

- 2F0h: Para direccionar al puerto de salida PS 0 y al puerto de entrada PE 0.
- 2F1h: Para direccionar al puerto de salida PS 1 y al puerto de entrada PE 1.
- 2F2h: Para direccionar al puerto de salida PS 2 y al puerto de entrada PE 2.
- 2F3h: Para direccionar al puerto de salida PS 3 y al puerto de entrada PE 3.

La señal AEN ha sido considerada dentro de la decodificación porque así se previene la decodificación inválida de alguno de los cuatro puertos durante ciclos DMA (Acceso Directo a Memoria) que podrían suceder en el computador.

Cuando se use cualquiera de las cuatro direcciones citadas (para lectura o escritura de puerto), se producirá un "0" en la salida Y0 del 74LS138 (U2), el cual habilita al transceptor 74LS245 (U3), que está destinado a interconectar el bus interno del interfaz con el bus de datos del computador. Las señales del computador IOW y IOR, se usan para la escritura en un puerto de salida o lectura de un puerto de entrada, respectivamente. La señal IOR se usa adicionalmente para fijar la dirección de los datos en el transceptor (74LS245). Ya que el 74F139 contiene dos decodificadores en línea de 2 por 4, uno (U7A) es utilizado para decodificar los 4 puertos salida y el otro (U7B) para decodificar los puertos de entrada.

Los diagramas de tiempos tanto para la escritura a un puerto de salida o para la lectura de un puerto de entrada han sido realizada tomando en cuenta las señales eléctricas de temporización del computador personal AT, es decir para los microprocesadores 80286, 80386 ó 80486, no obstante también es válido para el computador XT es decir para el microprocesador 8088.

Debido a diferencias significativas en velocidad y tecnología de dichos microprocesadores y de los circuitos de soporte, es imposible tener una sola temporización del bus que sea válido para todas las implementaciones del PC AT. Sin embargo el bus en el PC AT es síncrono, lo que significa que todas las señales de temporización de control, de datos y de dirección son relativas a reloj del procesador.

La tabla N° 3 muestra el número de pulsos de reloj así como el número de estados de espera para los tipos de transferencias de datos de la mayor parte de diseños de PC's AT. No obstante para diseños de PC de alta velocidad de reloj, se inserta estados de espera adicionales para el bus I/O. Esto asegura compatibilidad para nuestra tarjeta de interfaz, para ser instalada en máquinas modernas.

Tipo de Transferencia de Bus	Número de Ciclos Reloj	Número de estados de espera
8 bits a 8 bits	6	4
16 bits a 8 bits	12	10
16 bits a 16 bits	3	1

Tabla N° 3. Ciclos de reloj y estados de espera para la PC AT

Para nuestro caso vamos a considerar el primero de la tabla, ya que el presente proyecto trata de un tipo de transferencia de bus de 8 bits a 8 bits, y vamos a considerar una frecuencia de 16Mhz, es decir un periodo de reloj de 62.5ns.

3.2.1.- Escritura a un Puerto

Cuando \overline{IOW} e $\overline{Y0}$ son "0" lógico, se habilita los puertos de salida, que usan los bits BA0 y BA1 del bus de direcciones del computador para seleccionar a uno de los cuatro existentes. Cada una de las salidas del decodificador 74F139, U7A, las que denominamos S0, S1, S2 y S3, se emplean como señales de reloj para los 4 puertos de salida (PS 0, PS 1, PS 2 o PS 3).

El C.I. 74LS273, U8, es el puerto de salida PS 0, por el cual salen los bits OS0..OS7 que es parte del OFFSET, el C.I 74LS273, U9, es el puerto de salida PS 1, por el cual

sus tres bits menos significativos salen las señales OS8, OS9 y OS10 que junto con los bits del puerto PS 0, forman la información completa del OFFSET; por los bits más significativos del puerto PS 1 salen las señales de control E, PT e INI. El C.I 74LS273, U10, es el puerto de salida PS 2, por el que salen los bits menos significativos de la palabra de disparo, estos se han etiquetado como PD0 hasta PD7. El C.I 74LS273, U11, es el puerto de salida PS 3, por el que salen los bits más significativos de la palabra de disparo, estos se han etiquetado como PD8 hasta PD15 (ver Figura 4).

En el diagrama de tiempos de la Figura 5 se observa las señales que se activan en el ciclo de escritura a un puerto de salida. En el diagrama podemos observar que la señal $\overline{Y0}$ va a "0" luego de 28.2ns después de que la dirección de puerto ha sido estabilizada.

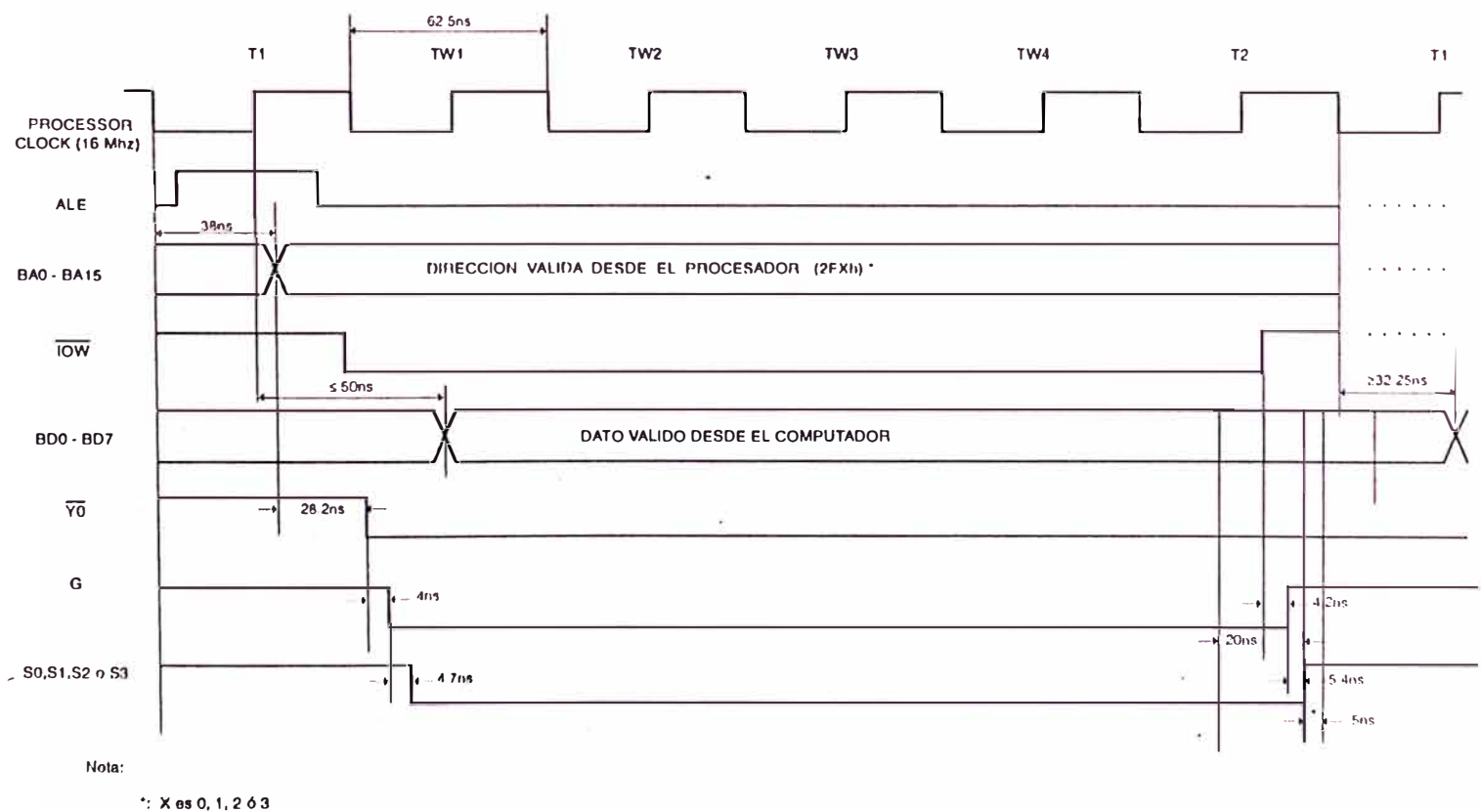


Figura 5. Diagrama de tiempos del ciclo de escritura de un puerto de salida.

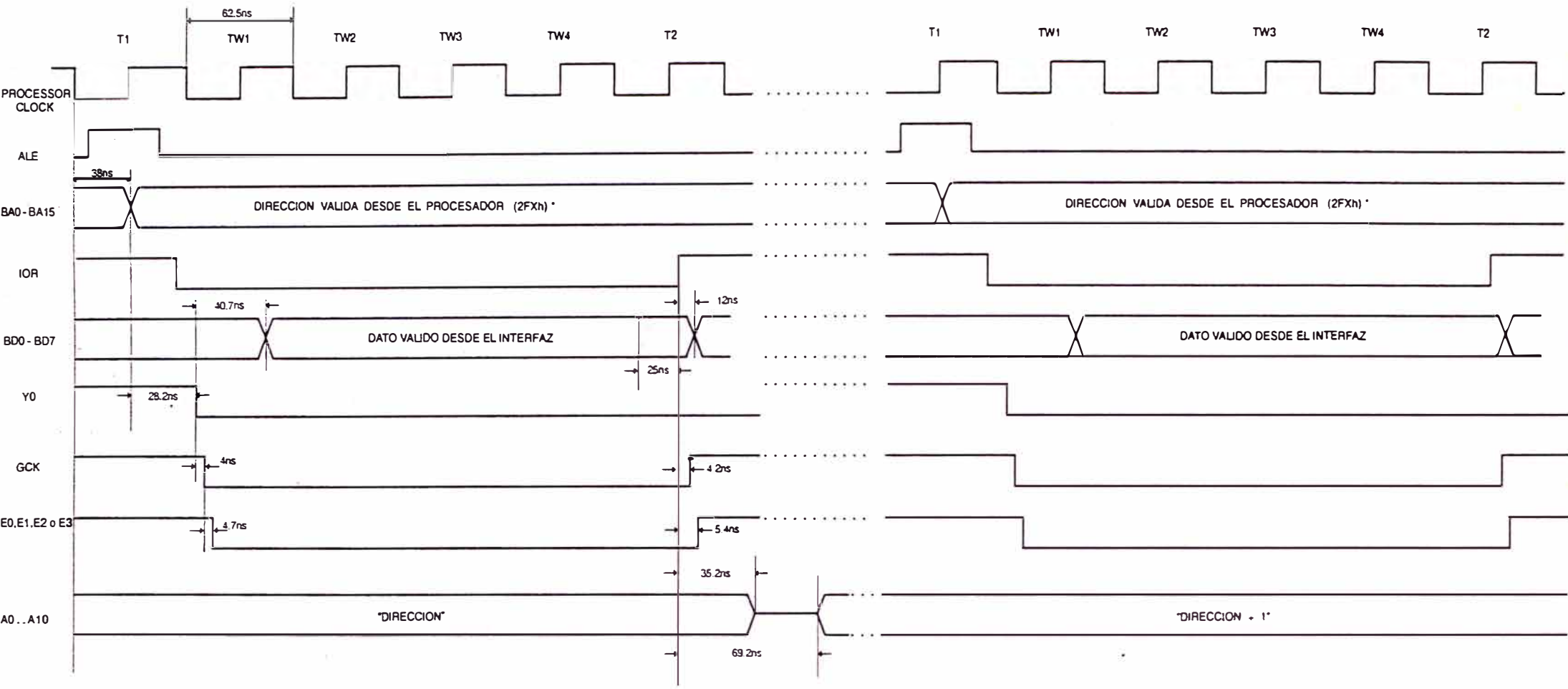
Los datos son escritos en el flanco positivo de S0, S1, S2 o S3, es decir al término del ciclo de escritura. Así los cuatro registros 74LS273, usados como puertos de salida, tienen el tiempo suficiente para poder tomar el dato. Para poder asegurar una perfecta grabación del dato del CPU en el registro, tendremos que revisar los parámetros Setup time y Data hold time del registro 74LS273. Ellos exigen como mínimo 20ns de Set-Up time y 5ns de Data hold time; (intervalo sombreado) y como podemos ver en el gráfico de la Figura 5, tenemos tiempo de realizar la grabación de la información en forma segura.

3.1.2.- Lectura desde un Puerto

Cuando ambas IOR e Y0 son "0", se habilitan los puertos de entrada, que usa los bits BA0 y BA1 del bus de direcciones del computador personal para seleccionar a uno de los cuatro puertos de entrada. Cada una de las salidas de este decodificador 74F139, U7B (ver Figura 4), las cuales son etiquetadas como E0, E1, E2 y E3, se usan para habilitar a los 4 buffers octales tri-state 74LS244 que representan a los puertos de entrada. Estos comunican el bus de datos de cada una de las memorias con el bus interno del interfaz.

En el diagrama de tiempos de la Figura 6 se muestra las señales que se activan cuando ocurre una lectura de un puerto.

La señal GCK de habilitación de cualquiera de los puertos de entrada (ver Figura 4), se usa como señal de reloj para el momento de la transferencia de datos entre el computador y el interfaz. Esta señal siempre está en "1" cuando no se decodifica ningún puerto. Cuando la señal GCK entrega un flanco positivo, el cual se dá al final de la lectura, produce el cambio en uno del bus de direcciones interno A0..A10 (en la Figura 6 cambia de "DIRECCION" a "DIRECCION+1"), ya que este flanco ingresa a la entrada de reloj



LECTURA DESDE EL PUERTO DE ENTRADA 2Fxm *

Nota:

*: X es 0, 2 o 3

Figura 6. Diagrama de tiempos del ciclo de lectura de un puerto de entrada.

del circuito contador de direcciones, como veremos más adelante en la sección 3.2. Así queda todo listo para que en una siguiente lectura se tome el dato de otra dirección.

El dato válido suministrado por interfaz, es colocado en el bus de datos del computador después de 40.7ns de haber Y_0 cambiado a nivel bajo. En el cálculo de este tiempo de retardo, se está considerando los circuitos integrados 74F32, 74F139, 74LS244 y el 74LS245, sin considerar el tiempo de acceso de la memoria 6116, debido a que la memoria ya posee un dato válido en su bus cuando se produce la lectura del puerto. El tiempo de acceso de la memoria es considerada entre lectura y lectura del puerto, y no tiene importancia si el tiempo que existe entre dos lecturas consecutivas de puerto, es mayor que 150ns (tiempo de acceso de la memoria). En el diseño de la rutina del manejador de interrupciones (que es la rutina que realiza el ingreso de la información desde el interfaz hacia el computador), se ha agregado instrucciones NOP para que la separación entre las lecturas sea mayor que el tiempo de acceso de la memoria.

El hecho de prescindir del tiempo de acceso de la memoria, en el cálculo del retardo del dato válido desde el interfaz al computador (sólo es de 40.7ns), nos da la seguridad de una correcta lectura inclusive para modernos computadores personales que trabajan con velocidades elevadas de reloj.

Para asegurar una lectura válida desde el slot de expansión, el diseño del computador exige que el dato debe estar válido en el bus un mínimo de 25ns (intervalo sombreado) antes del flanco de subida de la señal IOR, lo cual se está cumpliendo ya que como se mencionó anteriormente, el dato ya es válido a partir de 40.7ns de que Y_0 cambia a "0". La información desde el interfaz se mantiene 12ns

después que la señal IOR va a “1”, este tiempo es suficiente para el hold time del procesador que es de 1ns. Así aseguramos una perfecta lectura de los datos desde el interfaz hacia el computador.

Un circuito basado en una resistencia de 10K, un capacitor de 1uF y un inversor 74S04, se usa como RESET inicial de todo el sistema. Esta señal se va a “0” solo cada vez que se enciende el computador y dura aproximadamente 10mseg ($T = 1\mu F * 10K$), luego va a “1” y permanece inalterable. El objeto de esta señal RESET es limpiar en el encendido del computador todos los registros de salida y otros dispositivos de la lógica de control que señalaremos luego. Así se logra que las señales de control del puerto PS 1 (E, PT e INI) estén en “0”, cada vez que se enciende el computador.

3.2.- EL Circuito de Temporización

El circuito de Temporización se muestra en la Figura 7 y se compone de un oscilador cuya salida entrega una señal de 20 Mhz y de un ciclo de servicio (duty cycle) del 50%. La señal de este oscilador ingresa a un inversor 74S04, el cual produce la señal de reloj etiquetada por CK1, la cual vuelve a negarse, produciendo la señal CK2, y que es empleada como reloj para el circuito divisor de frecuencia; el mismo que está compuesto por tres flip-flops tipo JK (U22A, U22B y U23B), cada uno de los cuales es sensible al flanco de bajada del reloj.

El diseño de este divisor, se basa prácticamente en un contador escalador con cuenta ...0,7,3,2,0,7,...; de ahí que necesitamos 3 flip-flops. Se eligió flip-flops tipo JK para que la implementación sea bastante simple. La tabla para el diseño del divisor por 4, es la siguiente:

A	B	C	JA	KA	JB	KB	JC	KC
0	0	0	1	X	1	X	1	X
1	1	1	X	1	X	0	X	0
0	1	1	0	X	X	0	X	1
0	1	0	0	X	X	1	0	X

De la cual podemos rápidamente deducir que:

$$\begin{aligned}
 JA &= \bar{B} & JB &= 1 & JC &= \bar{B} \\
 KA &= 1 & KB &= \bar{C} & KC &= \bar{A}
 \end{aligned}$$

En la misma Figura 7 también se muestra el circuito contador de direcciones. El divisor de frecuencia es limpiado (A, B y C a "0") o preseteado (A, B y C a "1"), por las señales que provienen de la lógica de control, CLEAR y $\overline{\text{MODO}}$ respectivamente.

El contador de direcciones está formado por tres 74LS193 (U25, U26 y U27), conectados en cascada con cuyas salidas tenemos los 11 bits del bus de direcciones (A0..A10). La señal INI que proviene del puerto de salida permite el precargado del contador, el dato de precargado es fija e igual a 7FEH.

La salida Carry Out (CO) del 74LS193 más significativo (U27), etiquetado como CARRY, se envía a la lógica de control para poder parar la toma de datos; esta señal es un índice de que ya se han grabado en todas las direcciones de las memorias (desde la 000H a la 7FFH), esto es válido para los modos Manual y Automático.

En cuanto al reloj del contador de direcciones observamos que es la señal \emptyset , la cual proviene de una compuerta AND de dos entradas 74F08 (U21A), a la cual ingresan la señal C y la señal GCK. Cuando se está grabando datos en las memorias la señal GCK está en "1", por lo tanto \emptyset es igual a la señal C del divisor. Y cuando se está en el modo de lectura de memoria, es decir

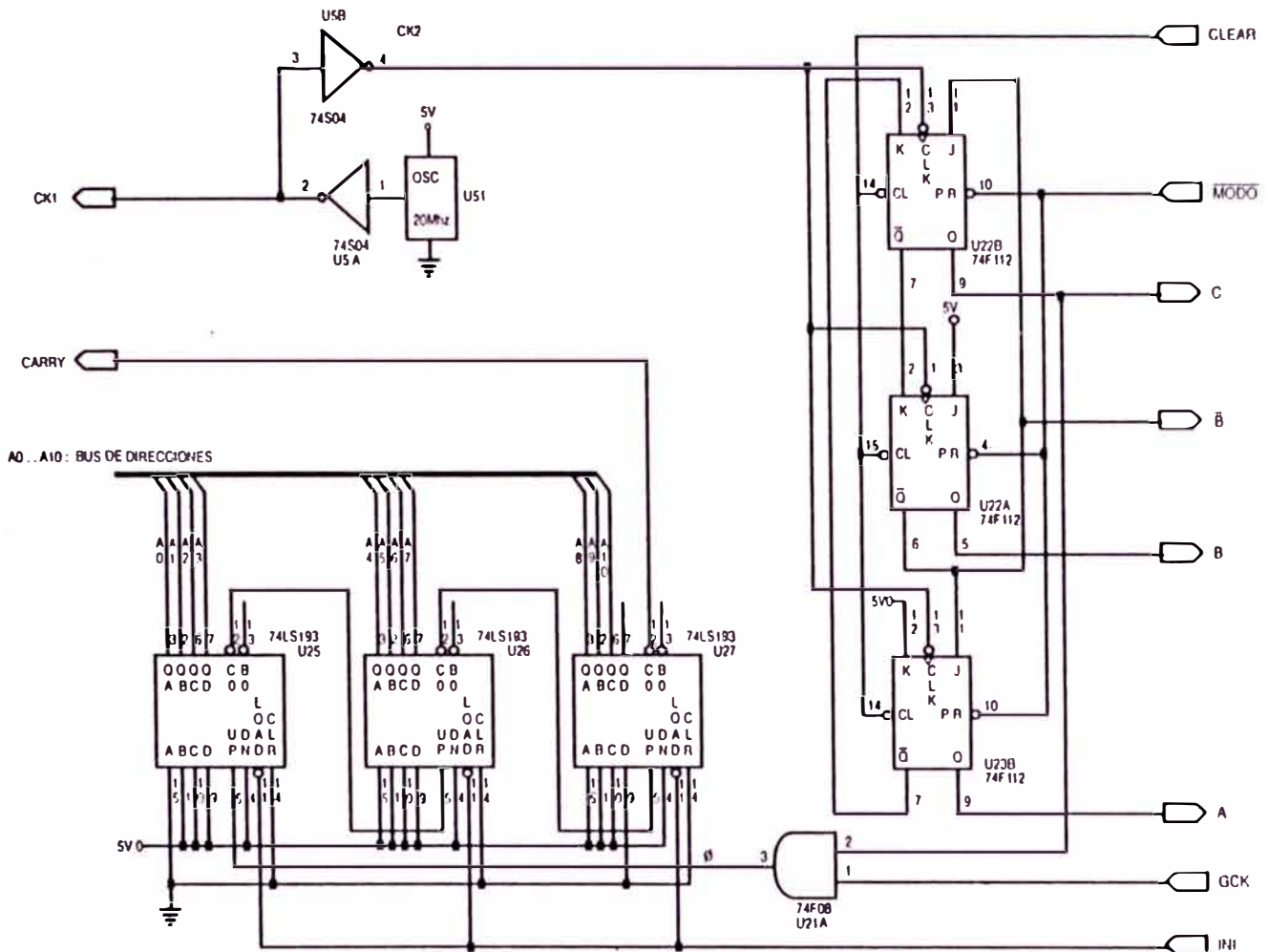


Figura 7. Diagrama del circuito de Temporización.

MODO está en "0", se presetea al divisor de frecuencia, lo que indica que la señal C está en "1"; por lo tanto \emptyset es igual a GCK que es la señal que se encarga de hacer los pulsos de reloj cada vez que se introduce un dato desde el interfaz hacia la memoria del computador.

En la Figura 8 se muestra el diagrama de tiempos de las señales del divisor de frecuencia. La Figura 8a, muestra el diagrama de tiempos de las señales del divisor de frecuencia, cuando se empieza a tomar datos, para cualquier modo de disparo. Como se aprecia, el bus de direcciones generado por el contador empieza desde 7FEh lo cual es válido para cualquier modo de disparo, la razón de esto es que los primeros datos válidos

se graben en la dirección 000h. También se muestra que las señales A, B y C parten de cero; todas estas señales tienen un retardo de propagación de 5ns (del C.I 74F112) respecto a CK2. La señal \emptyset tiene un retardo de 4.2ns (del C.I 74F08 U21A) respecto a la señal C. El bus de direcciones, que es suministrado por el contador, entrega una dirección estable luego de, en el peor caso, 65ns respecto a \emptyset .

Decimos que 65ns es en el peor de los casos, porque existirán direcciones donde solamente cambie su cuenta el 74LS193 (U25) que contiene los bits menos significativos; en ese caso solo se tomará en cuenta su retardo (31ns). Respecto a cualquiera de las señales A, B o C, el bus de direcciones es estable luego de 69.2ns, como se puede deducir fácilmente del gráfico.

La dirección permanece válida luego de 31ns de que se produce otro pulso positivo de \emptyset , que es el tiempo de retardo de propagación de las señales Q del 74LS193 (U25), que contiene los bits menos significativos, respecto a \emptyset ; y respecto a cualquiera de las señales A, B o C, el bus de direcciones deja de ser estable luego de 35.2ns, como se muestra en la Figura 8.

De los 12 pines del contador solo se usan 11, por lo que, cuando es inicializado en 7FEH, con el primer pulso cambia a 7FFH, con el segundo cambia a 800H, pero como al bus ingresan 11 pines en ellos se observa el número 000H.

La Figura 8b muestra como se activa la señal CARRY, la cual es usada para los modos de disparo Manual y Automático. La señal de CARRY inicia en "1" lógico y va a "0" cuando el contador ingresa a su cuenta FFFH (para 11 bits : 7FFH), luego de 63ns del pulso de bajada de \emptyset . CARRY vuelve a subir cuando el pulso de subida de \emptyset hace cambiar de FFFH a 000H, luego de 51ns.

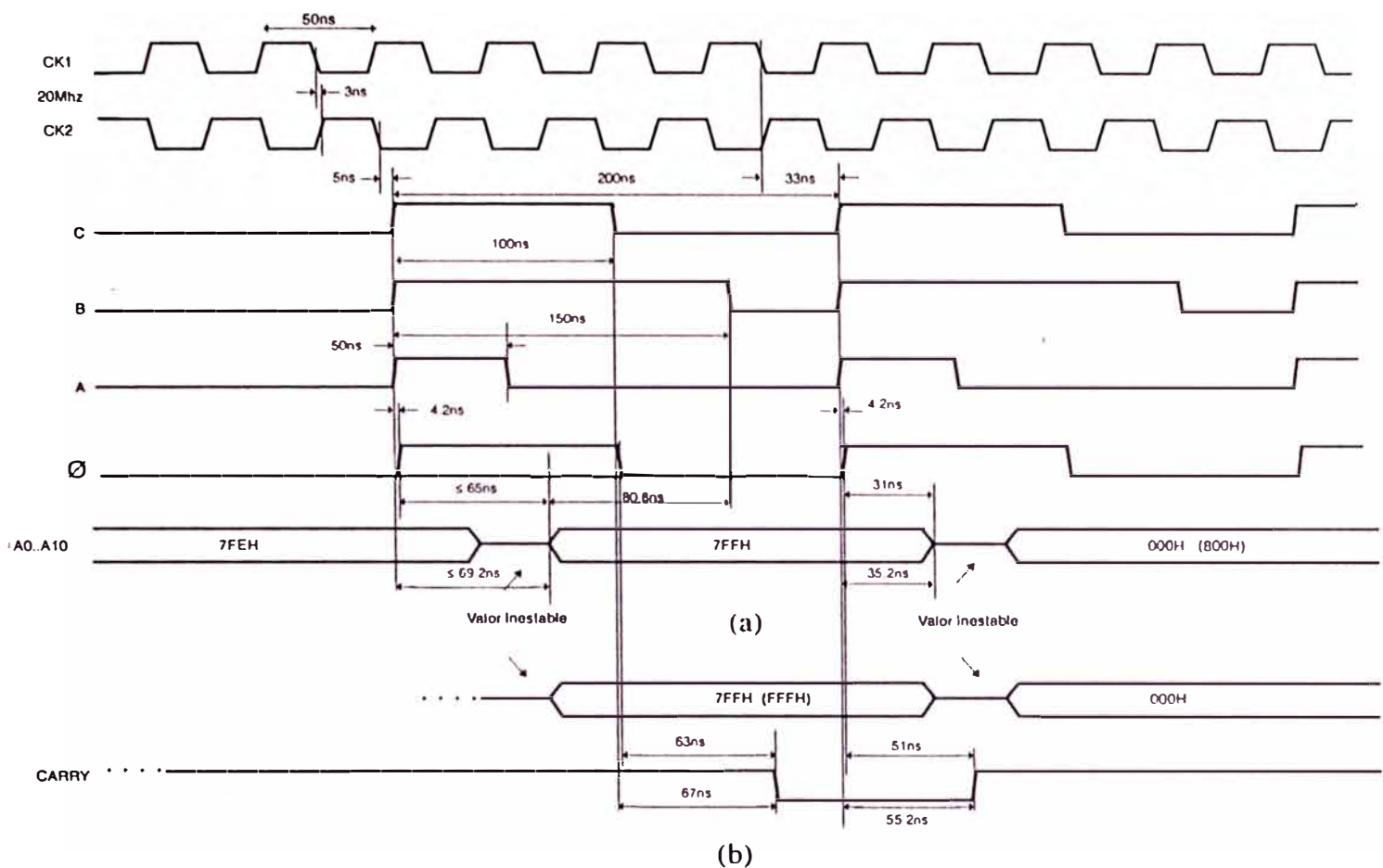


Figura 8. Diagrama de tiempos del divisor de frecuencia y del contador de direcciones

3.3.- El Circuito de Control

En la Figura 9 se muestra el diagrama del circuito de la lógica de control. Como se observa, se cuenta con un C.I dual flip-flop tipo D sensible al flanco de subida 74S74, uno de los cuales, U24 A, es el flip-flop de MODO o de INTERRUPCION y el otro, U24 B, se emplea para el DISPARO. El flip-flop JK 74F112 se utiliza como el flip-flop de CAPTURA; el resto del circuito son compuertas AND 74F08, OR 74F32 y un inversor 74S04.

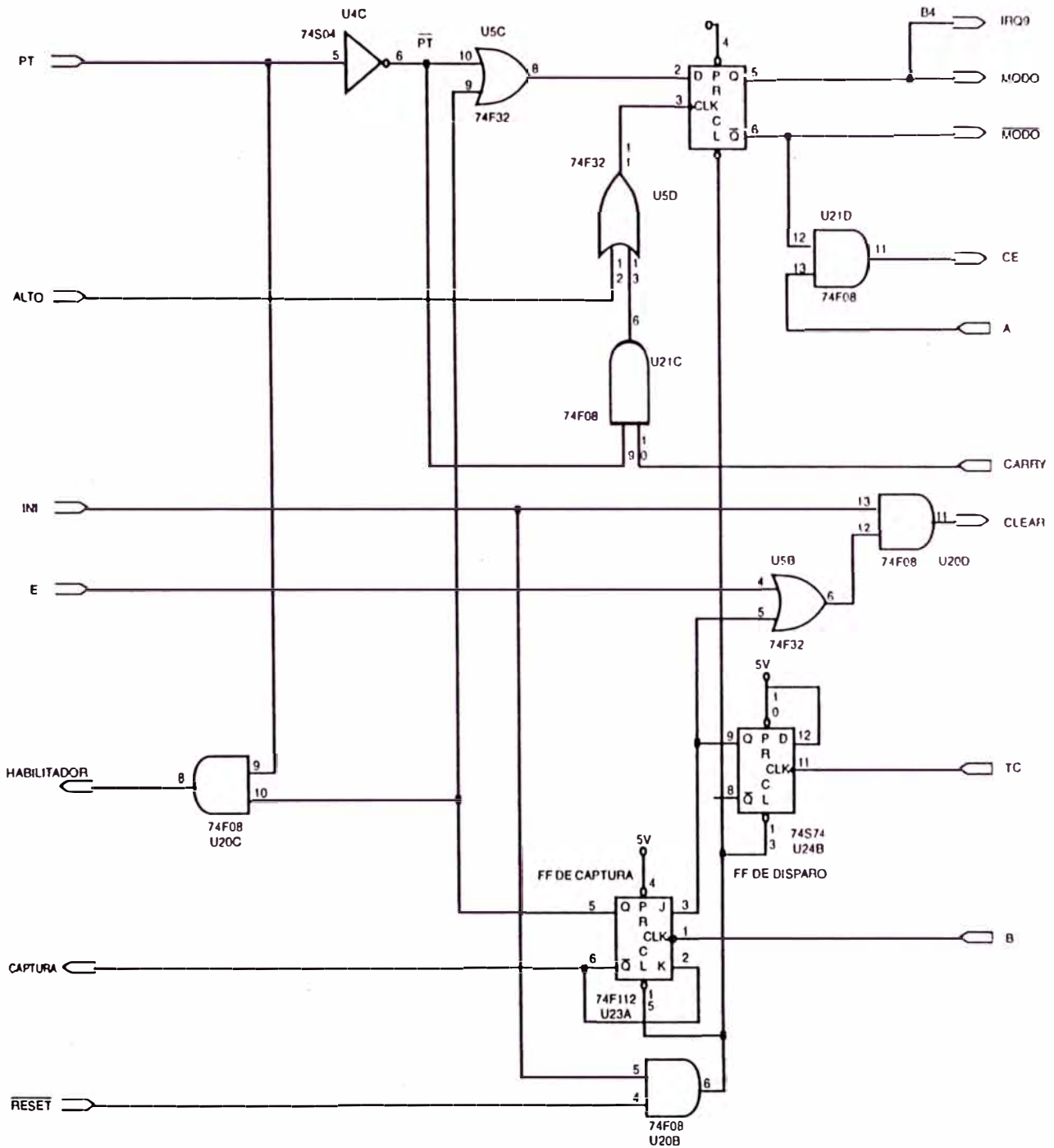


Figura 9. Diagrama del circuito de la Lógica de control

Ya que este circuito es el eje de todo el interfaz, revisaremos cada una de sus entradas y salidas empezando por las señales que ingresan al circuito de control, estas son las siguientes (ver Figura 9):

- PT:** Señal de entrada que cuando es "1" indica al control que existe pre-disparo. Viene desde el puerto PS 1.
- E:** Señal que cuando es "0" le indica al control que el modo Automático debe ser activo. Proviene desde el puerto PS 1.
- INI:** Señal que cuando es "1" indica el arranque de cualquier modo de disparo.
- ALTO:** Señal que viene del circuito de predisparo, cuando es "1" le indica al control que debe detener el muestreo. Solo es usado en el modo Con Pre-Disparo y siempre permanece en "0" cuando no se trabaja en ese modo.
- RESET:** Señal que solamente en el encendido del computador es "0" durante 10ms, lo que limpia a los flip-flops internos del circuito de control. Luego de los 10ms permanece inalterable en "1".
- TC:** Es una señal que viene del circuito Comparador. Un pulso positivo en él, le dice al control que ha ocurrido un disparo; esto es válido para los modos Automático y Con Pre-Disparo.
- A:** Esta señal proviene del divisor de frecuencia y le sirve al circuito de control para la habilitación de las memorias RAM's cuando se está grabando información en ellas.
- B:** Esta señal también viene del divisor de frecuencia. Un pulso negativo en ella indica al control que puede dar la orden al circuito de pre-disparo para que se capture una dirección válida; esto es cierto si es que previamente en TC ha ocurrido un pulso positivo. la señal B es usada en el modo Con Pre-Disparo.
- CARRY:** Señal que viene del contador de direcciones y sirve para detener la secuencia de escritura en memorias RAM en los modos Manual y Automático.

Con estas señales de entrada, el circuito de control produce otras de salida, las cuales describimos a continuación:

- CAPTURA:** Es una señal de salida que va al circuito de pre-disparo, al cual le indica con un pulso negativo que “capture” la dirección proveniente del contador de direcciones. Válido solo en el modo de pre-disparo.
- HABILITADOR:** Es una señal que va también al circuito de pre-disparo y cuando es “1” le indica, a este circuito, que ya puede realizar su comparación interna y así poder producir su señal ALTO.
- CLEAR:** Esta señal limpia a los flip-flops que forman el divisor de frecuencia, es decir cuando CLEAR es “0” las señales A, B y C son hechas “0” y cuando CLEAR pasa a “1” A, B y C empiezan a oscilar, es decir con esta señal, el circuito de control, da inicio a la toma de información en cualquiera de los modos.
- CE:** El circuito de control hace que esta señal sea igual que la señal A cuando se está grabando información en las memorias; y cuando se las lee, la señal CE es colocada en “1”.
- MODO o IRQ9:** Esta señal es “0” cuando se escribe en las memorias y es “1” cuando se las lee, esto se debe a que va cableado directamente a la entrada WE de cada una de las memorias. Esta misma señal sirve para pedir la interrupción al computador personal.
- MODO:** Es la negación de la señal anterior y sirve para colocar en “1” a los flip-flops que constituyen el divisor de frecuencia (señales A=B=C=“1”). Es decir que la señal MODO indica el término de la secuencia de escritura de datos en las memorias.

A continuación explicaremos que es lo que ocurre con la unidad de control, para cada uno de los modos de operación.

3.3.1.- Modo Manual

Para este modo de trabajo las señales de control que provienen del puerto de salida PS 1 (cuya dirección es 2F1h), se colocan previamente por orden del computador en los siguientes valores $PT = "0"$, $E = "1"$ e $INI = "0"$, esto hace que (ver Figura 9):

Los flip-flops de DISPARO, CAPTURA e INTERRUPCION sean aclarados.

La entrada D del flip-flop de INTERRUPCION esté en "1".

Las señales MODO y CLEAR sean colocadas en "0".

CE es igual a la señal A.

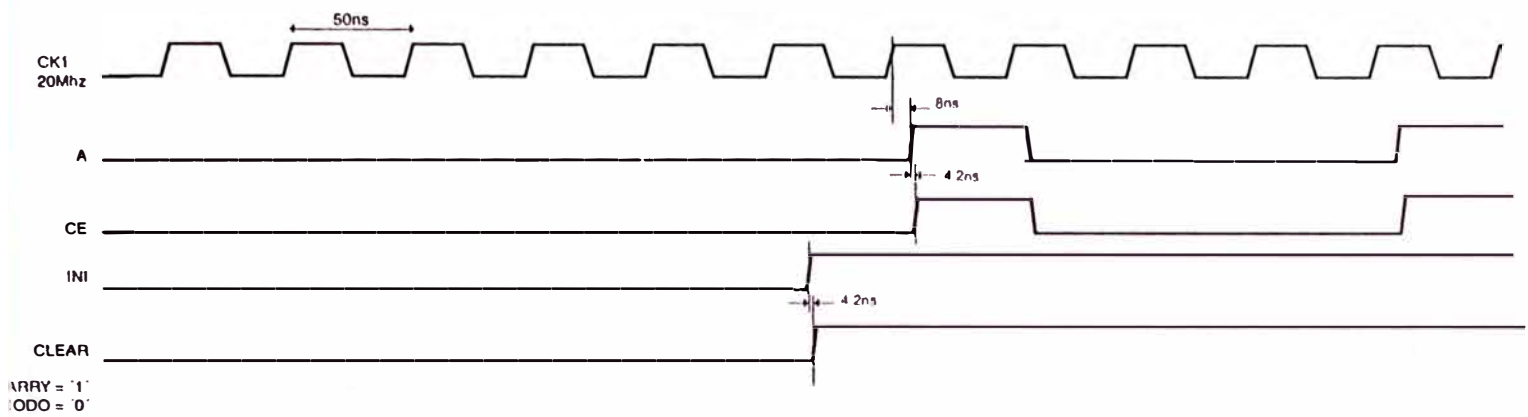
La señal HABILITADOR sea "0". Esto trae como consecuencia que la señal ALTO, que proviene del circuito de pre-disparo esté en "0". La razón de esto la demostraremos cuando se estudie el circuito de pre-disparo.

Luego, desde el computador se le envía $PT = "0"$, $E = "1"$ e $INI = "1"$. Es decir, solo cambia INI, lo que produce que:

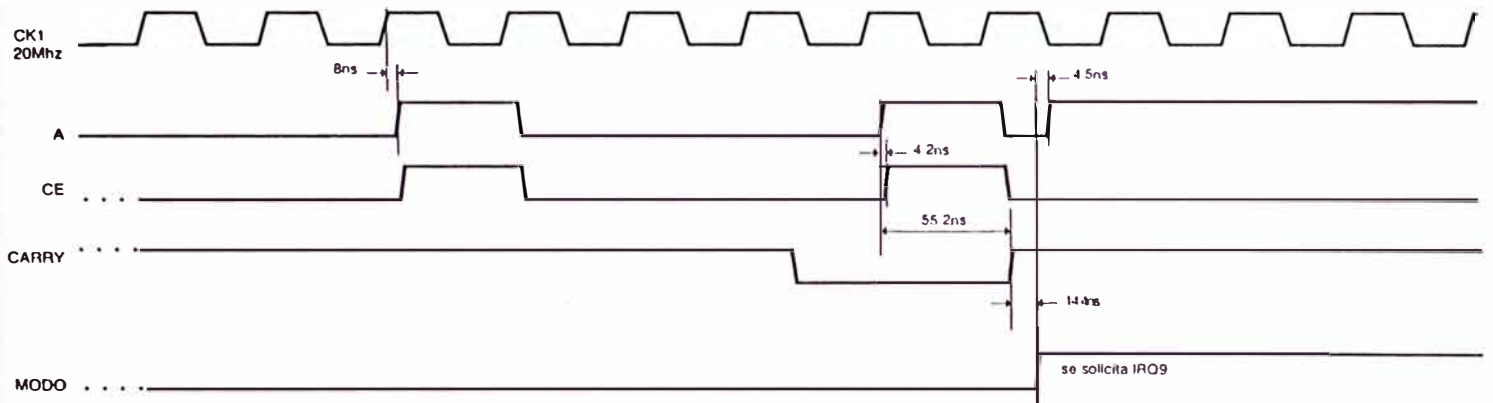
CLEAR se coloque en "1", lo que libera al divisor por cuatro (ver Figura 10a), para que pueda responder a los pulsos negativos de CK2 (invertido de CK1); a partir de ese instante se empieza la toma de datos.

- También el flip-flop de INTERRUPCION queda habilitado para que responda a los pulsos positivos que pudieran existir a la salida de la compuerta 74F32 (USD), esto se debe que su entrada asíncrona CL (clear) es colocado en "1".

Luego de esto, se espera hasta que ocurra el primer pulso de subida de CK1, y es cuando comienza la secuencia de escritura de información a las memorias. La Figura 10a muestra el diagrama de tiempos del inicio del modo manual desde que INI cambia



(a) Inicio del Modo Manual



(b) Fin del Modo Manual

Figura 10. Diagrama de tiempos del Modo de Disparo Manual.

de "0" a "1"; mientras que la Figura 10b nos muestra cuando termina este modo. Esto ocurre cuando CARRY produce el pulso de reloj en el flip-flop de INTERRUPCION, cambiando éste su señal de salida MODO a "1" (ver Figura 9).

Cuando la señal MODO cambia a "1", solicita al computador la interrupción IRQ9 así como también las memorias son colocadas en modo de lectura. Obviamente la señal $\overline{\text{MODO}}$ ha cambiado a "0" así se ordena que las señales A, B y C cambien a "1". En el diagrama de tiempos de la Figura 10b se aprecia a la

señal A, no obstante, le sucede lo mismo a las señales B y C. Cuando MODO cambia a "1", la señal CE es colocada a "0" lo que habilita a las memorias a ser leídas.

3.3.2.- Modo Automático

Para este modo de funcionamiento se escribe previamente en el puerto de salida PS 1 (dirección 2F1h) los bits de control como sigue: PT = "0", E = "0" e INI = "0". Esto origina en la lógica de control lo siguiente (ver Figura 9):

- Los flip-flops de DISPARO, CAPTURA e INTERRUPCION sean aclarados.
- La entrada D del flip-flop INTERRUPCION esté en "1".
- Las señales MODO y CLEAR sean colocadas en "0".
- CE es igual a la señal A.
- La señal HABILITADOR sea "0". Esto trae como consecuencia que la señal ALTO se mantenga en "0" porque el circuito de pre-disparo no está habilitado.

Luego, desde el computador se le envía PT = "0", E = "0" e INI = "1". Es decir solo cambia INI lo que produce que (ver Figura 9):

Los tres flip-flop's: INTERRUPCION, DISPARO y CAPTURA, ahora sean sensibles a los pulsos que puedan suceder en sus respectivas entradas de reloj. Así, el flip-flop de INTERRUPCION queda a la espera del pulso positivo de CARRY en igual forma en que se hizo para el Modo Manual.

- En cuanto a la señal CLEAR que es la que inicio a la toma de datos ya no es iniciada por INI, sino por el pulso positivo de TC que proviene del circuito comparador. Si este pulso ocurre, entonces CLEAR es puesto a "1", empezando así la secuencia de escritura de la información en las memorias.

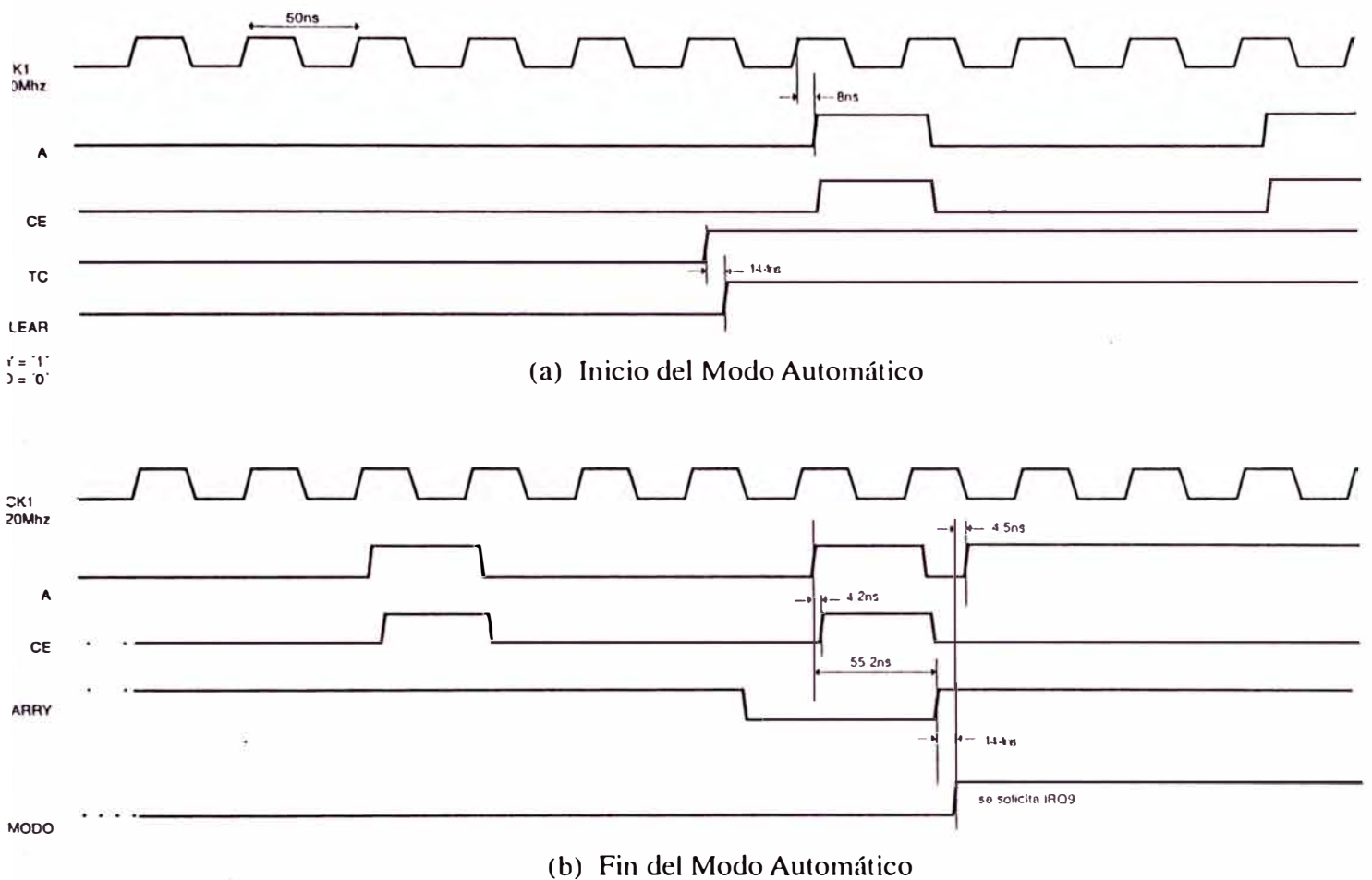


Figura 11. Diagrama de tiempos del Modo de Disparo Automático

Una vez que la señal TC cambia a "1", se espera que ocurra el primer pulso de subida de CK1, empezando la secuencia de escritura de datos a las memorias. La Figura 11a muestra el diagrama de tiempos del inicio del modo Automático desde que TC cambia a "1"; mientras que la Figura 11b nos muestra cuando termina este modo, es decir cuando CARRY produce el pulso de reloj en el flip-flop de INTERRUPCION, cambiando éste su señal de salida MODO a "1" (ver Figura 9).

Cuando la señal MODO cambia a "1", solicita al computador la interrupción IRQ9, y las memorias son colocadas en modo de lectura. Como la señal $\overline{\text{MODO}}$ ha

cambiado a "0", se ordena que las señales A, B y C cambien a "1". Cuando MODO cambia a "1" hace que la señal CE se coloque en "0" habilitando a las memorias a ser leídas.

3.3.3.- Modo Con Pre-Disparo

En este modo, similarmente al modo anterior, también se usa la señal TC para producir el disparo, pero la diferencia radica en que cuando se da la orden INI = "1", ya se empieza a tomar datos. Veamos paso a paso lo que ocurre en este modo.

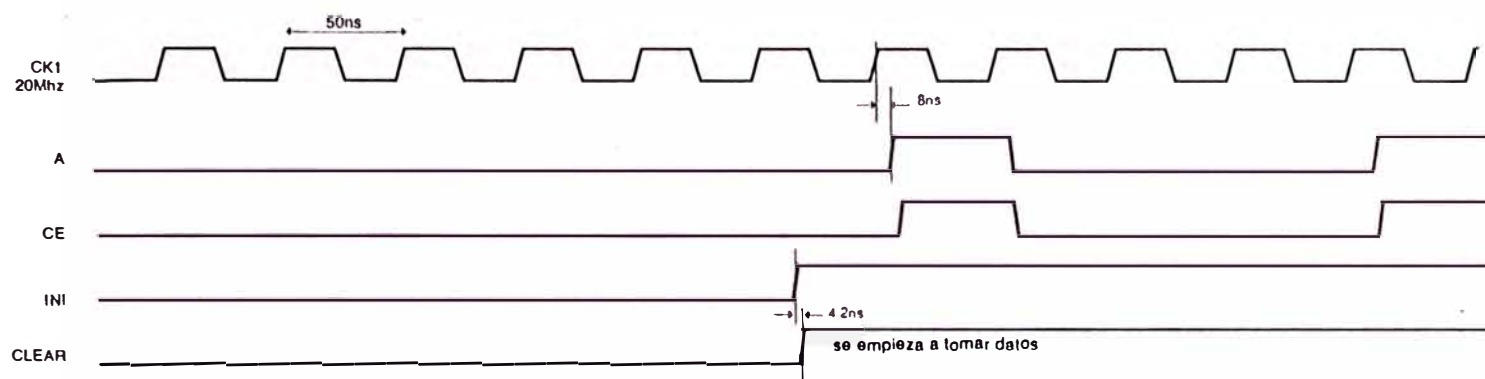
Primero desde el computador se envía al puerto de salida PS 1 (dirección 2F1h) los bits de control PT = "1", E = "1" e INI = "0", esto causa en la lógica de control lo siguiente (ver Figura 9):

- Los flip-flops de DISPARO, CAPTURA e INTERRUPCION son aclarados.
- La entrada D del flip-flop de INTERRUPCION se pone en "0".
- Las señales MODO y CLEAR son colocadas en "0".
- CE es igual a la señal A.
- La señal HABILITADOR es "0". Esto trae como consecuencia que la señal ALTO esté en "0" porque el circuito de pre-disparo no está habilitado.

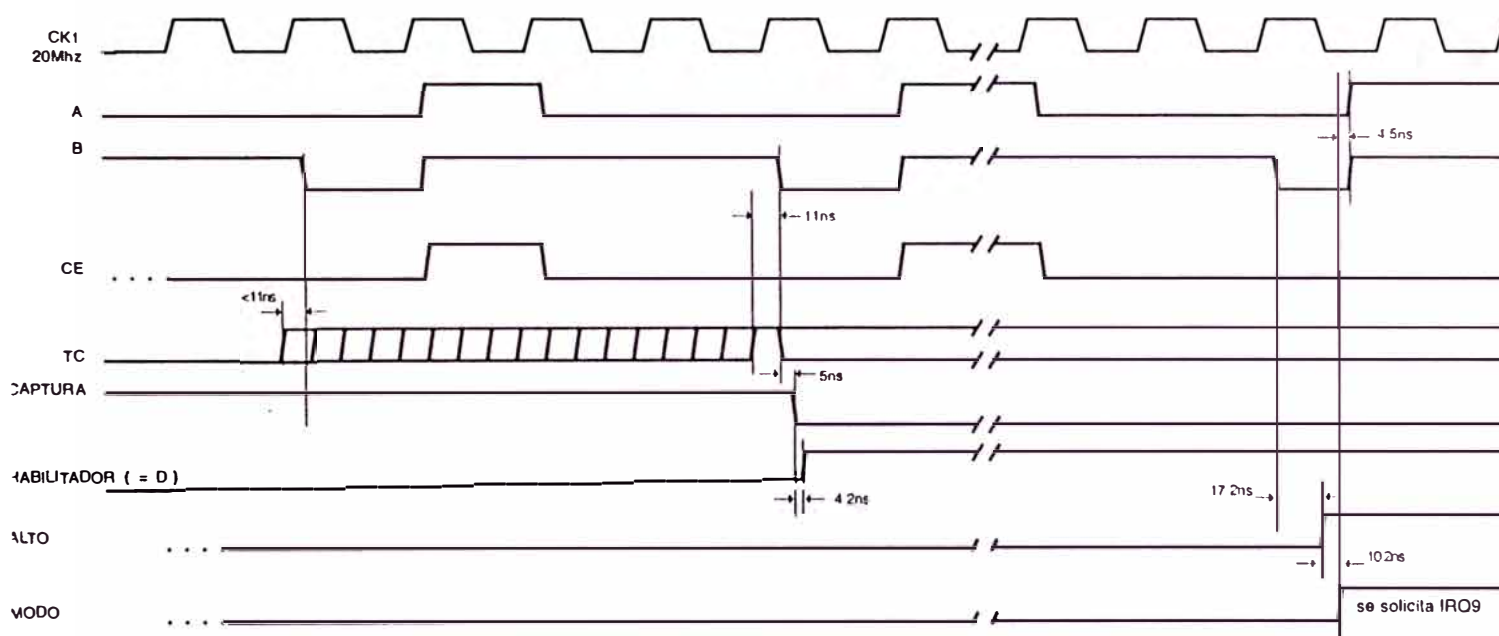
La señal CAPTURA esté inicialmente en "1".

Luego desde el computador se le envía PT = "1", E = "1" e INI = "1". Es decir solo cambia INI lo que produce que:

- Los tres flip-flop's: INTERRUPCION, DISPARO y CAPTURA ahora sean sensibles a los pulsos que puedan suceder en sus respectivas entradas de reloj. Así el flip-flop de INTERRUPCION queda a la espera del pulso positivo de ALTO.



(a) Inicio del modo con Pre-Disparo



(b) Disparo y Fin del modo con Pre-Disparo

Figura 12. Diagrama de tiempos del Modo Con Pre-Disparo.

Esta señal es análoga a la señal CARRY del modo manual y Automático ya que produce el detenimiento de la escritura en memorias.

- CARRY es bloqueado por la compuerta 74F08 (U21C) ya que \overline{PT} es "0".
- CLEAR se coloque en "1", lo que libera al divisor de frecuencia, para un trabajo en modo sincrónico, de manera que empieza la toma de datos.

Hasta aquí el circuito entra a un estado de grabación de los datos en memoria en forma cíclica es decir cada vez que se llenan de datos las memorias, se vuelve a grabar encima borrando los datos anteriormante tomados. El diagrama de tiempos que se muestra en la Figura 12a, se grafica el comienzo de la toma de datos desde que INI cambia a "1" y en la Figura 12b se muestra cuando TC cambia a "1" que es cuando se advierte que ha ocurrido un disparo; este cambio como puede verse en el gráfico debe suceder entre dos pulsos de bajada de la señal B.

El pulso TC hace que la salida del flip-flop de DISPARO vaya a "1" (ver Figura 9) lo que habilita al flip-flop de CAPTURA a que pueda responder cuando un pulso negativo provenga desde la señal B del divisor de frecuencia. Cuando esto ocurre, la salida (Q) del flip-flop de CAPTURA va a "1" esto hace que la señal D del flip-flop de INTERRUPCION esté en "1". La señal CAPTURA envía un pulso negativo al circuito de pre-disparo indicándole que "capture" la dirección proveniente del contador de direcciones. Adicionalmente la señal HABILITADOR va a "1", habilitando al circuito de pre-disparo.

La señal TC debería subir a "1" 11ns (ver Figura 12b) antes de un flanco de bajada de B; este tiempo responde al tiempo de retardo del flip-flop de DISPARO el cual es 6 ns y al tiempo de establecimiento de la señales JK del flip-flop de CAPTURA el mismo que es de 5ns. Luego que TC ha sido capturado, ya puede bajar o mantenerse en "1"; es decir no importa su estado ya que el circuito solo sensa el primer disparo anulando los siguientes pulsos que pudieran haber en TC.

En la Figura 12b también se muestra cuando termina el Modo Con Pre-Disparo, que es cuando la señal ALTO produce el pulso de reloj en el flip-flop de INTERRUPCION, lo que obliga a que su señal de salida MODO cambie a "1", pidiendo de esta forma la interrupción IRQ9 y dejando a las memorias en modo de

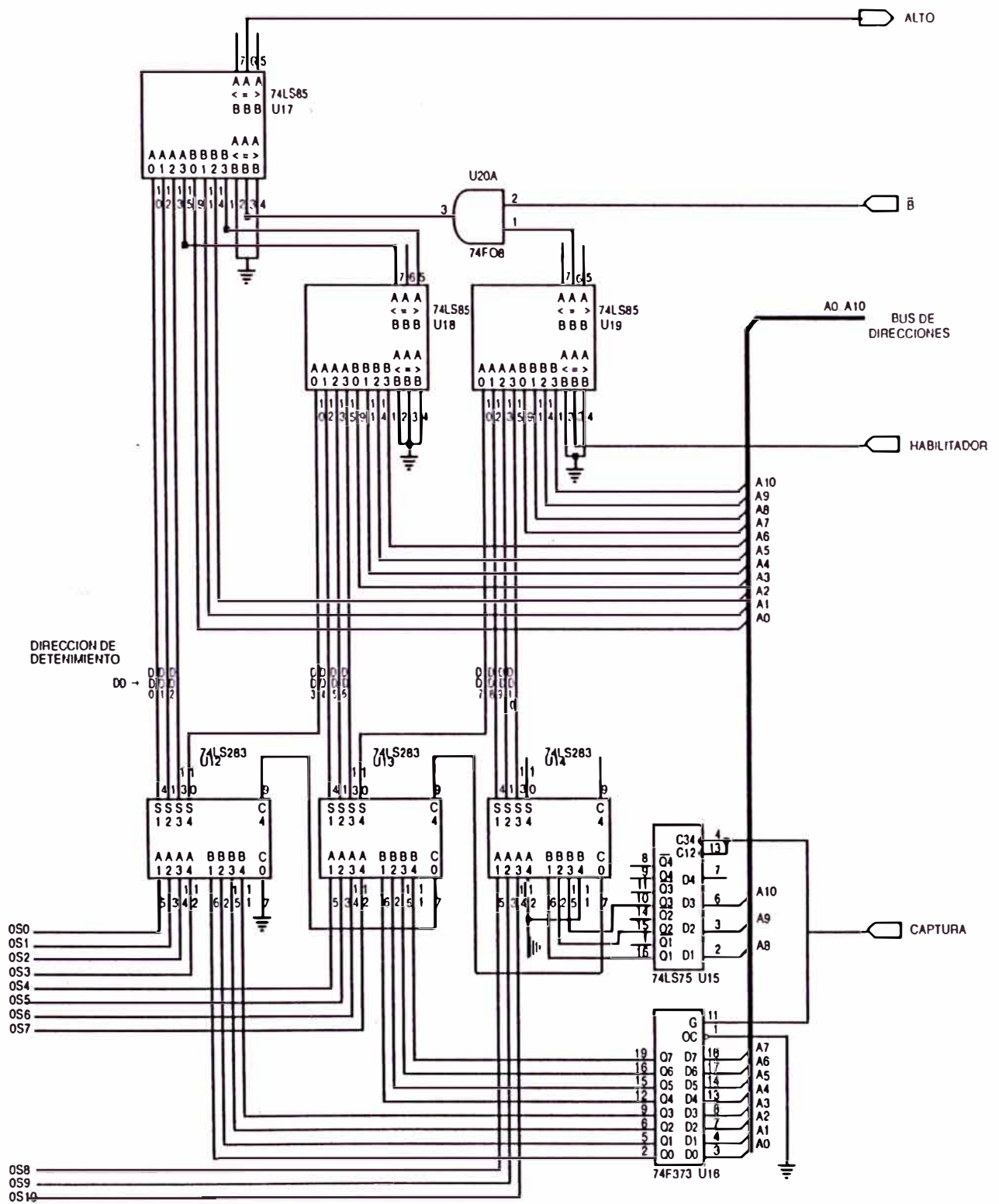


Figura 13. Diagrama del Circuito de Pre-Disparo

lectura. Al mismo tiempo, CE es colocado a “0” así las memorias son mantenidas activas para su lectura.

3.4.- El Circuito de Pre-Disparo

Este circuito se encarga de generar la característica de tener muestras previas al disparo, y es programable desde el computador. El diagrama de este circuito se muestra en la Figura 13. El circuito está formado por tres C.I 74LS283 (sumador paralelo binario de 4 bits) que hacen la etapa del SUMADOR, los cuales están conectados en cascada de forma que tengamos un sumador paralelo de 11 bits; la etapa de REGISTRO la forman dos latches, uno el 74F373 (U16 latch de 8 bits) y el otro el 74LS75 (U15 latch de 4 bits), estos son conectados juntos para formar un Latch de 11 bits; y por último para la etapa del COMPARADOR se usa 3 comparadores 74LS85 (comparador de 4 bits), que están conectados de forma que se pueda comparar los 11 bits de la Dirección de Detenimiento (DD), que salen del SUMADOR y los 11 bits del bus de direcciones (A0..A10). Allí podemos observar que la dirección tomada por los latches, son entregadas a los sumadores los cuales añaden el OFFSET (provenientes de los puertos de salida PS 0 y PS 1), para producir la Dirección de Detenimiento (DD).

Los sumadores realizan la adición de la información del OFFSET (líneas OS0..OS10 que vienen de los puertos PS 0 y PS 1, Figura 13) y la dirección que se captura en los Latch (U15 y U16). Las líneas de salida de estos sumadores, forman la dirección de detenimiento DD (DD0..DD10).

Revisemos la instalación de los comparadores. El C.I 74LS85 (U19) que compara la igualdad de los bits A7, A8, A9 y A10 con los bits DD7, DD8, DD9 y DD10 respectivamente (es decir es el comparador más significativo), espera la señal HABILITADOR para poder entregar su salida A=B en “1”. El comparador 74LS85 (U18) al que se conectan las señales

A3, A4, A5 y A6 para la comparación con las señales DD3, DD4, DD5 y DD6, se le ha instalado a "0" las señales de entrada A<B, A>B y A=B de forma que cuando las señales A3..A6 difieran con las de DD3..DD6 produzcan en las señales de salida A=B igual a "0" y, A<B y A>B diferentes entre sí dependiendo de que datos sea mayor que el otro. Pero cuando las señales A3..A6 se igualan con las de DD3..DD6, se produce en las señales de salida A<B, A>B y A=B los bits "1", "1" y "0" respectivamente.

Resumiendo:

Condición	Resultado
$(A3..A6) = (DD3..DD6)$	$(A>B) = (A<B) = "1"$
$(A3..A6) \neq (DD3..DD6)$	$(A>B) \neq (A<B)$

Esta característica se aprovecha para entregar a las señales A>B y A<B como bits de datos (y no de control) a que las compare el último 74LS85 (U17). Este último comparador (el menos significativo), busca la igualdad de los bits A0, A1, A2 y A>B con los bits DD0, DD1, DD2 y A<B; pero para que su señal de salida, es decir la señal ALTO entregue un "1" lógico, debe la señal de entrada A=B del comparador 74LS85 (U17), estar en "1" lógico. Del circuito observamos que este bit lo controla una compuerta 74F08 (U20A), la cual entregará el "1" si tanto la señal de salida A=B del comparador más significativo (U19) y la señal \overline{B} , estén ambas en "1". La señal \overline{B} ha sido incluida como señal de control para que la comparación se realice con datos del bus de direcciones en estado estable, ya que cabe la posibilidad de que en ciertos instantes la información que lleva el bus de direcciones (A0..A10) puede tener datos todavía en formación (debido a los retardos en el contador de direcciones). En otras palabras la señal \overline{B} , sirve de habilitador ordenando la comparación cuando el bus de direcciones ya tiene un dato estable.

Si la señal HABILITADOR está en "0" lógico, se produce un "0" a la salida del comparador más significativo (U19), el cual hace "0" a la salida de la compuerta 74F08 (U20A), la misma que al ingresar al comparador menos significativo (U17) hace que su

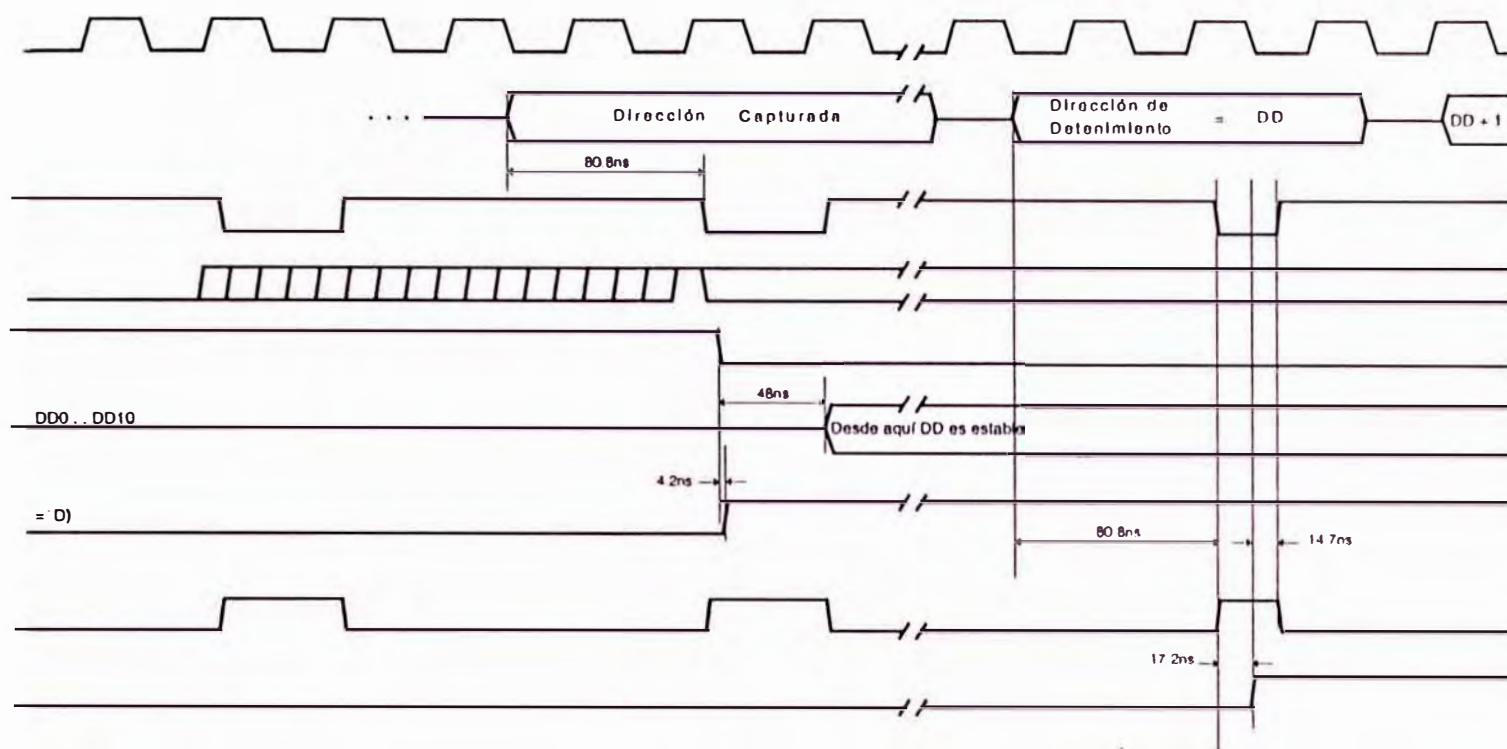


Figura 14. Diagrama de tiempos del Circuito de Pre-Disparo

salida (señal ALTO) sea también “0”. Es por eso que, para que pueda producirse la comparación de manera adecuada, es requisito que la señal HABILITADOR que proviene del circuito de control esté en “1”.

La Figura 14 muestra el diagrama de tiempos del funcionamiento cuando es habilitado el modo Con Pre-Disparo. Obviamente este gráfico está basado en el análisis de tiempos que se hizo cuando se estudió el circuito de control bajo el Modo Con Pre-Disparo (Figura 12).

Si TC cambia a estado alto, la señal CAPTURA cambia a nivel bajo, lo que hace que se capture en los latches (U15 y U16) la dirección actual, y después de 48ns de que la señal CAPTURA ha pasado a nivel bajo se produce la Dirección de Detenimiento (DD) válida para poder ser comparada.

Luego se tiene que esperar un tiempo equivalente a lo que se tarda el contador de direcciones en entregar una dirección idéntica a la Dirección de Detenimiento (DD). La cantidad de eventos (cuentas) que el contador de direcciones tiene que contar a partir de la dirección que se “capturó” es una cantidad igual a “OFFSET”. Esto se debe a que a la dirección “capturada” se le ha agregado justamente la cantidad OFFSET. Es decir:

$$\text{Dirección de Detenimiento} = \text{Dirección Capturada} + \text{OFFSET}$$

Cuando el contador de direcciones entrega la dirección DD, los contadores, habilitados por B, entregan un pulso positivo en la señal ALTO, quedando así todo listo para iniciar el proceso de lectura de memoria.

3.5.- El Circuito de Comparación

El circuito de comparación que se muestra en la Figura 15, se basa en dos comparadores de 8 bits 74F521 (U29 y U30), los cuales se han dispuesto en modo de expansión en paralelo para poder minimizar el retardo. Esta disposición necesita de una puerta NOR 74F02 (U28) cuya salida, etiquetada por TC, se pone en “1” cada vez que los 16 bits de las señales TA0..TA15, provenientes del circuito bajo prueba, son iguales a los 16 bits PD0..PD15, provenientes de los puertos de salida PS 2 y PS 3.

En la Figura 15 se observa que las entradas habilitadoras de los dos C.I 74F521 (entradas G) son usadas como entradas de habilitación para la comparación, y se han etiquetado como E0 y E1. Así, ellas definen el momento en que ordenan a los comparadores a que realicen su labor.

El diagrama de tiempos de la Figura 16a muestra cuando se coloca en el bus de TA el valor PD, es decir el mismo que fué programado previamente en los puertos, y luego los

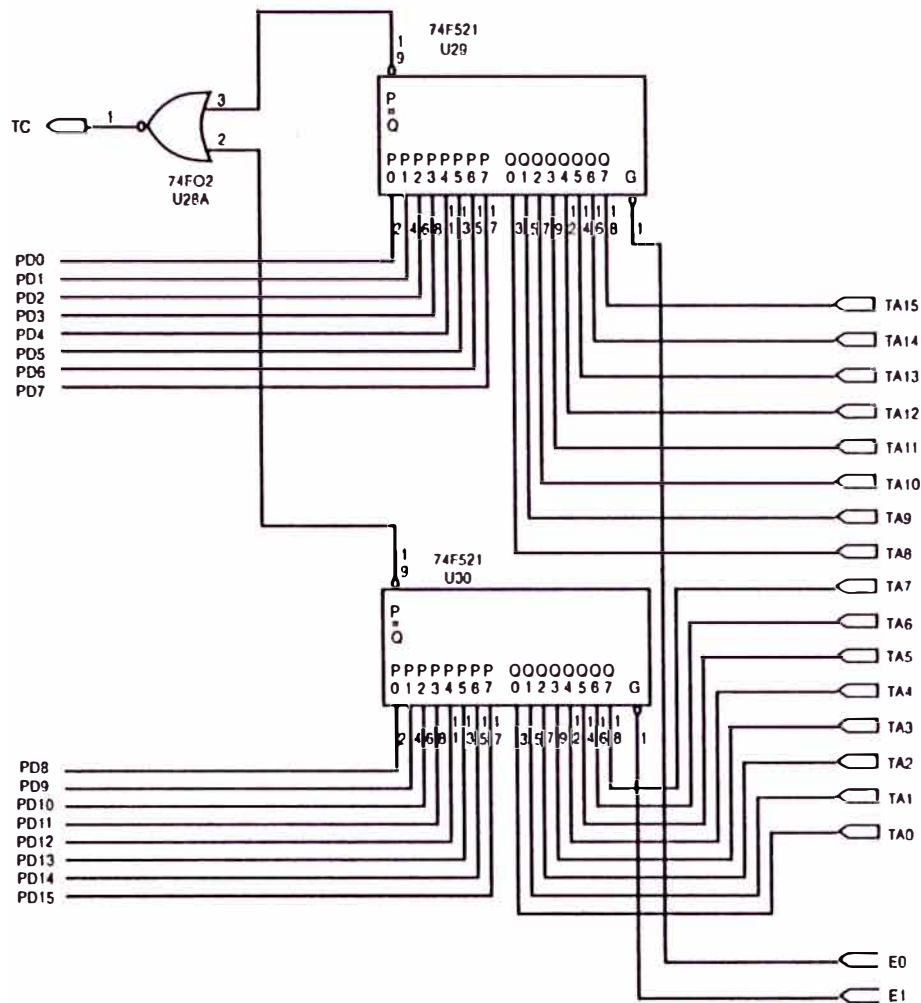


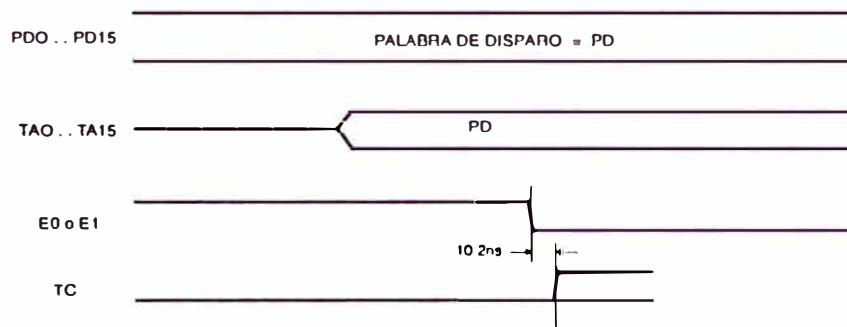
Figura 15. Diagrama del Circuito de Comparación.

dos habilitadores E0 y E1 son puestos en “0” . Esto significa la señal TC sale con un retardo de 10.2ns respecto a la última señal E0 o E1 que se fué a “0”.

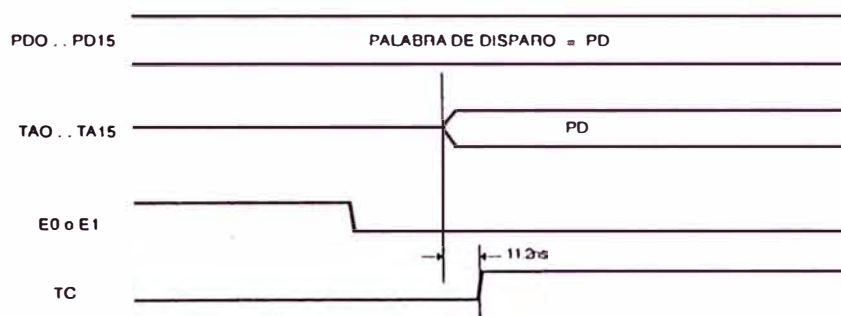
El diagrama de tiempos de la Figura 16b muestra el retardo de propagación de TC cuando los habilitadores E0 y E1 ambos son puestos a “0” antes de que aparezca en TA0..TA15 el valor PD el cual fué programado previamente en los puertos PS 2 y PS 3. Aquí la señal TC sale con un retardo de 11.2ns.

Cualquiera de los dos gráficos son válidos para producir el disparo, no obstante se

recomienda usar los habilitadores como en el primer gráfico, porque asegura estabilidad en el bus de prueba TA0..TA15.



(a) Bus TA válido antes que E0 o E1



(b) E0 o E1 activos antes que el Bus TA

Figura 16. Diagramas de tiempos del Circuito Comparador

3.6.- El Circuito del Banco de Memoria

El diagrama de este circuito se muestra en la Figura 17, en el cual, cada módulo se compone de 2 registros de desplazamiento de 4 bits 74LS95, un registro buffer de carga paralela sincrónico 74LS374 y una memoria RAM de 2Kx8 TMM2016 (6116). El circuito consta de 4 módulos, los que nos da la posibilidad de contar con 8 canales, pero podría aumentarse para así contar con más canales.

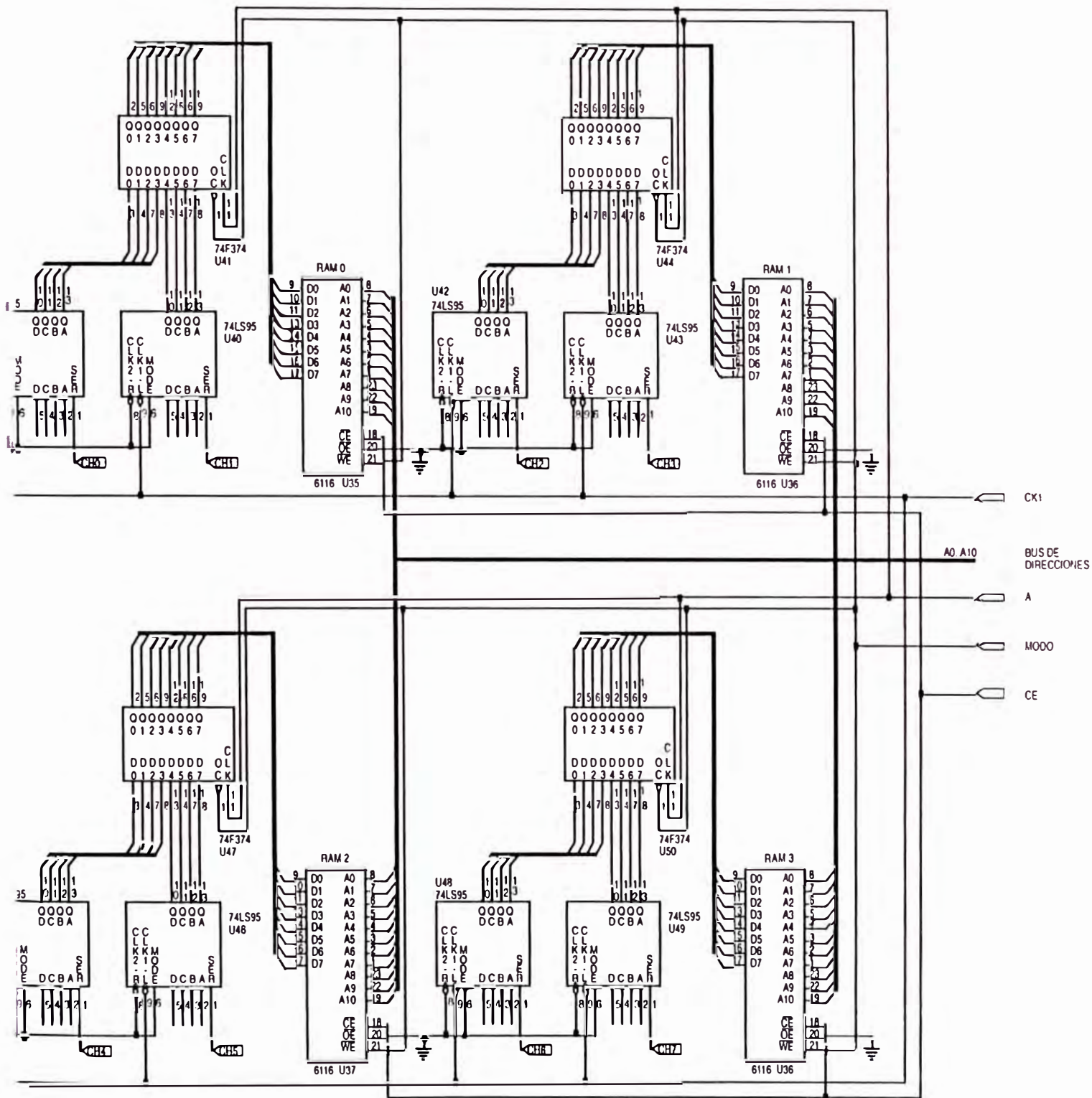


Figura 17. Diagrama del Circuito del Banco de Memoria

Como podemos apreciar en la Figura 17 todas las memorias comparten el mismo bus de direcciones, las cuales provienen del contador de direcciones, y comparten también las mismas señales de control, es decir, el pin WE, el cual es controlado por el bit MODO de la unidad de control. Las entradas CE (Chip Enable) de las memorias son unidas a la señal CE del circuito de control. El pin OE de las memorias, es decir OE, es colocado a nivel activo ("0" lógico).

La señal MODO también controla la habilitación de los registros 74F374. Los datos a muestrear ingresan por las ocho sondas, etiquetadas como CH0, CH1, CH2, CH3, CH4, CH5, CH6 y CH7, hacia los registros de desplazamiento 74LS95, todos ellos están trabajando en sincronismo con la señal de reloj CK1 de 20Mhz y toman el dato en el pulso bajo. Debido a que los registros de desplazamiento 74LS95 son circuitos TTL, la interfaz solamente podrá medir señales digitales de esa lógica. Además no se podrá medir señales de tres estados; si éstas estuvieran presentes en las sondas se tomarán datos aleatorios.

Analicemos las secuencias de escritura y de lectura hacia y desde memoria, con sus respectivos diagramas de tiempos.

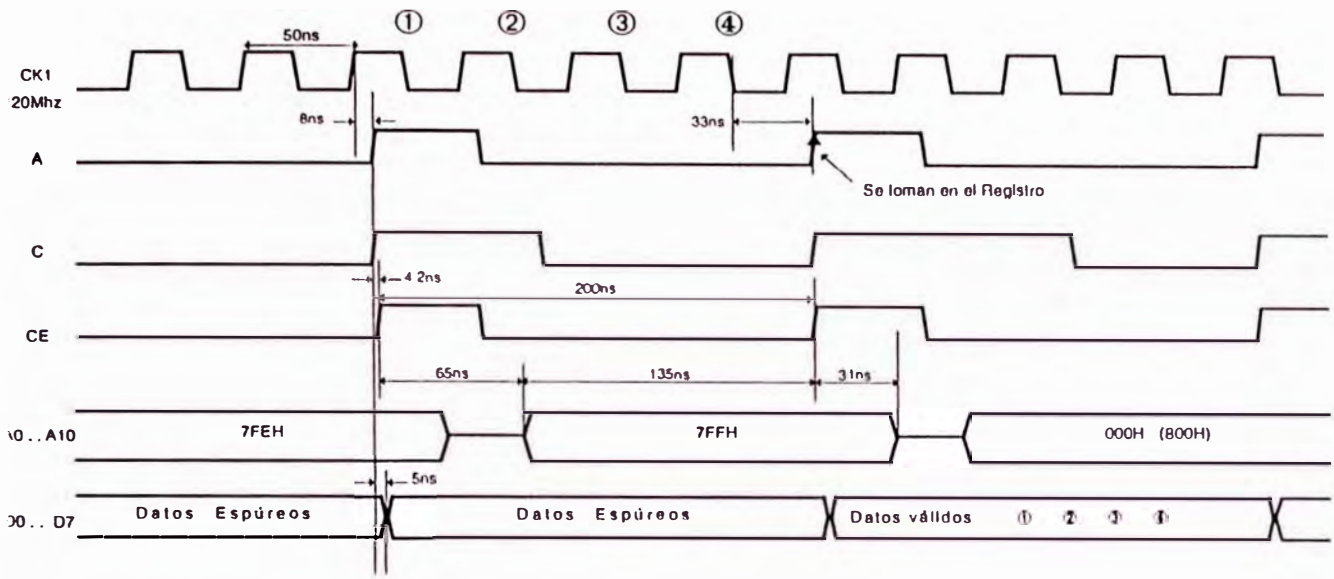
3.6.1.- Escritura de datos en las memorias

Ya que la onda del reloj tiene una frecuencia de 20Mhz (50% de ciclo de servicio) es decir un periodo de 50ns, entonces se cuenta con 50ns para poder grabar el bit muestreado dentro del registro de desplazamiento 74LS95. Este dispositivo según sus especificaciones técnicas tiene un tiempo de retardo de propagación de 25ns, así que con los 50ns que le proporcionamos tiene el tiempo suficiente para tomar el bit correspondiente. Como el C.I. 74F374, que es el que toma los 4 bits muestreados para entregárselos a la memoria, está recibiendo los pulsos positivos de la señal A, observamos en la Figura 18

que está distante del cuarto pulso negativo de CK1 33ns, lo cual es mayor que los 25ns que el 74LS95 necesita, entonces existe completa garantía que los datos a la frecuencia de 20Mhz estén ingresando correctamente a los registros 74F374, para luego poder ser almacenados en memoria.

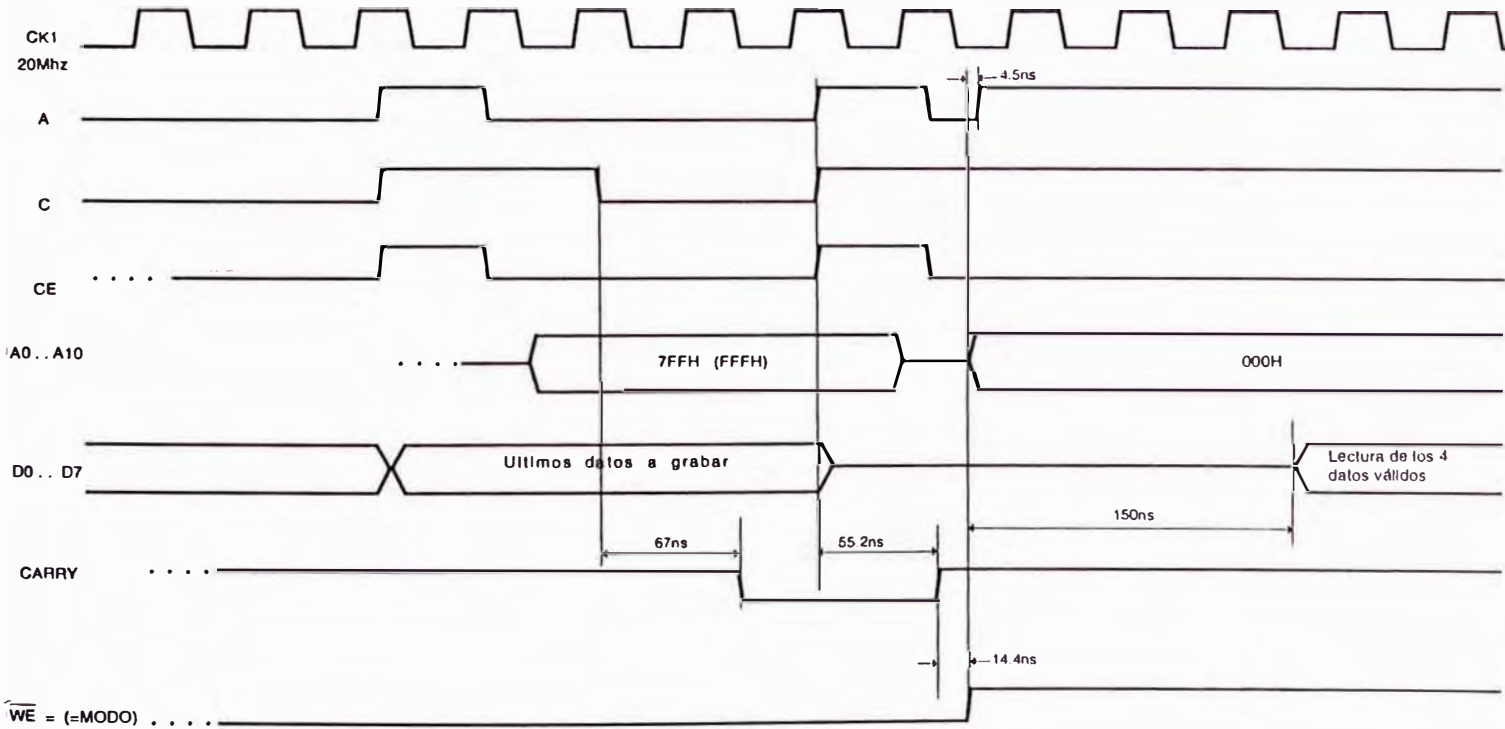
El comportamiento eléctrico de los modos Manual y Automático en lo que respecta a la escritura de información, son idénticos. Por esta razón los vamos a estudiar en un mismo gráfico, que es el que se muestra en la Figura 18. La Figura 18a muestra el inicio del ciclo de escritura, en ella observamos que efectivamente tenemos, en el peor de los casos, 135ns para poder grabar el dato; lo que es suficiente ya que la memoria necesita sólo 90ns desde que su CE esté en "0" para poder grabar la información. Observamos también que el bus de direcciones de las memorias empiezan con la dirección precargada 7FEh, mientras que las muestras tomadas en los pulsos de CK1 1, 2, 3 y 4 ingresan al 74LS95; éstos son tomados en el 74F374 en el segundo pulso positivo de la señal A y este registro luego de 5ns se los entrega a la memoria y ésta los graba en la dirección 000H (800H para el contador de direcciones). En la dirección 7FFH se ha grabado Datos Espúreos que provienen de lo que el registro 74F374 tomó en el primer pulso de la señal A.

En la Figura 18b se muestra cuando termina el ciclo de escritura, la cual se origina cuando el contador llega a la cuenta FFFh (que para el bus de direcciones A0..A10 significa 7FFh, es decir la última dirección de memoria). Entonces se genera un pulso en CARRY, y esto origina que MODO cambie a "1", el contador se estabiliza en la cuenta 000h. A partir de ese instante se ingresa a modo de lectura de las memorias, las cuales después de 150ns de tiempo de acceso se produce un dato estable y éstos son los cuatro primeros datos grabados en la dirección 000h por canal, es decir las memorias quedan listas para que las lean.



CARRY = '1'
WE = '0' (=MODO)

(a) Inicio del ciclo de escritura



(b) Fin del ciclo de escritura

Figura 18. Diagrama de tiempos del ciclo de escritura a las memorias RAM para los Modos Manual y Automático

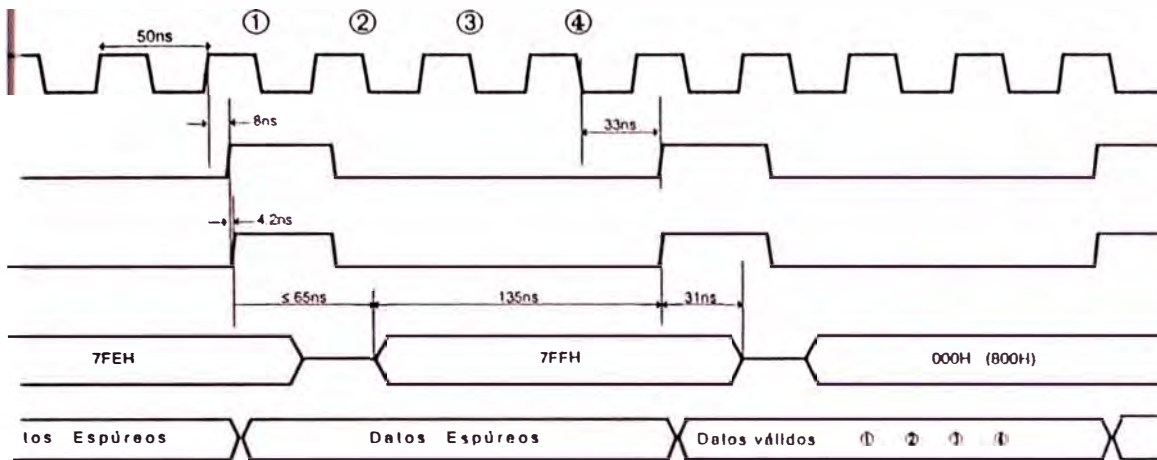
Observemos que inicialmente en la dirección de memoria 7FEh se grabaron datos espúreos, pero al finalizar el ciclo de escritura en dicha dirección se graban los últimos datos válidos, en resumen cada una de las direcciones de memoria tienen un dato válido.

En la Figura 19, se grafica la dinámica de tiempos para la escritura a memoria bajo el Modo Con Pre-Disparo. En la Figura 19a se muestra el arranque del modo, el cual es de la misma forma que se analizó en la Figura 18a. En la Figura 19b se muestra dos eventos, primero cuando ocurre el disparo de TC generada por el circuito comparador y segundo cuando se genera la señal de ALTO que indica el fin de la escritura en las memorias.

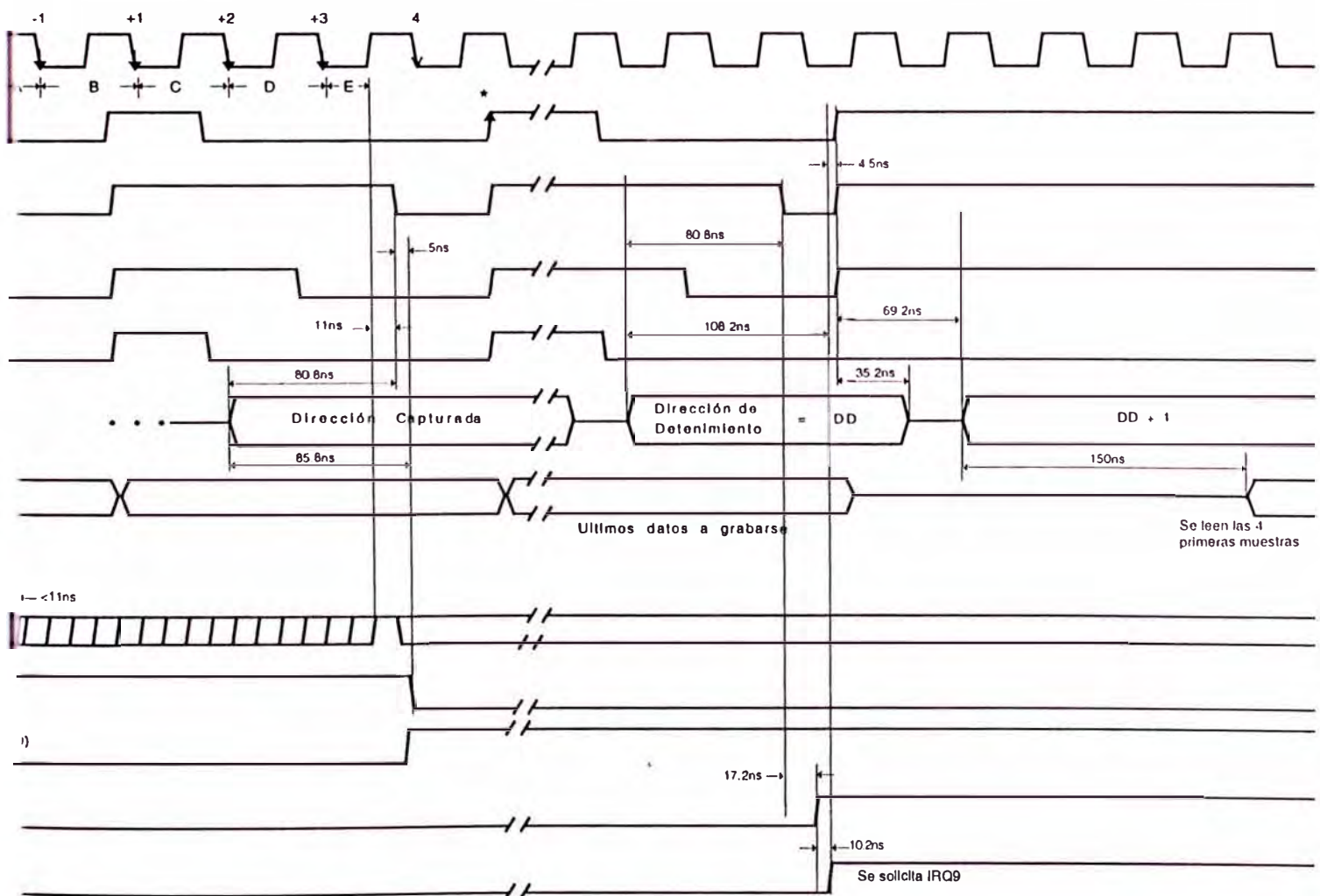
Así observamos que cuando se produce el pulso en TC, la señal CAPTURA hace que se grabe la dirección que entrega el contador de direcciones, (que en el gráfico aparece como: "Dirección Capturada"). Luego de esto, se espera un tiempo hasta que aparezca en el bus de direcciones la Dirección de Detenimiento (DD); cuando esto ocurre, el pulso de bajada de la señal B entrega la habilitación para que tanto la señal ALTO como MODO cambien a "1", finalizando la secuencia de escritura de información en las memorias.

Debido a que la señal C genera un pulso adicional, observamos en la Figura 19b, que la dirección en la que se detiene el contador de direcciones es $DD + 1$, la cual se convierte en la primera dirección que será leída, en tanto que la última en ser leída es la dirección DD.

El hecho que la dirección $DD+1$ sea la primera dirección en ser leída es conveniente ya que, como puede observarse, cada dirección de memoria contiene datos que fueron tomados cuatro pulsos antes de su grabación, esto significa que la



(a) Inicio del ciclo de escritura



(b) Disparo y fin del ciclo de escritura

Figura 19. Diagrama de tiempos del ciclo de escritura a las memorias RAM para el Modo Con Pre-Disparo.

“Dirección Capturada” no contiene datos que estaban sucediendo cuando ocurría el disparo, sino más bien la que los contiene es la “Dirección Capturada + 1”. Esto significa que la dirección DD + 1 debe ser la primera dirección en leerse y no DD.

Para aclarar lo anteriormente expresado veamos un ejemplo. Supongamos que se necesite 200 bits de muestras previas al disparo, entonces el computador calcula:

$$\text{OFFSET} = 2048 - (200)/4 = 800h - (C8h)/4$$

$$\text{OFFSET} = 1998 = 7CEh \text{ el cual es enviado a los puertos PS 0 y PS 1.}$$

Supongamos ahora que la **Dirección Capturada** en el latch sea 12Ch. Por lo tanto la **Dirección de Detenimiento (DD)**, entregada por el circuito sumador es:

$$\text{DD} = 12Ch + 7CEh (= 8FAh)$$

En realidad la suma es 8FAh pero como el sumador sólo considera 11 bits y no 12, entonces la Dirección de Detenimiento queda así:

$$\text{DD} = 0FAh$$

Ahora, cuando la Dirección Capturada (12Ch) es tomada por el latch, en el circuito de pre-disparo están sucediendo en el tiempo los 4 pulsos siguientes (en la Figura 19b son nombrados como +1, +2, +3 y 4) que son respecto a estos los que debemos considerar el pre-disparo. Estos bits obviamente son escritos en la dirección siguiente, es decir en la 12Dh. El circuito, según la Figura 19b, aún graba los últimos bits de información en la dirección DD, para el ejemplo sería 12Ch, y empieza a leer desde 12Dh (DD+1). Por consiguiente, los datos en la memoria quedan divididos en dos grupos:

Direcciones de Pre-disparo: Desde la dirección 0FBh (DD + 1) hasta 12Ch
(Dirección Capturada).

Direcciones de Post disparo: Desde la dirección 12Dh (Dirección Capturada + 1) hasta 0FAh (DD).

Es decir tenemos: $12Ch - 0FBh + 1 = 32h = 50$ direcciones de pre-disparo, es decir a $50 * 4 = 200$ bits de pre-disparo. En realidad son 200 bits aproximadamente, porque existe cierta incertidumbre sobre la precisión del número de muestras previas solicitadas desde el software. Para entender esto en la Figura 19b hemos denotado cuatro pulsos de reloj como -1, +1, +2 y +3 con el objeto de poder hablar de ellos en forma específica.

Ya sabemos que la señal TC debe subir entre los flancos negativos de la señal B, exactamente 11ns antes de los flancos negativos de B. Esto significa que TC, que es quien dispara, podría subir en cualquiera de los intervalos siguientes (ver Figura 19b):

- Intervalo A: Entre 11ns antes de la señal B y el pulso -1.
- Intervalo B: Entre los pulsos -1 y +1.
- Intervalo C: Entre los pulsos +1 y +2.
- Intervalo D: Entre los pulsos +2 y +3.
- Intervalo E: Entre el pulso +3 y 11ns antes de la señal B.

Ahora analicemos lo que ocurre si sube TC, en cualquiera de los intervalos, no sin antes recordar que los datos: +1, +2, +3 y 4, sea cual fuera el intervalo, son almacenados por el registro 74F374 en el flanco positivo de la señal A, que en la Figura 19 se ha señalado con un *; y son almacenados en la memoria en la localización Dirección Capturada + 1, que representa la primera dirección de post disparo.

- Caso A:** Si TC sube en el Intervalo A, entonces la primera dirección de post-disparo debería contener a la muestra -1, sin embargo no lo contiene porque esta muestra pertenece a la dirección anterior (Dirección Capturada). Aparentemente se pierde un bit de post-disparo, pero en realidad no lo pierde, porque ese bit se puede apreciar en los datos de pre-disparo. Esto porque erróneamente se lo estamos asignando a los datos de pre-disparo. En conclusión, los datos de pre-disparo son los que pierden un bit.
- Caso B:** Si TC sube en el Intervalo B, entonces la primera dirección de post-disparo contiene justamente cuatro muestras después de darse el disparo. Es el caso que nos da la exactitud en cuanto a las muestras múltiplos de 4 de pre-disparo que fueron ingresadas mediante el software.
- Caso C:** Si TC sube en el Intervalo C, entonces la primera dirección de post-disparo, esta incluyendo un muestra de pre-disparo. En este caso se gana un bit para los datos de pre-disparo y se pierde uno para el post-disparo.
- Caso D:** Si TC sube en el Intervalo D, entonces la primera dirección de post-disparo, esta incluyendo dos muestras de pre-disparo. En este caso se gana dos bits para los datos de pre-disparo y se pierde dos para el post-disparo.
- Caso E:** Este, así como el Caso A, son los menos probables debido a que estos intervalos son los más angostos. Si TC sube en el Intervalo E, entonces la primera dirección de post-disparo está incluyendo tres muestras de

pre-disparo. En este caso los datos de pre-disparo ganan tres bits y se pierde en igual número para los datos de post-disparo.

Para el operador todos los casos son considerados como el Caso B, pero debe tener en cuenta las imprecisiones que existen en cuanto al número de muestras de pre-disparo. Así podemos concluir que el número de muestras puede tener un error absoluto de: -1, 0, +1, +2 ó +3. Además debemos señalar que esta imprecisión no es crítica ya que como se dijo se puede solicitar desde 4 a 8188 muestras de antelación al disparo.

3.6.2.- Lectura de datos de las memorias

Las memorias son leídas en igual forma que para cualquier modo de disparo. Nosotros podemos observar que tanto en la Figura 18 como en la 19 que cuando se solicita IRQ₉, el bit de MODO cambia a “1”, y el circuito de control realiza lo siguiente:

CE es colocado en “0”, es decir todas las memorias están activas.

MODO es colocado a “1”, lo que coloca a las memorias en modo de lectura.

MODO es colocado a “0” lo que presetea a las señales A, B y en especial a C la que deja habilitada a la compuerta AND 74F08 de manera que el reloj ya no depende de C sino de GCK.

Así como ya expresamos que cada vez que se produce una lectura de un puerto se produce un pulso en haciendo cambiar automáticamente una dirección. Los diagramas de tiempos para la lectura de datos desde memoria son los mismos contemplados en la sección 3.1 para la lectura de un puerto (ver Figura 6).

La primera dirección de inicio de la lectura tanto para los modos de Manual y Automático es 000h y termina en FFFh; en tanto que para el Modo Con Pre-Disparo inicia las lecturas desde la dirección señalada por la Dirección de Detenimiento más uno (DD+1) y termina en la dirección de Detenimiento (DD).

3.7.- Características Eléctricas de los Circuitos de Entrada

Para finalizar este capítulo entregaremos los datos técnicos de los circuitos de entrada, tanto para los canales de datos (CH0..CH7), como para la palabra de disparo (TA0..TA15).

Cada canal (CH0-CH7) tiene las siguientes características:

Nivel de Entrada Alto mínimo	:	$V_{IH\min} = 2.0 \text{ V}$
Nivel de Entrada Bajo máximo	:	$V_{IL\max} = 0.8 \text{ V}$
Corriente de Entrada Alto máximo	:	$I_{IH\max} = 0.1 \text{ mA}$
Corriente de Entrada Alto mínimo	:	$I_{IH\min} = 20 \text{ A}$
Corriente de Entrada Bajo máximo	:	$I_{IL\max} = -0.4 \text{ mA}$

Las características eléctricas de las entradas TA0-TA15 son las siguientes:

Nivel de Entrada Alto mínimo	:	$V_{IH\min} = 2.0 \text{ V}$
Nivel de Entrada Bajo máximo	:	$V_{IL\max} = 0.8 \text{ V}$
Corriente de Entrada Alto máximo	:	$I_{IH\max} = 0.1 \text{ mA}$
Corriente de Entrada Alto mínimo	:	$I_{IH\min} = 20 \text{ A}$
Corriente de Entrada Bajo máximo	:	$I_{IL\max} = -0.6 \text{ mA}$

En este capítulo se ha detallado cada parte de los circuitos que constituyen el hardware del interfaz, así como el estudio de ellos en pleno funcionamiento de acuerdo a los modos de disparo, los cuales se solicitan desde el software que se ejecuta en el computador. El siguiente capítulo muestra el software que controla al interfaz.

CAPITULO IV SOFTWARE DEL SISTEMA

En este capítulo, se examinarán las características del software que se ejecuta en el computador personal. El programa en su totalidad ha sido desarrollado en lenguaje Turbo Pascal versión 6.0 de Borland International, Inc. Específicamente se ha usado Turbo Vision, que es la interfaz que ha usado Borland para la confección de su Entorno Integrado de Desarrollo (EID) de Turbo Pascal versión 6. De manera que con esta herramienta de programación, se puede desarrollar software con el aspecto y funcionamiento que el EID.

Turbo Vision está dotado de un conjunto de herramientas extremadamente potentes, y no es más que una colección de software reutilizable, el cual está basado casi por completo en la Programación Orientada a Objetos (POO), una forma de programar en donde las características (datos) y el comportamiento (procedimientos) se ubican en una única entidad llamada *objeto*, los cuales gozan de ciertas características tales como herencia, encapsulación, polimorfismo y hasta se puede hacer asignación dinámica de objetos, lo cual involucra a que existan variables que son punteros a objetos y/o que objetos contengan punteros.

El software que soporta al sistema se compone de 2 archivos compilados : ALOG.EXE y ALOG.HLP. El primero es el archivo ejecutable y el segundo el archivo que muestra la ayuda del manejo del software y es invocado desde ALOG.EXE.

El ALOG.EXE es un programa cuyo ambiente de trabajo esta basado en un sistema de menús y ventanas de diálogo que hacen que la comunicación con el usuario sea realmente cómoda. Todo está basado en ventanas las cuales se manejan con bastante flexibilidad y está potenciado con rutinas de utilidad como grabar datos a disco para que puedan ser

recuperados en el momento que se requiera para su estudio; a la vez que se podrá imprimirlos para un mejor análisis. Adicionalmente, como se mencionó se podrá usar una ayuda para cada item del ambiente, lo que hace que no requiera un manual de referencia para su manejo.

Se ha programado integralmente en base al software reutilizable que brinda Turbo Vision, en donde los objetos tales como la barra de menú, la línea de estado, ventanas de diálogo, etc. son simplemente usados y modificados según las necesidades. Así la labor de programación ha sido la de orientar a los objetos de Turbo Vision el desarrollo de nuestro aplicativo, y desarrollar los objetos propios del presente proyecto. Turbo Vision es extremadamente flexible, pero éste tiene el efecto secundario de ser bastante complicado.

4.1.- Estructura del Software del Sistema

El programa ALOG podemos estructurarlo en tres partes funcionales e interrelacionadas.

Las cuales son:

El Programa de Gestión Principal.

El Programa de Gestión de Menús.

El Programa de Gestión de Ordenes.

A continuación detallamos los diagramas de flujo de cada una de estas partes funcionales.

4.1.1.- El Programa de Gestión Principal

Es la rutina principal, la cual se ejecuta en primer lugar, y soporta el reconocimiento del hardware, así como el examen de las teclas que son válidas en determinado

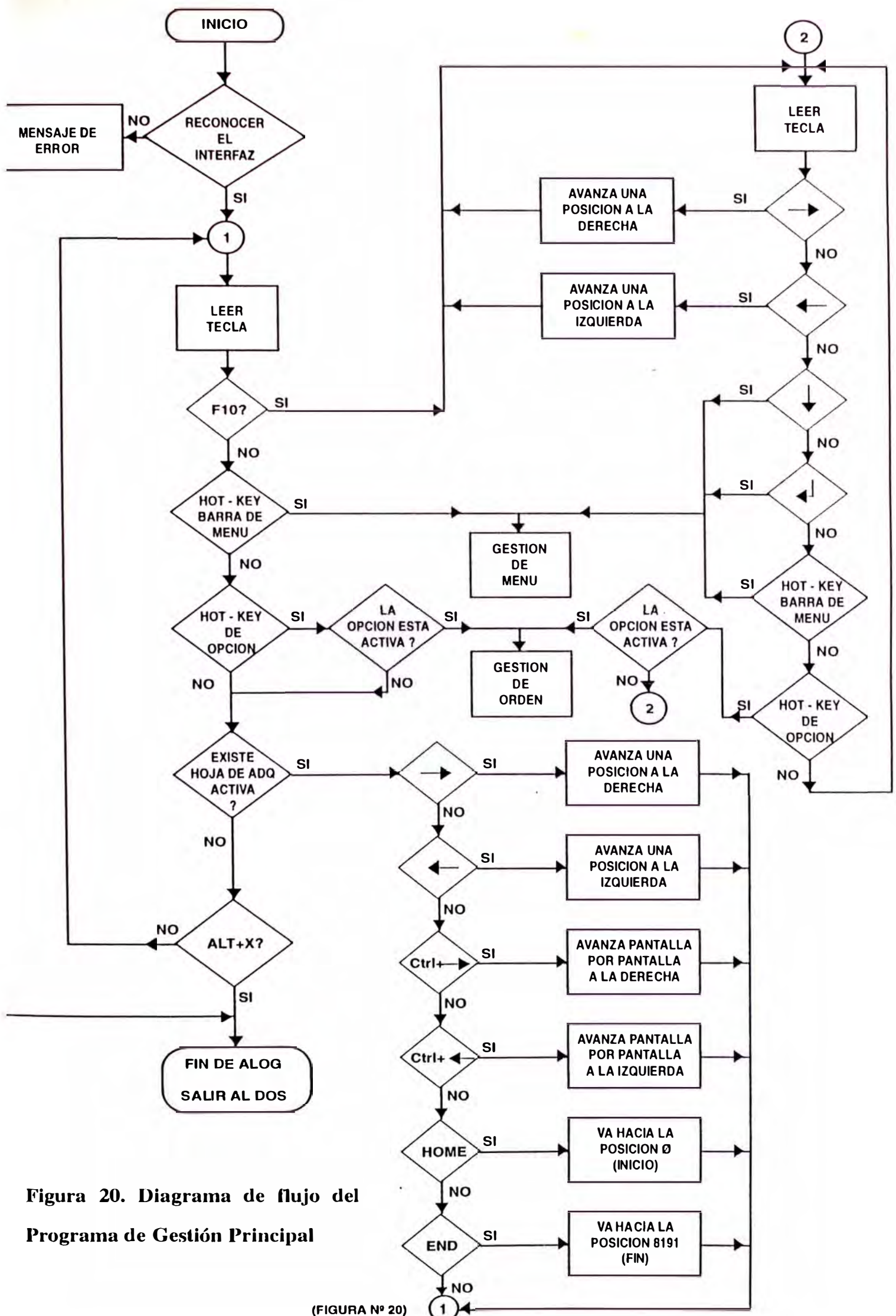


Figura 20. Diagrama de flujo del Programa de Gestión Principal

instante en que se encuentre el programa. Así, luego de reconocer las teclas válidas, se ingresa a las rutinas correspondientes, como son la gestión de Menús o la de Ordenes, para su procesamiento.

La Figura 20 muestra el diagrama de flujo de la Gestión Principal, en el cual se muestra que la primera tarea es de reconocer el interfaz, si esto fracasa el programa finaliza regresando al sistema operativo. Si se logra la comunicación con la interfaz, entonces el programa de gestión Principal toma control, reconociendo los comandos que ingrese el operador.

4.1.2. El Programa de Gestión de Menús

Esta parte se encarga de examinar las teclas que son válidas cuando se ha ingresado a la barra de menús para poder seleccionar un ítem y posteriormente pedir su ejecución si es que el ítem respectivo está habilitado. La Figura 21 muestra el diagrama de flujo del programa de Gestión de Menús, en ella se observa que para que se de paso a la ejecución de una orden solicitada por el operador, es decir para entregarle la posta al programa de gestión de orden, primero deberá preguntarse si la opción esta activa, si no es así se regresa al inicio de la gestión de menús.

4.1.3.- El Programa de Gestión de Ordenes

Esta parte está destinada a la ejecución de cierta orden contemplada en cualquiera de los menús. Luego de la ejecución de la orden, debe retornar al programa principal, exactamente a la posición ① (Figura 20). La Figura 22 muestra el diagrama de flujo del Programa de Gestión de Ordenes, en la cual se muestra todas las opciones contempladas en el programa ALOG.EXE. Todas las opciones regresan al programa de gestión principal, con la única excepción de la opción **Terminar la Sesión** la que hace que se retorne al sistema operativo.

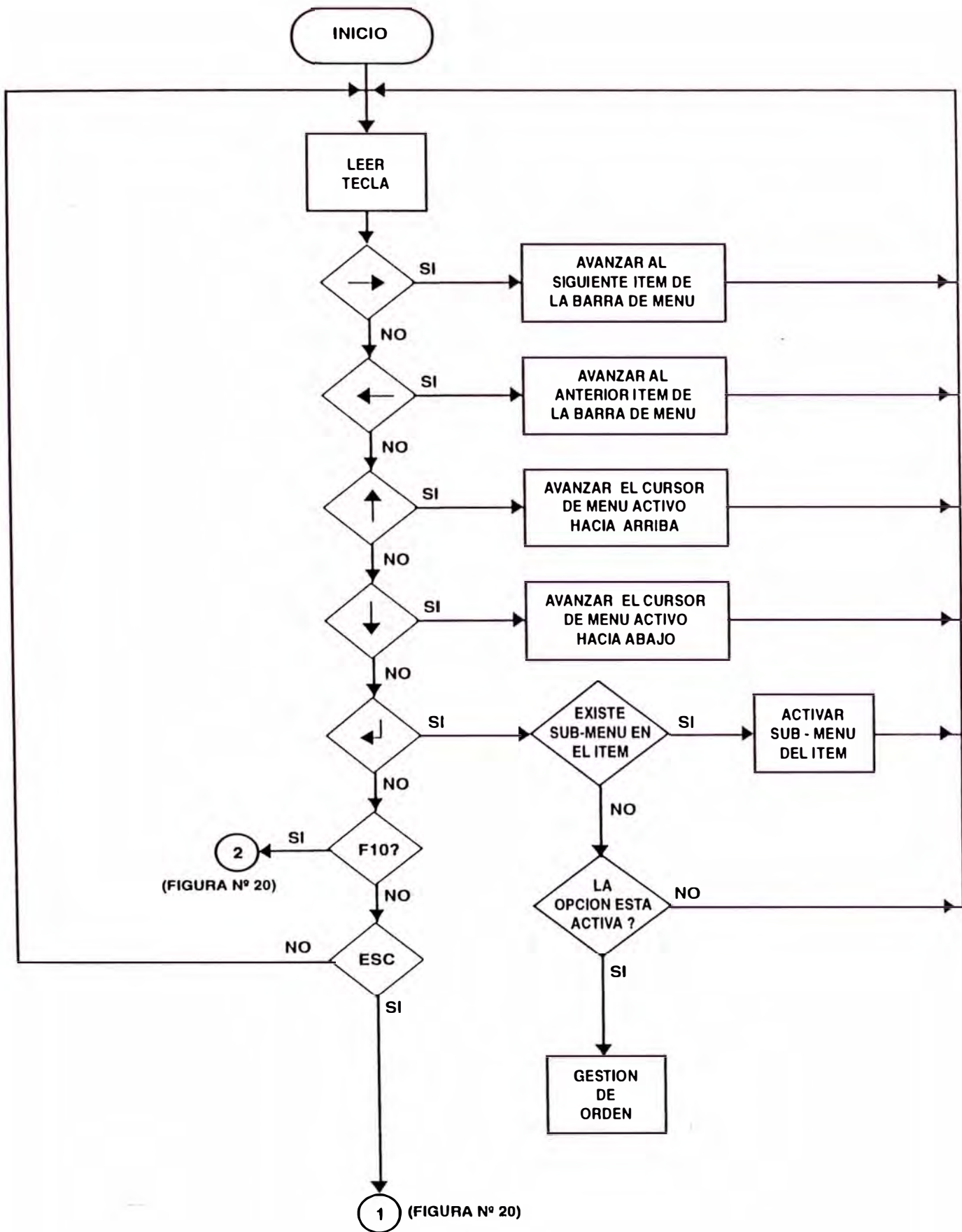


Figura 21. Diagrama de flujo del Programa de Gestión de Menús

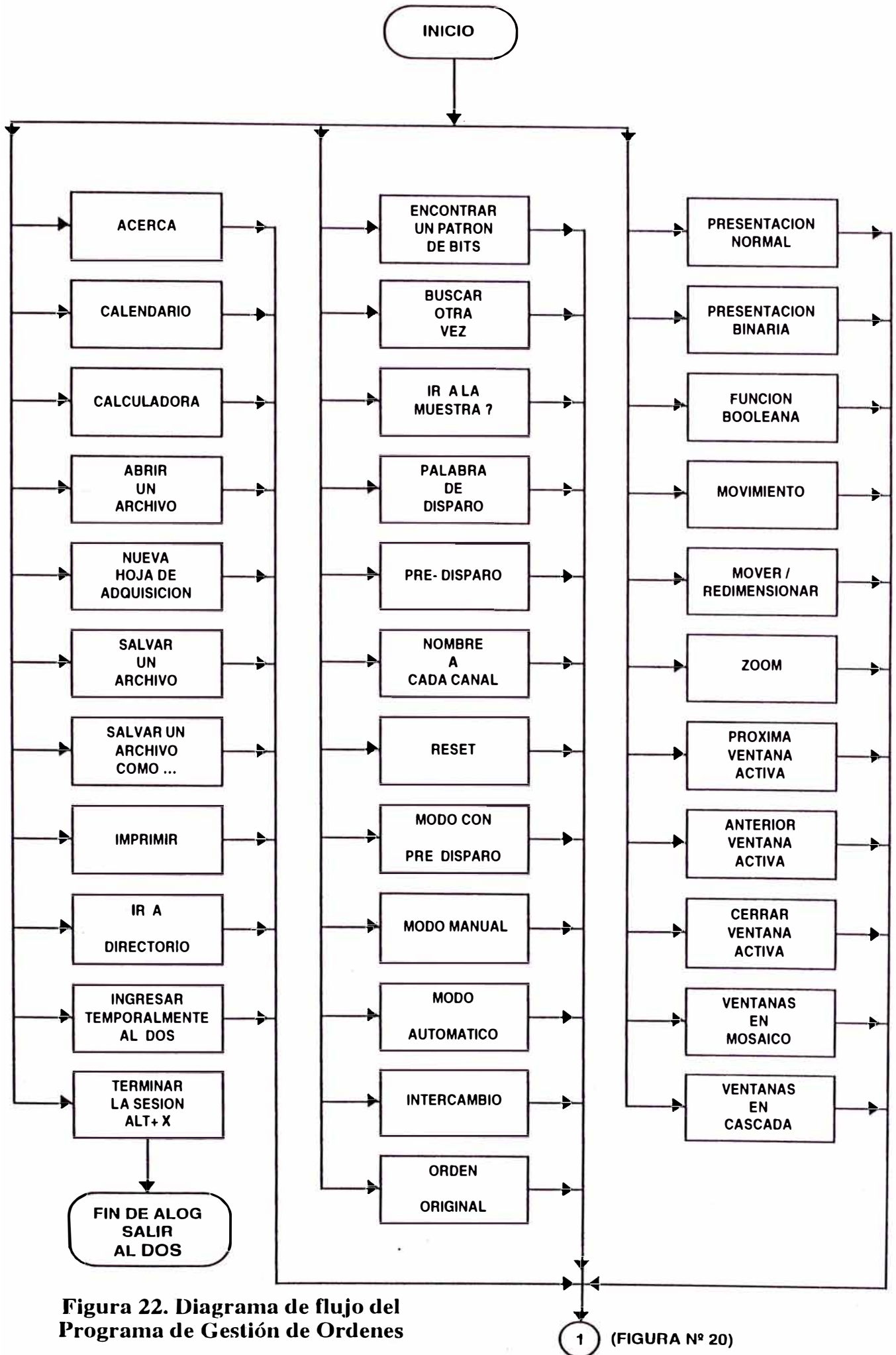


Figura 22. Diagrama de flujo del Programa de Gestión de Ordenes

4.2.- Los Menús

El software está basado en el uso de un sistema de menús, lo cual ofrece una facilidad en su empleo; así el operador puede solicitar y manejar cualquier evento ingresando al sistema de menús. Para solicitar una opción de menú se puede utilizar el teclado o el mouse; el ambiente y la manera de trabajar con ALOG resultarán familiares a los usuarios de Turbo Pascal. A continuación se describe el sistema de menús.

4.2.1.- El menú principal

El programa ALOG usa un sistema de menús desplegables que soportan un acceso por abreviaturas de teclado (Short-Keys) a las operaciones más frecuentemente usadas. Las opciones de menú principal son **Archivo, Buscar, Asignar, Ejecución, Resultados y Ventana**. El menú principal está representado por la barra de menú y por la línea de estado.

Cuando se ingresa satisfactoriamente al programa, es decir se reconoce el hardware instalado (el Interfaz), se presenta un mensaje de bienvenida y luego se muestra un ambiente donde se visualiza el menú principal, tal como se muestra en la Figura 23.

Existen dos maneras de seleccionar una opción del menú principal las cuales son mediante el teclado o el mouse. La tecla F10 activa el menú principal, resaltando la opción actualmente seleccionada. Se puede mover la barra de resalte utilizando las teclas **FLECHA DERECHA** y **FLECHA IZQUIERDA** del teclado y seleccionar después la opción pulsando **ENTER**. Un método más directo es pulsar simplemente F10 y la primera letra de la opción deseada (por ejemplo, **A** para **Archivo**). Los usuarios de mouse sólo necesitan pulsar sobre la opción deseada.

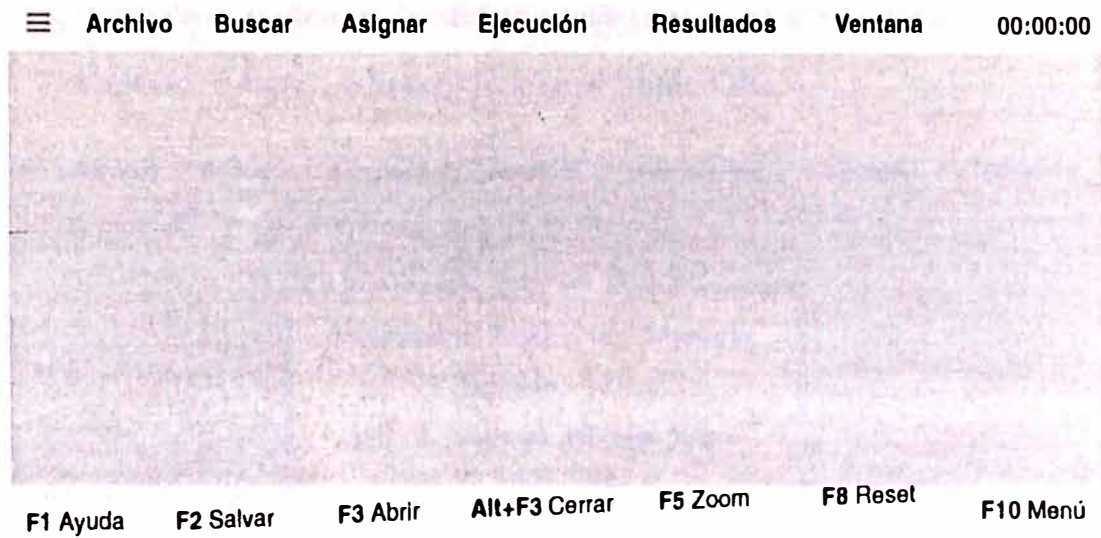


Figura 23. El Menú Principal

En la parte inferior de la pantalla se muestra la línea de estado que es una lista de las abreviaturas (short-keys) de teclado usadas con más frecuencia.

4.2.2.- El menú ≡ (Alt+barra espaciadora)

El menú del sistema aparece en la parte izquierda de la barra de menú y es representado por el símbolo ≡. Cuando se ingresa al menú ≡, se observa una ventana como se muestra en la Figura 24.

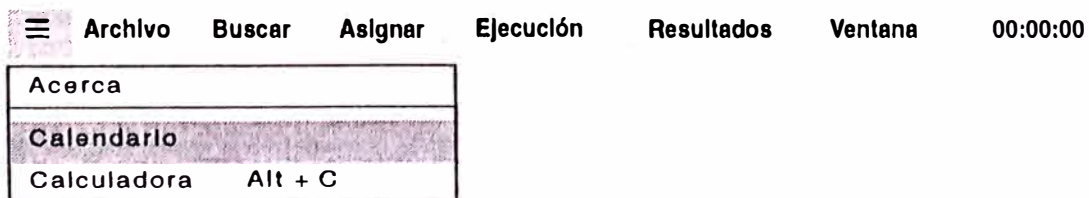


Figura 24. El Menú ≡

4.2.2.1.- La opción Acerca

Cuando se escoge el comando Acerca desde el menú (Sistema), aparece una ventana de diálogo como la que se muestra en la Figura 25, donde se

muestra la presentación del software. Para cerrar la ventana, presionar Esc, Espacio, o Enter; o hacer click en el botón **OK**.

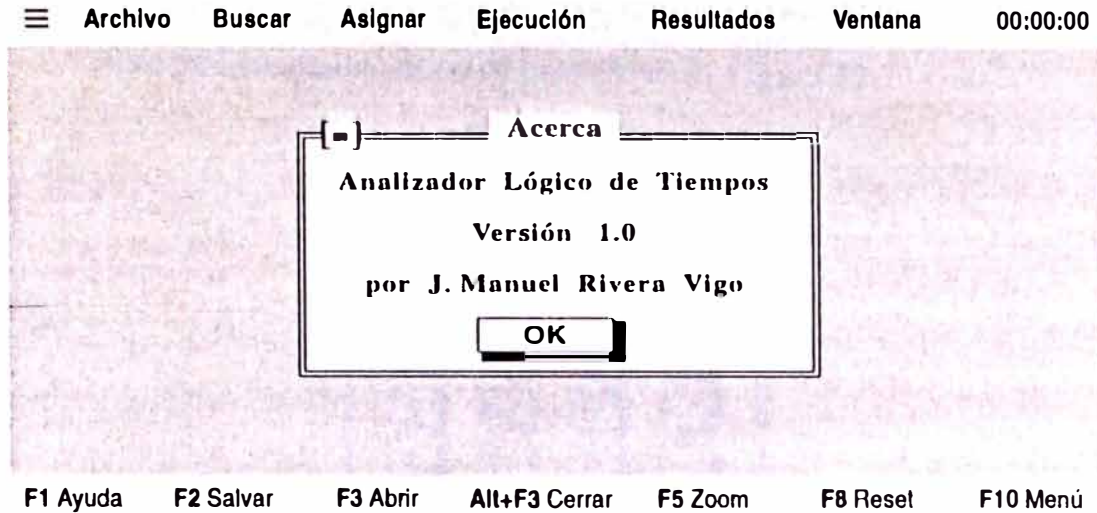


Figura 25. Ventana de Diálogo Acerca

4.2.2.2.- La opción Calendario

El item Calendario del menú, abre un pequeño calendario el cual muestra el mes actual. Permite además ver otros meses. El calendario muestra el mes actual, resaltando la fecha del día actual. Los meses próximos y previos pueden ser visualizados usando las teclas “+” o “-” respectivamente. El mouse puede también ser usado para cambiar el mes mediante un click en ▼ o ▲. El calendario tiene el aspecto mostrado en la Figura 26.



Figura 26. Ventana de Calendario

4.2.2.3.- La opción Calculadora (Alt+C)

Cuando se escoge Calculadora, una calculadora de cuatro operaciones se abre en la pantalla, como se muestra en la Figura 27.

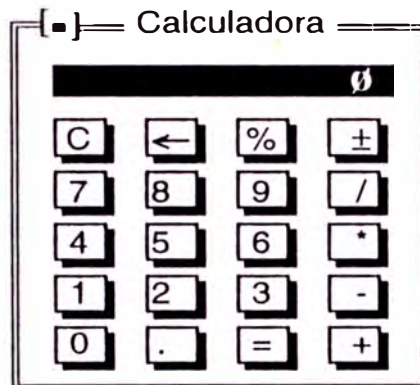


Figura 27. Ventana de Calculadora

Esta es una calculadora simple de 4 operaciones: adición, substracción, multiplicación y división. Para operar la calculadora se puede usar al teclado o presionando los botones con el mouse. La tecla “C” limpia al calculador, la tecla “←” borrará el último caracter tipeado, y el boton “±” permuta el valor de positivo a negativo o viceversa (en el teclado el equivalente de “±” es “_”).

4.2.3.- El menú Archivo (Alt+A)

Este menú contiene las funciones para abrir y salvar archivos de datos, los cuales son almacenados con la extensión DAL. Además, con él se puede cambiar el directorio actual, imprimir un archivo, salir temporalmente al DOS o salir totalmente de ALOG.

El menú Archivo es un menú desplegable con las opciones listadas verticalmente. Para seleccionar se utiliza las teclas FLECHA ARRIBA o FLECHA ABAJO para resaltar la opción y luego se pulsa ENTER. Algunas opciones tienen abreviaturas de teclado equivalentes (por ejemplo, F3 es para **Abrir** un archivo) que puede acelerar un poco las cosas, los usuarios de mouse sólo tienen que apuntar la opción y pulsar. El menú Archivo tiene el aspecto mostrado en la Figura 28.

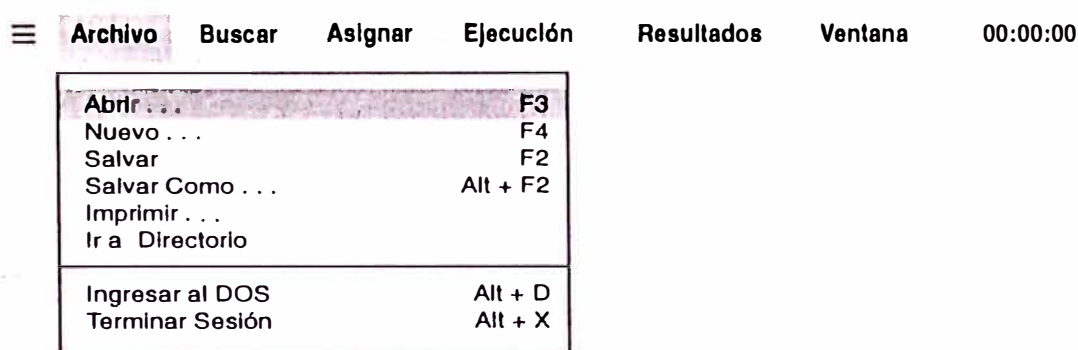


Figura 28. El Menú Archivo

4.2.3.1.- La opción Abrir (F3)

La opción abrir del Menú Archivo, permite seleccionar un archivo de una lista. El archivo seleccionado se abre posteriormente en su propia ventana.

Cuando se selecciona **Abrir...**, ALOG muestra un cuadro de diálogo, donde se podrá ingresar el nombre del archivo a abrir, incluyendo su ruta si es que se desea, o de lo contrario, ALOG usará el directorio actual para abrir el archivo. La ventana de diálogo se muestra en la Figura 29.

En la parte superior del cuadro se encuentra el campo Nombre, en él se puede escribir un nombre de archivo o una especificación de archivo, incluyendo los caracteres comodín * y ?. En la Figura 29 contiene *.DAL.

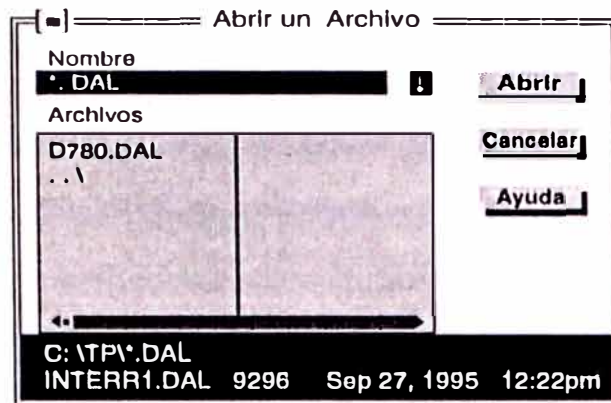


Figura 29. Ventana de Diálogo para abrir un archivo

Debajo del campo Nombre aparece una lista de todos los archivos que concuerdan con la especificación de archivo que se ha introducido. Esta lista contiene dos columnas, se puede pulsar con el mouse sobre el nombre del archivo deseado o pulsar la tecla TAB para mover el cursor hasta la lista de archivos y seleccionar el archivo deseado utilizando las teclas de flecha. Si los nombres de archivos que concuerdan no caben en las dos columnas, se puede desplazar el contenido de la ventana utilizando la barra de desplazamiento que hay debajo de los nombres de archivos. La barra de desplazamiento tiene flechas en ambos extremos y un selector de bloque entre las flechas. Para utilizar la barra de desplazamiento se puede pulsar el mouse sobre una de las flechas de la barra de desplazamiento o bien pulsar el botón izquierdo del mouse sobre el selector de bloque y desplazarlo hacia la izquierda o hacia la derecha.

En la parte derecha del cuadro de diálogo “Abrir Archivo” hay tres botones **Abrir**, **Cancelar** y **Ayuda**. Muchas ventanas en ALOG tienen botones similares, que pueden seleccionarse pulsando sobre ellos con el mouse, pulsando la tecla resaltada (por ejemplo, A para **Abrir**) o pasando con TAB al botón y pulsando ENTER. Si se selecciona el botón Abrir, ALOG crea una nueva ventana (hoja de trabajo) que contiene el archivo que

se ha pedido. El botón **Cancelar** cierra la ventana sin seleccionar ningún archivo (La tecla **Esc** siempre cancela una ventana de diálogo, incluso si el botón **Cancelar** no aparece) y el botón **Ayuda** proporciona una ayuda de acuerdo con el contexto para la ventana activa.

4.2.3.2.- La opción Nuevo (F4)

Cuando se selecciona **Nuevo** desde el menú **Archivo**, **ALOG** abre una ventana la cual es una nueva hoja de trabajo con el nombre por defecto: **Sin Título**, y automáticamente la hace activa para adquirir datos desde el interfaz a la vez que le asigna un número en la parte superior derecha de la hoja. Cabe resaltar que solo las primeras nueve ventanas que abran o que se ingresen como nuevas tendrán un número, más no las siguientes.

Adicionalmente luego de tomar los datos puede hacer uso de los menús **Resultados** y/o **Buscar** para poder analizar los datos. El aspecto de la hoja de trabajo o ventana de adquisición de datos nueva es mostrado en la Figura 30.

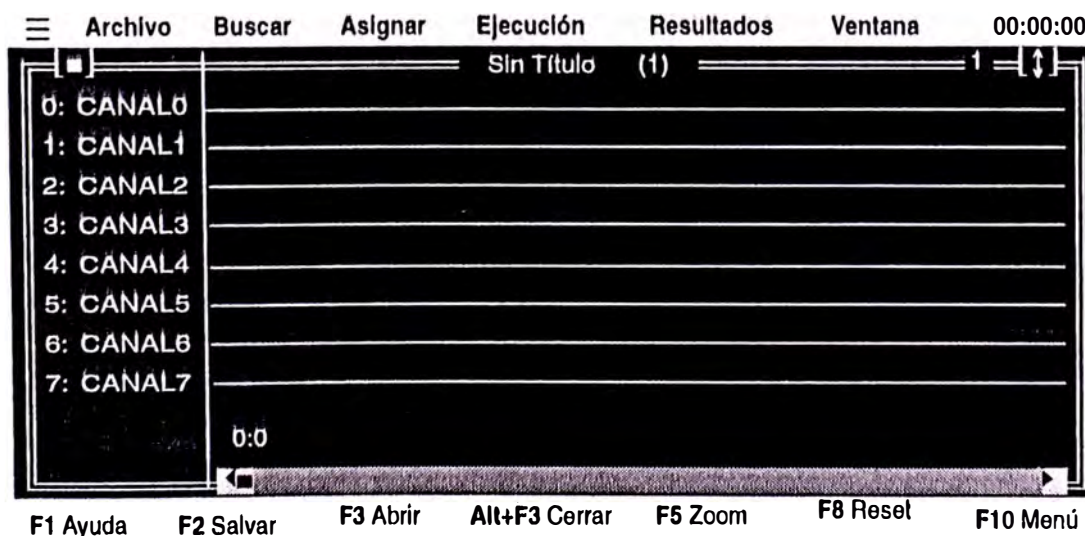


Figura 30. Hoja de Trabajo nueva (Sin Título)

La hoja de trabajo está dividida básicamente en dos partes, en la parte izquierda se listan los nombres de los canales que por defecto son: CANAL0, CANAL1, CANAL2, CANAL3, CANAL4, CANAL5, CANAL6 y CANAL7. En la parte derecha se van a representar los diagramas de tiempos de cada uno de los canales, que para empezar se encuentran todos los canales en nivel bajo. En la parte inferior se encuentran dos números separados por “:”, el número de la izquierda representa el número de muestra que se puede observar en la hoja de trabajo y oscila desde 0 hasta 8191; el número de la derecha representa la expresión hexadecimal del byte que forman los bits que aparecen a la izquierda del diagrama de tiempos de cada uno de los canales. El canal 7 es el más significativo y el canal 0 es el menos significativo.

Debajo de la representación de los diagramas de tiempo se puede observar una barra de desplazamiento que nos permite mover el diagrama de tiempos hacia la izquierda o hacia la derecha. La barra de desplazamiento tiene flechas en ambos extremos y un selector de bloque entre las flechas. Para utilizar la barra de desplazamiento se puede pulsar el botón izquierdo del mouse sobre una de las flechas de la barra de desplazamiento o bien pulsar el botón izquierdo del mouse sobre el selector de bloque y arrastrarlo hacia la izquierda o hacia la derecha.

El selector de bloque se mueve automáticamente así como el diagrama de tiempos con sólo presionar las siguientes teclas:

FLECHA IZQUIERDA: Que desplaza una muestra de los 8 canales a la izquierda.

FLECHA DERECHA: Que desplaza una muestra de los 8 canales a la derecha.

CONTROL+FLECHA IZQUIERDA: Que desplaza una cantidad de muestras igual al ancho actual de la hoja (que es redimensionable por el usuario) de adquisición a la izquierda.

CONTROL+FLECHA DERECHA: Igual que lo anterior pero para la derecha.

HOME: Regresa a la posición 0.

END : Que avanza hasta la última posición, es decir la 8191.

Si intenta salvar el contenido de una ventana de hoja de trabajo nueva, ALOG le pedirá que se renombre el archivo. En la hoja de adquisición de datos, se podrá usar los tres métodos para realizar el ingreso de datos desde la tarjeta de interfaz. Estos métodos se cubren en el menú **Ejecución** conjuntamente con los del menú **Asignar**.

4.2.3.3.- La opción Salvar (F2)

La opción salvar escribe en disco el contenido de la ventana de adquisición activa como un archivo con la extensión DAL y otro con la extensión BAK que es el de respaldo. Si la ventana de adquisición activa ha sido creada con la opción **Nuevo**, y si se ha adquirido datos en esa ventana, ALOG pedirá que se introduzca un nuevo nombre para el archivo; así como también cuando se quiera abandonar la sesión con ALOG y no se haya salvado los datos ingresados. Esto lo hace llamando a la ventana de diálogo **Salvar Archivo Como**, de modo que se pueda renombrar el archivo y salvarlo en un diferente directorio o en una unidad de disco diferente.

4.2.3.4.- La opción Salvar Como... (Alt+F2)

La opción **Salvar Como...** salva a disco el contenido de una ventana de adquisición activa en un archivo de **9296 bytes**, pero pide primero que se introduzca un nombre de archivo incluyendo su ruta. El nombre puede ser un nombre de archivo nuevo o el nombre de un archivo existente. Una vez seleccionado el nombre de archivo, la ventana de adquisición abierta para ese fichero se actualiza para reflejar el nuevo nombre seleccionado. **Salvar Como...**, abre la ventana de diálogo **Salvar Archivo Como** que se muestra en la Figura 31.

De acuerdo a la Figura 31, se puede salvar como un archivo de disco la hoja de Adquisición de datos activa, bajo un nombre diferente, un directorio diferente o un drive diferente. Luego seleccionar el nombre del archivo, junto con la ruta, escoger OK.

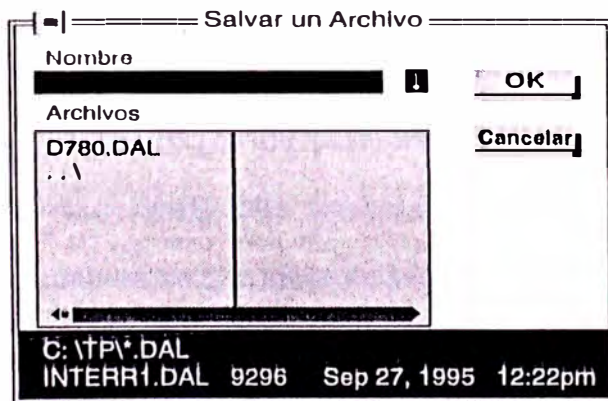


Figura 31. Ventana de Diálogo Salvar Como

4.2.3.5.- La opción Imprimir

La opción **Imprimir...** envía a la impresora el contenido de la ventana de adquisición activa. Con esta orden no se pueden imprimir ventanas de ayuda o similares.

La opción Imprimir muestra una ventana de diálogo, como se puede apreciar en la Figura 32.

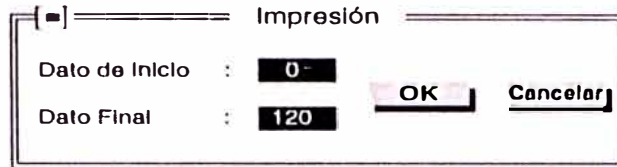


Figura 32. Ventana de Diálogo Imprimir

En la que se deberá ingresar el número de la muestra a partir del cual se desea empezar a imprimir así como también se deberá suministrar el dato final de impresión. Este último se tomará solamente como referencia debido a que la cantidad de datos impresos siempre es un múltiplo de 120, esto se debe a que se imprime en bloques de muestras de 120 en 120.

Si el dato final que se ingrese, respecto del inicial no llegue a ser una cantidad de datos requerida para imprimir un múltiplo de 120; entonces ALOG, colocará como dato final un número inmediato superior al que se ingresó; de manera tal que siempre sea la cantidad de muestras, a imprimirse, una cantidad múltiplo de 120.

La impresión coloca la fecha actual; la palabra de disparo y la cantidad de estados de pre-disparo son también colocados, si es que los hubiera.

El Modo de display Binario o Normal influye en la impresión; es decir la impresión se hará de acuerdo a lo que se tenga en pantalla, es decir datos binarios o líneas.

4.2.3.6.- La opción Ir a Directorio

La opción **Ir Directorio** permite cambiar el directorio en el que se encuentra actualmente. Cuando se elige esta opción, la ventana de diálogo "Ir a Directorio" que se muestra en la Figura 33, aparece en pantalla, en la cual se puede cambiar el directorio introduciendo la ruta deseada o seleccionar un directorio del árbol de directorios que se muestra en la misma ventana de diálogo.

Una vez seleccionado un directorio, se puede deshacer la selección utilizando el botón **Revertir**.

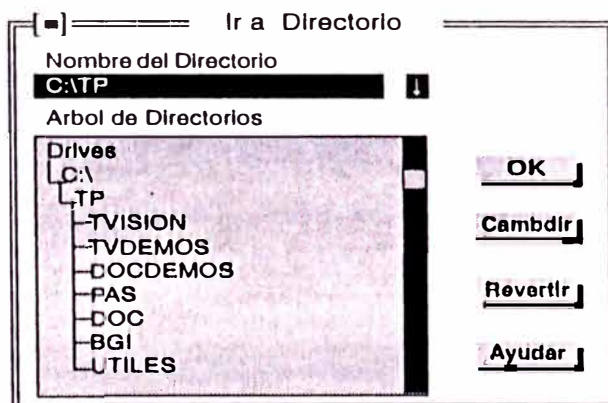


Figura 33. Ventana de Diálogo para Renovar Directorio

4.2.3.7.- La opción Salir al DOS (Alt+D)

Seleccionando la opción **Salir al DOS** se suspende temporalmente ALOG y se accede al DOS, donde se puede ejecutar órdenes del DOS o ejecutar otros programas que no sean demasiado extensos. Para volver al entorno de ALOG basta ejecutar **EXIT** en el prompt del DOS.

4.2.3.8.- La opción Terminar Sesión (Alt+X)

La opción **Terminar Sesión** finaliza la ejecución del programa ALOG y devuelve el control al DOS. Si al seleccionar **Terminar Sesión** existe una hoja de adquisición que contiene a un archivo modificado, ALOG preguntará si se desea salvar el archivo antes de salir al sistema operativo.

4.2.4.- El menú Buscar (Alt+B)

El análisis de datos adquiridos es en gran parte una tarea de búsqueda de ciertos instantes o condiciones que surgen en los canales. El menú Buscar que se muestra en la Figura 34, proporciona una serie de servicios que pueden ayudar en esas tareas.

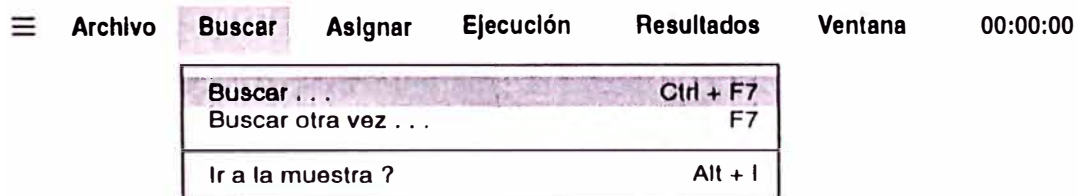


Figura 34. El Menú Buscar

4.2.4.1.- La opción Buscar (Ctrl+F7)

La opción **Buscar...** permite localizar un patrón de bits de los ocho canales. Cuando se selecciona esta opción, aparece en la pantalla la ventana de diálogo "Buscar" como se muestra en la Figura 35.

En este cuadro de diálogo se puede escribir el número en el sistema binario, teniendo en cuenta que el canal 7 contiene el bit más significativo y el canal 0 el bit menos significativo.

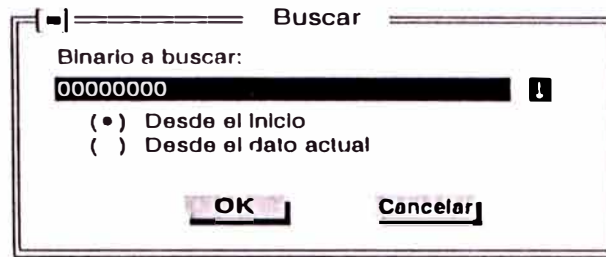


Figura 35. Ventana de Diálogo Buscar

La búsqueda la realiza en forma paralela, usando los bit de los ocho canales. Si no interesa el nivel lógico de ciertos bits o los considera términos que no importan, puede asignarle la variable "x" a estos bits. Por ejemplo, si solamente se desea encontrar un dato el cual sus 2 bits más significativos sean unos, deberá escribir en la ventana de diálogo 11XXXXXX.

Adicionalmente, existen dos opciones en donde puede seleccionarse la búsqueda; si se va a realizar desde el comienzo, o desde el dato actual que se encuentra en la pantalla hacia adelante.

4.2.4.2.- La opción Buscar otra vez (F7)

La opción **Buscar otra vez...**, simplemente repite la última operación de búsqueda realizada; y ésta se puede seleccionar desde el menú o pulsando la tecla de función F7. El dato ingresado a buscar en la ventana de diálogo Buscar, permanece cuando se escoje Buscar Otra Vez.

4.2.4.3.- La opción Ir a la muestra ? (Alt+I)

Cuando se selecciona la opción **Ir a la muestra ?** aparece en pantalla una ventana de diálogo la cual se muestra en la Figura 36, en la cual se puede

escribir un número comprendido entre 0 y 8191 para mostrar en pantalla a partir del número elegido la cantidad de datos que se puedan observar en la ventana.

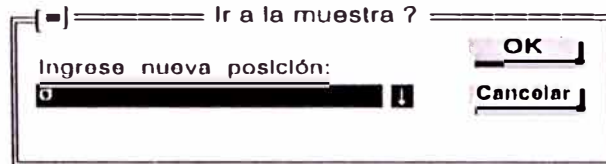


Figura 36. Ventana de Diálogo Ir a la Muestra ?

4.2.5.- El menú Asignar (Alt+S)

El menú de asignaciones permite especificar datos que inicialmente deben ser colocados para que se produzca un buen disparo así como también los nombres de las etiquetas que lucirán en la pantalla. El menú Asignar es el que se muestra en la Figura 37.

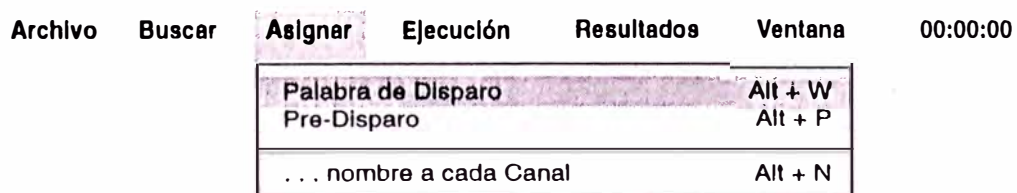


Figura 37. El Menú Asignar

4.2.5.1.- La opción Palabra de Disparo (Alt+W)

Con la opción **Palabra de Disparo** se podrá ingresar la palabra en Hexadecimal (4 nibbles) en la cual el analizador comenzará la toma de datos en el Modo de trabajo con Pre-Disparo y Automático. Esta Palabra de

Disparo no es tomada en cuenta cuando se trata de una Adquisición en el Modo Manual. Cuando se selecciona esta opción, aparece una ventana de diálogo como la que se muestra en la Figura 38.

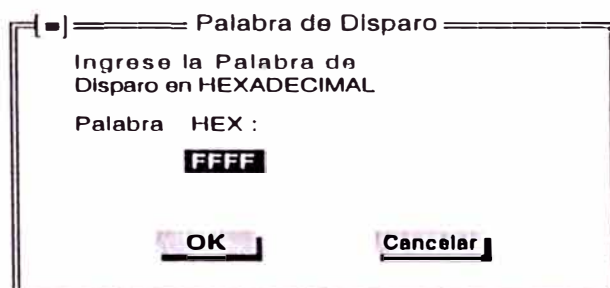


Figura 38. Ventana de Diálogo Palabra de Disparo

De acuerdo a la ventana de diálogo mostrada en la Figura 38 se podrá ingresar el número en hexadecimal (siendo válidos los caracteres 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, a, B, b, C, c, D, d, E, e, F, f) y luego se presionará ENTER para que sea tomado en cuenta para el disparo.

4.2.5.2.- La opción Pre-Disparo (Alt+P)

La opción **Pre-Disparo** abre una ventana de diálogo como se aprecia en la Figura 39. De acuerdo a esta ventana de diálogo, se podrá ingresar la cantidad de datos a mostrar antes de que ocurra el disparo, ésta cantidad tendrá que ser un múltiplo de cuatro comprendido desde 4 hasta 8188.

Si no se ingresa un número múltiplo de cuatro, ALOG colocará como dato interno de pre-disparo el múltiplo de cuatro inmediato inferior al que se ingresó.

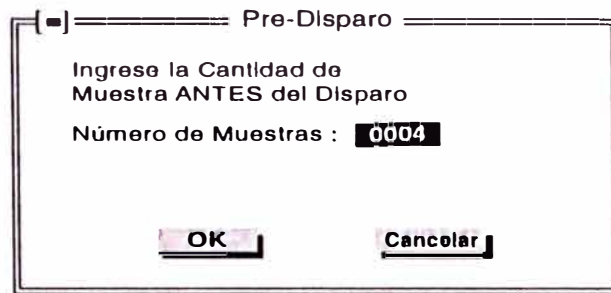


Figura 39. Ventana de Diálogo Pre-Disparo

4.2.5.3.- La opción Nombre a cada Canal (Alt+N)

La opción **...Nombre a cada Canal** permite ingresar o modificar el nombre de cada Canal, incluso luego de haber tomado los datos desde el interfaz. Las etiquetas iniciales, cada vez que se coloca en el desktop una nueva hoja de adquisición de datos mediante F4 son: CANAL0, CANAL1, CANAL2, CANAL3, CANAL4, CANAL5, CANAL6 y CANAL7.

Al seleccionar esta opción se mostrará una ventana de diálogo como se muestra en la Figura 40. Para pasar de etiqueta en etiqueta, se debe presionar TAB o Shift+TAB, para ir hacia adelante o retroceder respectivamente. La cantidad de caracteres como máximo a ingresar es de 7.

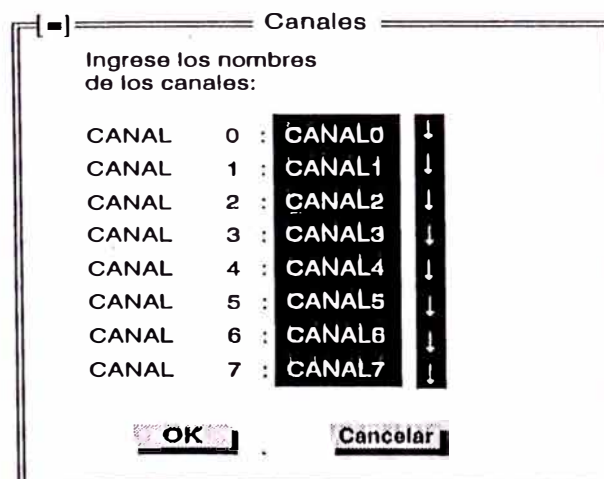


Figura 40. Ventana de Diálogo para nombrar a los canales

4.2.6.- El menú Ejecución (Alt+E)

El menú Ejecución contiene comandos para poder realizar la adquisición de datos desde la tarjeta de interfaz y mostrarlos en pantalla para su análisis.

Las modalidades de funcionamiento del analizador lógico son: Con Pre-Disparo y Sin Pre-Disparo, en este último modo se contempla el Modo Manual y el Automático.

El menú Ejecución es el que se muestra en la Figura 41.



Figura 41. El menú Ejecución

4.2.6.1.- La opción Con Pre-Disparo (Ctrl+Insert)

Bajo esta modalidad de trabajo del Analizador Lógico, la adquisición de datos se realiza tomando en cuenta la palabra de disparo y la cantidad de datos de Pre-Disparo introducidos en el menú **Asignar**.

Por lo tanto, el disparo se realiza cuando se igualan los 16 bits ingresados (en hexadecimal) en la Palabra de Disparo con el que físicamente se encuentre en los 16 bits (TA0-TA15) de la tarjeta de interfaz. Cuando se realiza el disparo en este modo, sonará un pitido e inmediatamente se mostrará en la pantalla los datos tomados.

4.2.6.2.- La opción Sin Pre-Disparo

La opción **Sin Pre-Disparo** hace aparecer un sub-menú, que se muestra en la Figura 42.

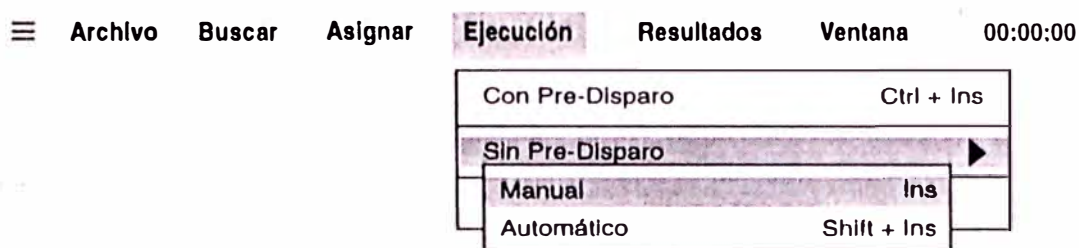


Figura 42. Opciones del sub-menú Sin Pre-Disparo

Como podemos observar para la toma de datos Sin Pre-Disparo se contempla los modos: Manual y Automático.

a) En la opción **Manual (Ins)**, la adquisición de datos se realiza a partir de la presión de la tecla Insert. Este modo obviamente no contempla los datos palabra de disparo y la cantidad de datos de Pre-Disparo introducidos en el menú **Asignar**.

Luego de solicitar esta opción, deberá sonar un pitido, el cual nos indicará que la adquisición ha sido satisfactoria e inmediatamente se mostrará en pantalla los datos tomados.

b) En la opción **Automático (Shift+Ins)** los datos empiezan a ser tomados a partir del disparo automático, esto es, luego de que se equiparen los 16 bits ingresados en hexadecimal en la Palabra de Disparo con los que físicamente se encuentren en los 16 bits (TA0-TA15) de la tarjeta de interfaz.

Los datos de Pre-Disparo ingresados en el menú **Asignar** no son tomados en cuenta. Cuando se realiza el disparo automático sonará un pitido e inmediatamente se mostrará en pantalla los datos adquiridos.

4.2.6.3.- La opción Reset (F8)

Con esta opción se logra resetear los registros internos de la tarjeta de interfaz. Deberá usarse esta opción cuando se tenga problemas con el hardware del interfaz. De esta manera quedará listo para cualquier forma de toma de datos del menú Ejecución.

4.2.7.- El menú Resultados (Alt+R)

El menú **Resultados** contiene comandos para poder modificar la presentación de los datos adquiridos, luego de haber ejecutado la toma de datos bajo los modos contemplados en el menú **Ejecución**, y dar facilidad para un mejor análisis de los datos adquiridos. La Figura 43 muestra las opciones del menú **Resultados**.

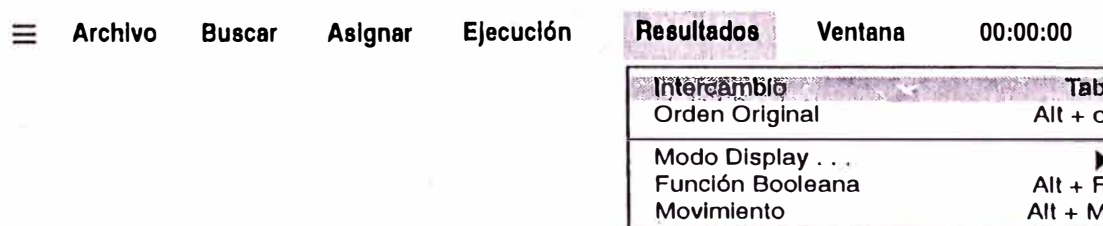


Figura 43. El Menú Resultados

4.2.7.1.- La opción Intercambio (Tab)

Esta opción proporcionará una ventana de diálogo como se muestra en la Figura 44.

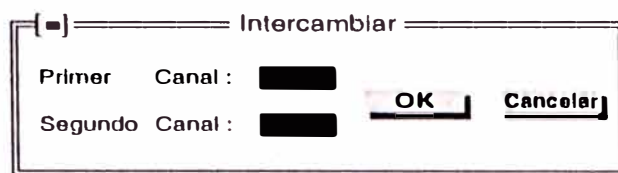


Figura 44. Ventana de Diálogo de la opción Intercambiar

De acuerdo a lo mostrado en la Figura 44 se podrá intercambiar dos canales (Primer y Segundo). La ventana de diálogo solicita el número de los dos canales a ser intercambiados. Puede pasar de Primer canal a Segundo canal o hacia los botones OK o Cancelar, presionando reiteradas veces TAB o Shift+TAB. Una vez que se digiten los números de los canales a intercambiar, presionar ENTER e inmediatamente se observará en pantalla el intercambio tanto de las formas de onda como de sus respectivas etiquetas.

Este intercambio de canales es simplemente lógico, es obvio que físicamente nada ha sido intercambiado. Por ejemplo los datos que ingresan por la sonda etiquetada por CH5 sigue ingresando y se observará para futuras tomas en la posición 5: de la pantalla. Sin embargo luego de intercambiar dos canales, realiza una grabación a disco, los datos son guardados a disco con los dos canales intercambiados. Esta opción se puede usar para comparar más de cerca dos canales, así como para reordenar.

4.2.7.2.- La opción Orden Original (Alt+O)

Con esta opción se puede restaurar la posición original de cada uno de los canales, tal y conforme están físicamente conectados a la tarjeta de interfaz. Este comando se usa para restaurar los cambios hechos luego de usar Intercambio.

4.2.7.3.- La opción Modo Display

La opción **Modo Display...** hace aparecer un sub-menú, que se muestra en la Figura 45.

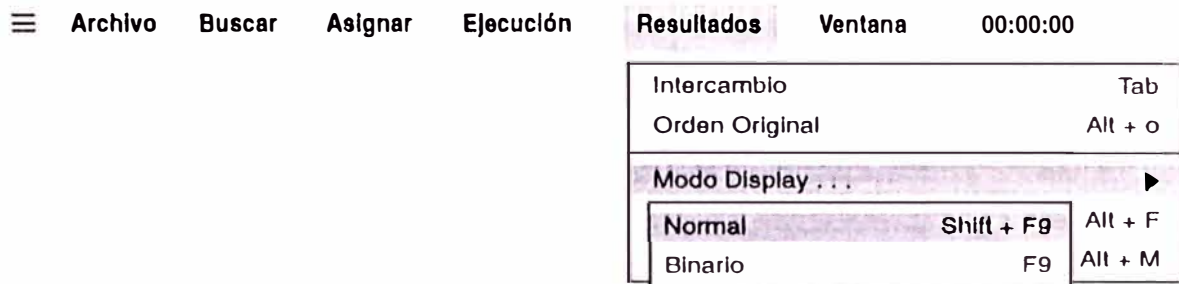


Figura 45. Opciones del sub-menú Modo Display

Con este sub-menú se dispone de dos modos de presentar la información en la pantalla: Modo Normal y Modo Binario.

a) En el modo **Normal (Shift+F9)**, que es el estado de presentación en la pantalla por defecto, los datos se muestran en forma de diagrama de tiempos.

Las transiciones de alto a bajo y viceversa, tienen una cota de error absoluto máximo de 50ns que es el periodo de la señal de 20Mhz.

b) En el Modo **Binario (F9)**, las tomas hechas a razón de 20Mhz (cada 50ns), son mostrados en la pantalla como unos ó ceros. Como en el caso anterior, las transiciones de un “1” a un “0” y viceversa, tienen una cota de error absoluto máximo de 50ns.

4.2.7.4.- La opción Función Booleana (Alt+F)

Este comando brinda una ventana de diálogo la que se muestra en la Figura 46.

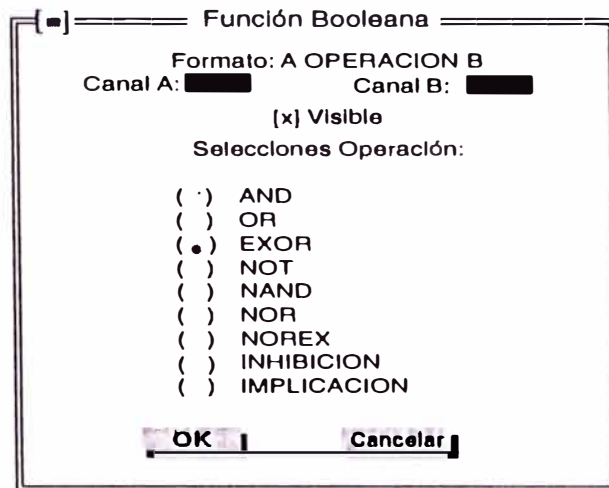


Figura 46. Ventana de Diálogo Función Booleana

En la ventana de Diálogo de la Figura 46, se puede realizar una operación entre dos canales. En primer lugar tendrá que presionar TAB para poder ubicarse en el Canal A, y allí se colocará el número del canal que se desee, nuevamente se presionará TAB para pasar al recuadro de Canal B y se hará lo mismo; presionando un nuevo TAB se ubicará en un campo en el que tiene la etiqueta de Visible y un recuadro en el que se encuentra una aspa, indicando que el resultado de la operación va a salir en pantalla. Cuando se necesite que la función respuesta ya no aparezca en pantalla, se tendrá que presionar la barra espaciadora y el aspa desaparecerá. Y volverá a aparecer si es que se presiona la barra espaciadora nuevamente, haciendo que la función nuevamente se observe en la pantalla.

Por último se presionará otra vez TAB para pasar al ambiente donde se seleccionará la función booleana, para lo cual se tendrá que presionar las teclas FLECHA ARRIBA y FLECHA ABAJO, para moverse entre una y otra función. Una vez seleccionada la función se presionará ENTER e inmediatamente aparecerá en la hoja de adquisición debajo del canal 7, como un noveno canal, el resultado de la operación solicitada.

Las funciones booleanas que se contemplan son las siguientes:

Función	Operación
AND	$A.B$
OR	$A + B$
EXOR	$A.B + \bar{A}.\bar{B}$
NOT	\bar{A}
NAND	$\overline{A.B}$
NOR	$\overline{A + B}$
NOREX	$A.B + \bar{A}.\bar{B}$
INHIBICION	$A.\bar{B}$
IMPLICACION	$\bar{A} + B$

4.2.7.5.- La opción Movimiento (Alt+M)

Esta opción permite visualizar todas las muestras de todos los canales desplazando la pantalla hacia la derecha, hasta que se presione una tecla o hasta que llegue a la última posición es decir la 8191.

4.2.8.- El menú Ventana (ALt+V)

Este menú proporciona un control absoluto sobre todos los aspectos del manejo de ventanas, en él se encuentran todos los comandos de manipulación de ventanas tales como mover ventanas, realizar un zoom de una ventana, redimensionarla, etc. El menú Ventanas, se muestra en la Figura 47.

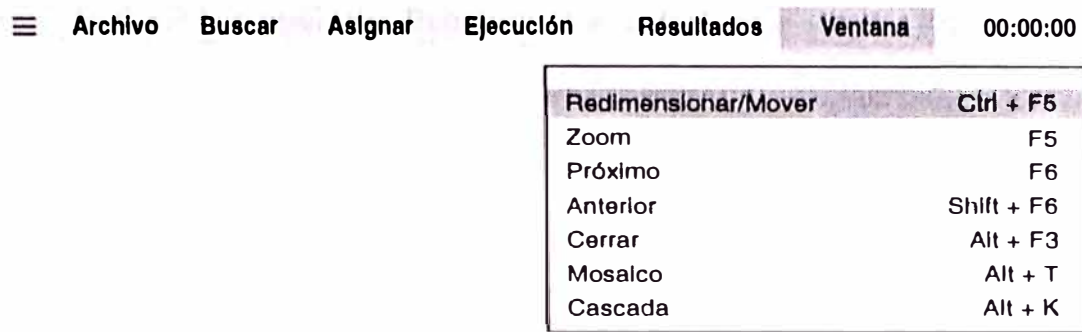


Figura 47. Las opciones del Menú Ventana

4.2.8.1.- La opción Redimensionar/Mover (Ctrl+F5)

Cuando se selecciona la opción **Redimensionar/Mover** se puede mover la ventana activa por la pantalla utilizando las teclas de flecha. Para contraer o expandir el tamaño de la ventana basta con mantener pulsada la tecla SHIFT y utilizar las teclas de flecha hasta que el tamaño de la ventana sea el deseado. Si se usa el mouse se deberá dar un click en la esquina inferior derecha y arrastrar hasta llegar a la posición deseada.

4.2.8.2.- La opción Zoom (F5)

Seleccionando **Zoom** se expande la ventana activa hasta el tamaño de la pantalla completa. Seleccionando nuevamente zoom, devuelve la pantalla al tamaño y posición anteriores.

También se puede dar un doble-click en cualquier parte de la barra del título (excepto donde aparece el símbolo [■]).

4.2.8.3.- Las opciones Próximo (F6) y Anterior (Shift+F6)

Pulsando la tecla F6 o seleccionando desde el menú Ventana el ítem Próximo, se obtiene una forma rápida de pasar a la siguiente ventana de acuerdo con la orden de numeración.

Si se pulsa las teclas Shift y F6 o se selecciona desde el menú Ventana el ítem Anterior, se consigue una forma rápida de pasar a la anterior ventana de acuerdo con la orden de numeración. Estas opciones son empleadas cuando se tiene varias hojas de trabajo abiertas.

4.2.8.4.- La opción Cerrar (Alt+F3)

La opción **Cerrar** elimina la ventana activa de la pantalla. Si la ventana contiene un archivo que ha sido modificado, se preguntará si se quiere salvar el archivo antes de que se cierre la ventana. También se puede dar un click con el mouse sobre el símbolo [■] ubicado en la parte superior izquierda para cerrar la ventana.

4.2.8.5.- La opción Mosaico (Alt+T)

Cuando se tienen muchas ventanas abiertas a la vez, las cosas pueden resultar un tanto confusas. Se recomienda utilizar la orden Mosaico para encoger todas las ventanas hasta un tamaño que permita que se visualicen todas en la pantalla, como se muestra:

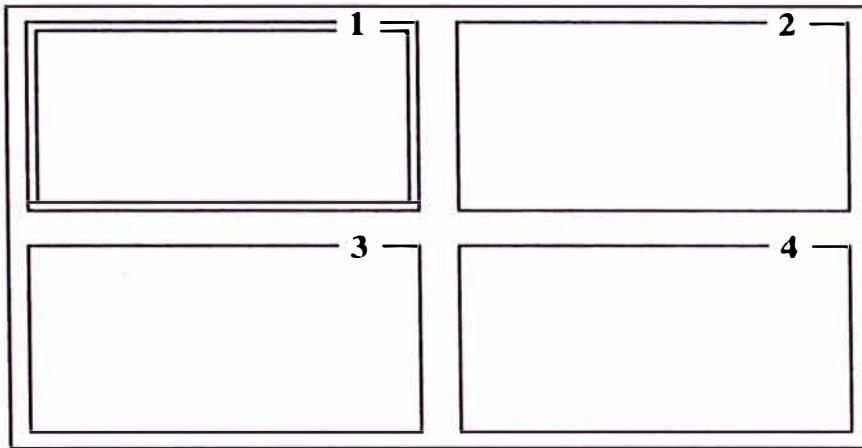


Figura 48. Ventanas dispuestas en Mosaico

Para seleccionar la ventana que se desea que esté activa, basta con mantener la tecla Alt mientras se pulsa el número de la ventana elegida, si es que esta tiene numeración, como se indicó solo las 9 primeras ventanas tienen numeración.

4.2.8.6.- La opción Cascada (Alt+K)

Esta opción es otra forma de ordenar las ventanas. En lugar de reducirlas, las coloca en cascada, de forma que queda visible la línea superior de cada ventana, como se muestra en la Figura 49.

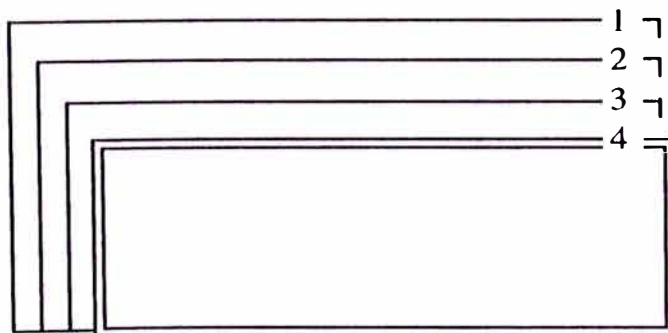


Figura 49. Ventanas colocadas en Cascada

Debemos mencionar que el software muestra mensajes de error cada vez que se ingrese en las ventanas de diálogo un dato erróneo, lo que nos entrega la posibilidad de volver a solicitar la ventana para corregir el error.

4.3.- El Manejador de Interrupciones

La interrupción usada por nuestro programa es la interrupción de hardware IRQ9; ésta a la vez llama la interrupción de software INT 0Ah, por lo que para poder usarla adecuadamente se ha cambiado el vector de interrupción de la interrupción INT 0Ah por otro que contiene la dirección de nuestro propio manejador de interrupción llamado: **ManejaInt** (esta rutina se encuentra en el Apéndice B). El cambio se hace cuando se corre el programa principal; y cuando se sale del programa se devuelve el vector de interrupción original a la INT 0Ah del sistema.

El manejador de interrupciones es una rutina cuya labor es la de realizar el ingreso de datos desde la memoria de la tarjeta de interfaz hacia la memoria del computador, además tiene la tarea de separar los datos ya que como se sabe, los datos de un canal, que provienen de la memoria del interfaz, vienen unidos a los de otro canal. Así la rutina que maneja la interrupción, deja todo listo para que las rutinas de visualización presenten los datos en pantalla para su análisis.

Como se sabe existen tres modos de disparo: Manual, Automático y Con Pre-Disparo, los cuales son implementados en tres rutinas que preparan a la tarjeta de interfaz para que adquiera datos; luego de hacer esto, dichas rutinas esperan a que en la tarjeta de interfaz ocurran las condiciones de disparo. Si estas condiciones se dan, se adquieren los datos en las memorias del interfaz, y cuando finaliza, la interfaz produce un pulso en la línea IRQ9 solicitando la interrupción, lo que hace que se ejecute **ManejaInt**, el cual al término de su ejecución produce un pequeño sonido (pitido), resetea la tarjeta de interfaz y por último

cambia una clave devolviendo el control a la rutina que quedó esperando, la misma que al detectar la clave inmediatamente llama a las rutinas de visualización para que actualicen la pantalla presentando así a los datos recientemente adquiridos.

En este capítulo se ha presentado el software del sistema, que se desarrolla en el computador personal, orientado a su manejo. En el Apéndice B se lista el programa fuente, es decir el programa principal llamado **program LOGIC** y sus respectivas unidades las cuales son **Analizer, Hcxxxx, AlogCmds y Apoyo**. Las unidades que también son usadas son: Dos, Objects, Drivers, Memory, Views, Menus, Dialogs, StdDlg, MsgBox, App, Crt, Gadgets, Calendar, Calc, HelpFile y Printer; todas ellas pertenecientes a la versión 6 de Turbo Pascal, y no son listadas. En el siguiente capítulo se explica el manejo del sistema desarrollado, y a la vez se realizarán pruebas experimentales.

CAPITULO V OPERACION DEL ANALIZADOR LOGICO Y RESULTADOS EXPERIMENTALES

En este capítulo describiremos la operación del analizador lógico, es decir la manera como el operador deberá manejar tanto el hardware como el software, para poder así analizar los diagramas de tiempos que involucran tanto los circuitos digitales y aquellos basados en microprocesadores. También en este capítulo haremos el análisis de un sistema basado en microprocesador, mostrando los resultados obtenidos.

5.1.- Instalación del Interfaz

En primer lugar se deberá verificar que ninguna tarjeta del computador personal use la Interrupción de Hardware IRQ9 y además que las direcciones de puerto usadas, es decir 2F0h, 2F1h, 2F2h y 2F3h también no sean utilizadas, eso se podrá hacer usando cualquier utilitario como el Norton, Checkit etc. Luego se procederá a instalar la tarjeta de interfaz en cualquiera de los slots de ocho bits de la PC (Bus ISA).

5.2.- Ejecución Básica del Software

Luego de hacer la instalación de la Tarjeta de Interfaz debemos ejecutar el programa ALOG.EXE desde el prompt del DOS. Este programa tiene como acompañante un archivo de ayuda llamado ALOG.HLP al cual ALOG.EXE recurrirá cuando el usuario solicite ayuda de las diversas opciones que contiene este software.

Después de escribir ALOG desde el prompt del DOS y presionar ENTER, aparecerá el mensaje de presentación como se muestra en la Figura 50.

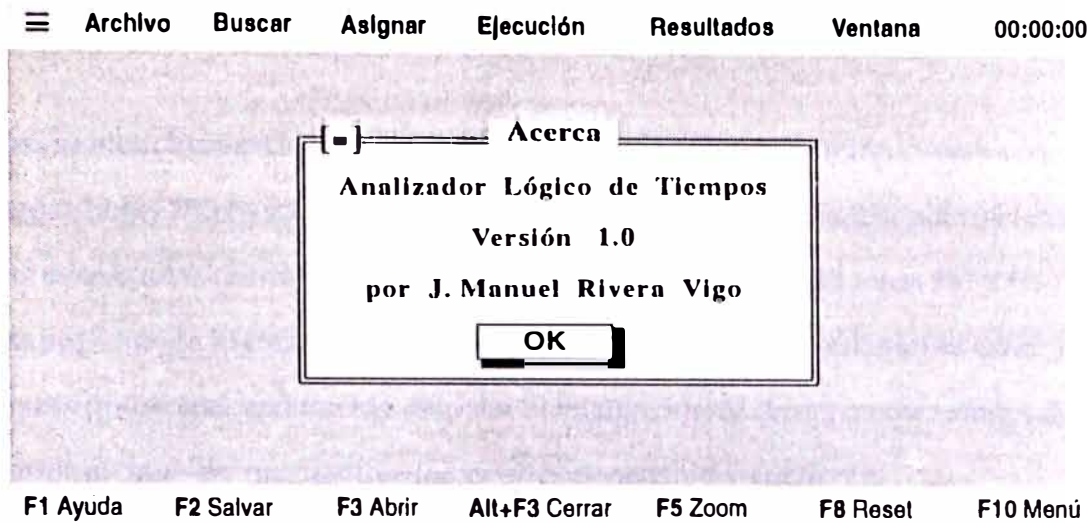


Figura 50. Presentación del programa ALOG

Si al presionar ENTER, saliese el mensaje de error mostrado en la Figura 51, es porque la tarjeta de interfaz no se está comunicando bien con el software, lo cual indica que tendrá que verificar la instalación. Lo mismo ocurre cuando se pretende usar el software sin que se haya instalado la Tarjeta de Interfaz. Si se presiona Enter o si se da un click con el ratón en OK, se saldrá al prompt del DOS nuevamente. Por el contrario, si es que no existiese ningún problema con la Tarjeta de Interfaz, luego del mensaje de presentación al presionar ENTER, se ingresará correctamente al programa que maneja al interfaz. De esta manera se está validando la comunicación entre el hardware y el software del sistema.

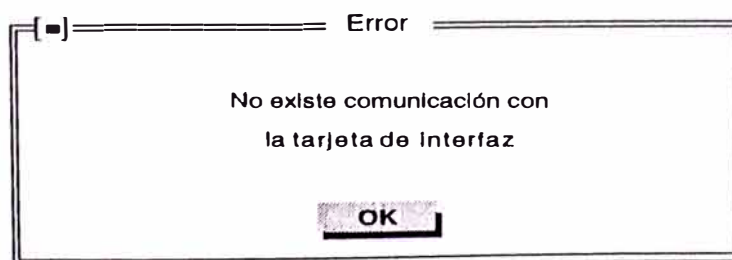


Figura 51. Mensaje de Error de Comunicación

5.3.- Instalación de las Sondas de la Tarjeta

Como se mencionó en los capítulos II y III, la tarjeta de adquisición cuenta con ocho sondas etiquetadas con CH0 hasta CH7 para poder sensar los ocho canales; además cuenta con 16 entradas etiquetadas como TA0-TA15, más 2 entradas de habilitación E0 y E1. Todas estas entradas soportan la lógica TTL y deberán ser instaladas apropiadamente en el circuito bajo prueba para poder adecuadamente disparar el analizador. Si desea medir señales de otra lógica deberá usar un interfaz que cambie los niveles de tensión y corriente.

En cuanto a las sondas CH0-CH7 se instalarán en donde se quiera observar las formas de onda de aquellas señales. La instalación de las sondas TA0-TA15, E0 y E1 en el circuito bajo prueba dependerá de la forma en que se desee disparar al analizador lógico.

Si el modo que se pretende usar es el Manual, entonces no se requiere que esas entradas se instalen, pueden quedar libres de conexión o conectadas a tierra. Por el contrario, si se va a disparar usando el modo Con Pre-Disparo o el modo Automático, entonces esas entradas deberán conectarse al circuito bajo prueba para servir de referencia para que se pueda producir el disparo del analizador lógico.

5.4.- Condiciones Iniciales

En primer lugar debemos solicitar una hoja de adquisición vacía, como se muestra en la Figura 52. Esta hoja se invoca desde el menú Archivo, seleccionando **Nuevo** o usando la tecla de función **F4**, lo que hace que se ubique en pantalla una hoja de adquisición no titulada. En esta hoja se ubicarán los resultados luego del disparo. Como se puede apreciar por defecto los nombres de los canales son CANAL0, CANAL1, etc. Esto se puede modificar ingresando al menú Asignar seleccionando **...nombre a cada Canal** o simplemente presionando **Alt+N**. No obstante los cambios de nombre pueden hacerse incluso luego de haber capturado los datos.

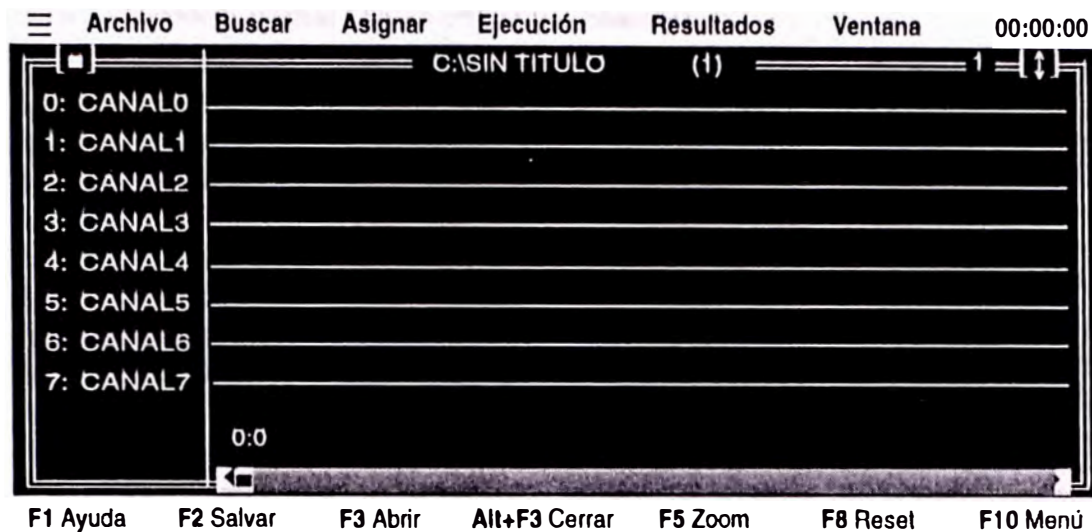


Figura 52. Hoja de Adquisición Sin Título

En el mismo menú Asignar se puede ingresar la Palabra de Disparo que va a compararse con los de las sondas TA0-TA15; este dato es necesario si es que el modo de disparo es diferente del modo Manual. Se puede invocar en forma rápida presionando la combinación Alt+W. También en el menú Asignar se puede seleccionar el Pre-Disparo, en donde se ingresará la cantidad de datos previos al disparo; obviamente este dato sólo será tomado en consideración cuando se trabaje con el modo Con Pre-Disparo. De esta manera se deja todo listo para producir el disparo del analizador.

5.5.- Disparo del Analizador

Para producir el disparo se tendrá que elegir uno de los 3 modos de disparo implementados en el proyecto. La elección del modo será de acuerdo a las características del circuito que se esté probando; los modos de disparo se encuentran en el menú Ejecución.

Si se usa el Modo Manual, el cual se invocará mediante la presión de la tecla **Insert**, se obliga a que la tarjeta adquiera datos inmediatamente, los cuales son presentados en pantalla luego de un pitido.

Ya que en este modo el disparo es de forma manual, no interesa la precisión del inicio del disparo. Obviamente la orden de disparo parte del teclado del computador para luego transmitirse por un puerto de salida (PS 1) y así la unidad de control de la tarjeta ordene la toma de datos.

Cuando se usa el Modo Automático, el cual se invoca mediante la presión simultánea de **Shift+Insert**, produce el siguiente mensaje en la parte inferior de la hoja de Adquisición:

Presione cualquier tecla para detener el Muestreo.

Si se presiona cualquier tecla, se cancela la opción, de lo contrario se tendrá que esperar a que se produzca un disparo, es decir cuando ocurra que los datos del circuito bajo prueba ingresados en TA0-TA15 sean iguales a los que fueron colocados en la ventana de diálogo Palabra de Disparo.

Cuando se usa el Modo Con Pre-Disparo, el cual se invoca mediante la presión simultánea de **Ctrl+Insert**, produce el mismo mensaje que en el modo automático, con las mismas características. En este modo también se espera a que se iguale los datos del circuito bajo prueba que ingresan por TA0-TA15 con los de Palabra de Disparo, para que pueda tener lugar el disparo. En este modo los datos ingresados en la ventana de diálogo Pre-Disparo son usados para obtener la cantidad de datos previos al disparo.

5.6.- Análisis de la información adquirida

Luego de haber disparado al analizador, automáticamente se presentará en la pantalla los datos adquiridos, con lo que se podrá usar las opciones del menú Resultados así como los del menú Buscar, para analizar los datos. Así se podrá:

- Encontrar un patrón de bits.
- Encontrar nuevamente los mismos datos.
- Intercambiar líneas.
- Recuperar el orden original.
- Modificar el modo de presentación en la pantalla.
- Realizar una función booleana entre los canales.
- Forzar a un movimiento hacia la izquierda.

5.7.- Análisis de un Sistema basado en el μ P Z80

En esta última parte vamos a utilizar el analizador lógico para estudiar los diagramas de tiempos de un sistema basado en el microprocesador de 8 bits Z80 de Zilog. El sistema es un Microprofessor (MPF-I de Multitech Industrial Corp.) cuyas especificaciones principales de hardware son:

1) CPU:

Zilog Z-80 CPU con 158 instrucciones y con 2.5 Mhz de frecuencia de la señal de reloj como máximo. Para el MPF-I, el sistema de reloj es de 1.79Mhz.

2) ROM:

EPROM 2516(2532)X 1, TOTAL 2K(4K) bytes. Las direcciones del programa monitor en EPROM: 0000-07FF(0FFF).

3) RAM:

RAM estática: 6116, total 2K bytes. Direcciones de la RAM: 1800-1FFF.

4) Area de Expansión de Memoria:

EPROM 2516/2716/2532/2732 o una RAM estática 6116 en las direcciones: 2000-2FFF.

5) Puertos de Entrada/Salida (I/O):

Intel 8255, Interfaz Paralela Programable con un total de 24 líneas I/O paralelas.

Direcciones I/O: 00-03H.

Z80 PIO, Interfaz Paralela Programable con un total de 16 líneas I/O paralelas.

Direcciones I/O: 80-83H.

Z80 CTC, Interfaz Contador/Temporizador.

Direcciones I/O: 40-43H.

6) Display: 6 displays de 7-Segmentos de 0.5" rojos.

7) Teclado: 36 teclas que incluyen: 19 teclas de función, 16 teclas hexadecimal y una tecla de usuario.

A continuación se presenta algunos ejemplos de captura de datos del sistema basado en el Z80; usando los métodos de disparo convenientemente para cada caso.

5.7.1.- Prueba 1

En primer lugar vamos a capturar con el analizador los diagramas de tiempo de las señales: CLK, M1, $\overline{\text{MREQ}}$, RD, WR, $\overline{\text{IORQ}}$, NMI y $\overline{\text{RFSH}}$, luego de que se

produzca en el sistema una interrupción no enmascarable. Para lo cual deberán colocarse las sondas como sigue:

CH 0: CLOCK	(pin 6)
CH 1: MI	(pin 27)
CH 2: MREQ	pin 19)
CH 3: RD	(pin 21)
CH 4: WR	(pin 22)
CH 5: IORQ	(pin 20)
CH 6: NMI	(pin 17)
CH 7: RFSH	(pin 28)

Los pines están referidos al Z80 CPU. Ahora debemos ubicar una hoja de adquisición en pantalla, esto lo hacemos presionando F4. Para solucionar el problema debemos usar el método de disparo Automático. Como se sabe en este modo se deberá especificar una palabra de disparo para que se compare con el bus TA0..TA15, pero como sólo se tiene una señal para el disparo la cual es el pin 17 del Z80 CPU (NMI) entonces todos los pines del bus TA serán colocados a "1", mientras que los habilitadores E0 y E1 son unidos al pin 17 de P Z80 de manera que podamos disparar al analizador cuando sucede el pulso negativo de NMI.

En el Microprofessor es fácil producir una Interrupción No Enmascarable, debido a que ella se produce mediante la presión de la tecla MONI.

En cuanto al software debemos en primer lugar etiquetar los nombres de los canales, luego como palabra de disparo colocaremos FFFFh y por último activar el modo Automático mediante la presión de las teclas **Shift+Insert**. Hasta aquí el analizador esta en estado de espera de que se produzca la interrupción NMI.

Para producir el NMI se presionará la tecla MONI con lo que el analizador mostrará una pantalla como se muestra en la Figura 53.

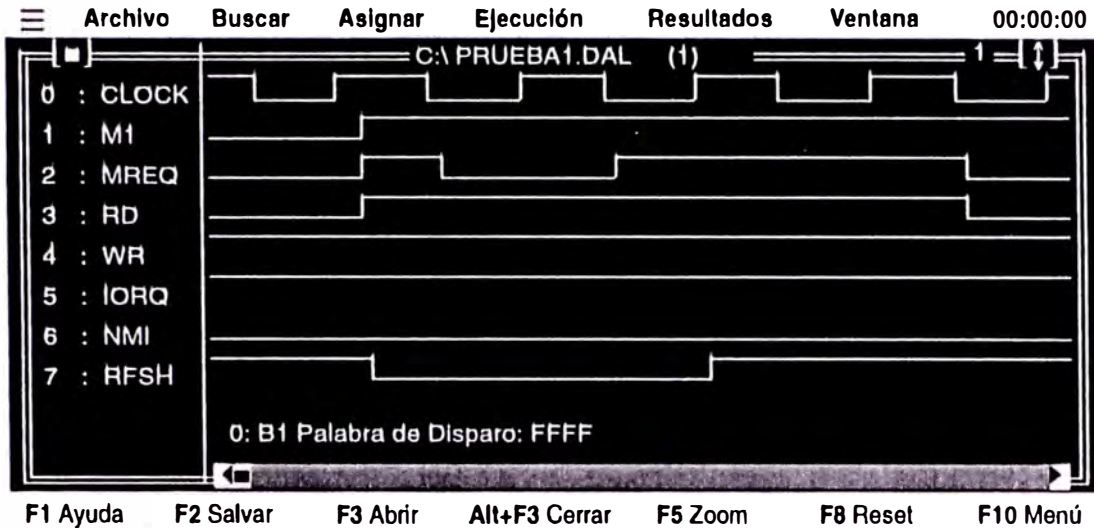


Figura 53. Resultados en formato Normal de la Prueba 1

Allí se observa el diagrama de tiempos que ocurre después que NMI ha ocurrido. Es de observar que el pin NMI se encuentra en nivel bajo y notamos parte de un ciclo fetch en el cual se observa el ciclo de refresco de memoria. Luego se podrá usar los items del menú Resultados para visualizar los datos, por ejemplo en binario, para lo cual se presionará la tecla F9, que produce el cambio en la pantalla como se muestra en la Figura 54.

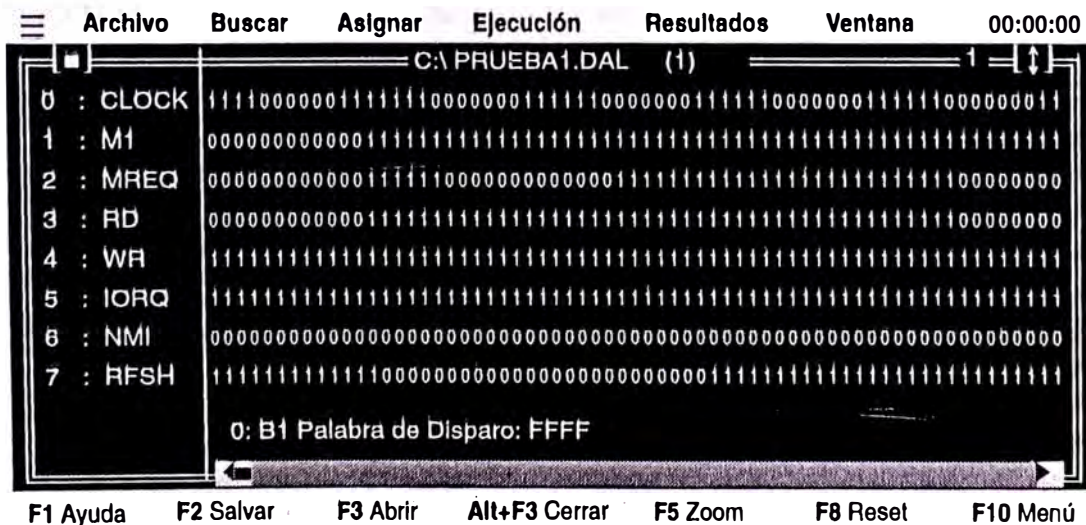


Figura 54. Resultados en formato Binario de la Prueba 1

Con esto se podrá calcular la frecuencia de reloj CLOCK. Observamos que 13 bits forman el periodo y como de bit a bit existe una separación de 50ns entonces el periodo es de 650ns lo que da una frecuencia de 1.54Mhz.

5.7.2.- Prueba 2

Supongamos que deseamos ver el momento en que el pin NMI pasa a nivel bajo, que en el experimento anterior no se logra captar.

Para eso tenemos que disparar al analizador en el Modo Con Pre-Disparo; en el que le especificamos 4 muestras antes previas al disparo, así presionamos **Alt+P** e ingresamos 4, las otras condiciones se mantienen del experimento anterior. Luego solicitamos el Modo Con Pre-Disparo y presionamos en el Microprofessor la tecla **MONI** con lo que se apreciará en la pantalla como se muestra en la Figura 55.

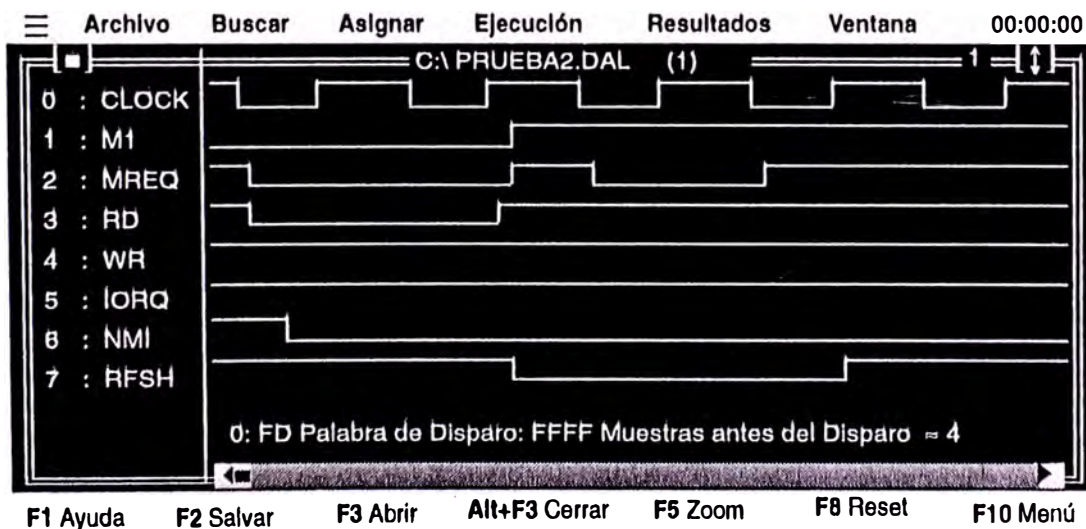


Figura 55. Resultados en formato Normal de la Prueba 2

En este diagrama se nota que el pin NMI ha bajado a nivel bajo justo cuando se realizaba un ciclo fetch.

5.7.2.- Prueba 3

Como tercera prueba vamos a capturar una interrupción enmascarable vía la línea INT. En el Microprofessor, esta línea del Z80-CPU la usan los dispositivos tales como el Z80-CTC, el Z80-PIO o mediante la presión de la tecla INTR; nosotros vamos a hacer el uso del Z80-CTC. Para lo cual vamos a ejecutar un programa que muestre en la pantalla del Microprofessor las horas, minutos y segundos. El programa es el siguiente (para mayor información referenciarse al Manual de Experimentos de Software/Hardware de Multitech Industrial Corp.):

```

                                ORG 1800H
                                SCAN EQU 05FEH
1800:3E18                        LD A,18H
1002:ED47                        LD I,A;
1804:3EB5                        LD A,10110101B
1806:D340                        OUT (40H),A
1808:3E20                        LD A,20H
180A:D340                        OUT (40H),A
180C:3EA8                        LD A,0A8H
180E:D340                        OUT (40H),A
1810:ED5E                        IM 2
1812:FB                          EI
                                MAIN:
1813:DD21041A                    LD IX,1A04H

```

1817:CDFE05	CALL SCAN
181A:18F7	JR MAIN
	SUME:
181C:11001A	LD DE,1A00H
181F:1A	LD A,(DE)
1820:3C	INC A
1821:12	LD (DE),A
1822:FEDA	CP 0DAH
1824:0604	LD B,4
1826:C0	RET NZ
1827:AF	XOR A
1828:05	DEC B
1829:12	LD (DE),A
182A:13	INC DE
182B:215318	LD HL,1853H
182E:1A	UNO: LD A,(DE)
182F:C601	ADD A,1
1831:27	DAA
1832:12	LD (DE),A
1833:96	SUB (HL)
1834:D8	RET C
1835:12	LD (DE),A
1836:23	INC HL
1837:13	INC DE
1838:10F4	DJNZ UNO
183A:C9	RET

SETDISPLAY:

```
183B:21041A      LD HL,1A04H
183E:11011A      LD DE, SEG
1841:0603        LD B,3
1843:1A          DOS: LD A,(DE)
1844:CD7806      CALL HEX7SG
1847:13          INC DE
1848:10F9        DJNZ DOS
184A:2B          DEC HL
184B:2B          DEC HL
184C:CBF6        SET 6,(HL)
184E:2B          DEC HL
184F:2B          DEC HL
1850:CBF6        SET 6,(HL)
1852:C9          RET
;VALORES MAXIMOS:
1853:60          DEFB 60H
1854:60          DEFB 60H
1855:12          DEFB 12H
                  ORG 18A8H
18A8:AA18        DEFW INTERRUPCION
                  INTERRUPCION:
18AA:F5          PUSH AF
18AB:C5          PUSH BC
18AC:D5          PUSH DE
18AD:E5          PUSH HL
18AE:CD1C18      CALL SUME
18B1:78          LD A,B
18B2:FE04        CP 4
```



```

18B4:C43B18      CALL NZ,SETDISPLAY
18B7:E1          POP HL
18B8:D1          POP DE
18B9:C1          POP BC
18BA:F1          POP AF
18BB:FB          EI
18BC:ED4D        RETI
HEX7SG          EQU 678H

                  ORG 1A00H

1A00:00
1A01:00          SEG      DEFB 0
1A02:00          MIN      DEFB 0
1A03:00          HORA     DEFB 0
1A04:0000        DISP     DEFW 0
1A06:0000        DEFW 0
1A08:0000        DEFW 0

                  END

```

En la rutina anterior, el Z80-CTC es programado para interrumpir al microprocesador (el cual se encuentra en un lazo de programa), para actualizar las horas, minutos y segundos.

Para observar en el analizador lógico una petición de interrupción, vamos a instalar las sondas del analizador a los pines del Z80 como sigue:

CH 0: CLOCK (pin 6)

CH 1	:	M1	(pin 27)
CH 2	:	MREQ	(pin 19)
CH 3	:	IORQ	(pin 20)
CH 4	:	INT	(pin 16)
CH 5	:	RD	(pin 21)
CH 6	:	WR	(pin 22)
CH 7	:	RFSH	(pin 28)

Los pines que aparecen a la derecha son referidos al Z80-CPU. En cuanto a las sondas TA0..TA15 estas se instalarán en el bus de direcciones del Z80-CPU, como sigue:

TA0	:	A0	(pin 30)
TA1	:	A1	(pin 31)
TA2	:	A2	(pin 32)
TA3	:	A3	(pin 33)
TA4	:	A4	(pin 34)
TA5	:	A5	(pin 35)
TA6	:	A6	(pin 36)
TA7	:	A7	(pin 37)
TA8	:	A8	(pin 38)
TA9	:	A9	(pin 39)
TA10	:	A10	(pin 40)
TA11	:	A11	(pin 1)
TA12	:	A12	(pin 2)
TA13	:	A13	(pin 3)
TA14	:	A14	(pin 4)
TA15	:	A15	(pin 5)

Los habilitadores E0 y E1 se colocarán en el pin M1 del Z80 (pin 27) para que el analizador sólo compare una dirección en los instantes en que el CPU realiza sólo ciclos fetch. Luego se procederá a ubicar una hoja de adquisición en pantalla y se etiquetará los canales. Como deseamos observar el inicio de la rutina de interrupción, la cual comienza en la dirección 18AAh, entonces debemos colocar la palabra de disparo en el analizador con dicha cantidad, es decir 18AAh y por último vamos a colocar una cantidad 500 muestras de pre-disparo, con el objeto de registrar el ciclo de solicitud y reconocimiento de interrupción.

Luego, se procederá a disparar el analizador, haciendo uso del modo Con Pre-Disparo. Así se procederá a presionar las teclas **Ctrl+Ins** y por último se ejecutará el programa en el Microprofessor produciéndose así el disparo del analizador.

Los datos obtenidos se han imprimido utilizando la opción de impresión de ALOG y son mostrados en el Apéndice B, en la cual se ha realizado la impresión a partir de la muestra 0 hasta la 1500.

Analizando las muestras adquiridas, podemos observar que el manejador de interrupciones comienza a partir de la muestra 500 en donde claramente puede observarse los diagramas de tiempos de las instrucciones PUSH AF, PUSH BC, PUSH DE, PUSH HL etc. Si observamos las muestras de pre-disparo (0-500), notamos que en la muestra 166 el Z80-CTC solicita la interrupción (la línea INT cambia a "0"). A partir de la muestra 254 aparece el diagrama de tiempos del reconocimiento de dicha interrupción.

Después de realizar cada una de estas pruebas, podemos almacenar los datos en disco usando la opción **Salvar (F2)**, para su análisis posterior e impresión.

CONCLUSIONES

- 1) Se ha logrado implementar, y experimentar con él, una tarjeta de interfaz a un computador personal, que ayudado de un software, hace posible que todo el sistema funcione como un analizador lógico de tiempos de bajo coste.

- 2) El equipo cuenta con las siguientes especificaciones técnicas:
 - Frecuencia de muestreo: 20Mhz.
 - Número de canales muestreados: 8
 - Todas sus entradas son compatibles directamente con TTL.
 - Profundidad de memoria por canal: 8Kbits.
 - Tres modos de disparo: Manual, Automático y Con Pre-Disparo.
 - Cantidad de muestras de Pre-Disparo es programable.
 - Palabra de disparo de 16 bits.
 - Software de comunicación con el usuario hecho a base de menús y ventanas de diálogo.
 - Almacenamiento de los datos obtenidos hacia disco como archivos de pequeña dimensión (9296 bytes) para su posterior visualización.
 - Crea archivo de respaldo.
 - Capacidad de poder imprimir los datos obtenidos.

- 3) Este instrumento ha sido desarrollado en base a componentes electrónicos netamente comerciales. Esto lo hace completamente permisible de implementar para ser de utilidad en diversas aplicaciones como son: el diseño circuitos digitales o de sistemas electrónicos basados en microprocesadores.

- 4) También este módulo puede ser de utilidad para fines académicos, ya que se podrá comprender mejor el comportamiento de los microprocesadores en lo que respecta a su dinámica de tiempos, así como cualquier sistema digital en el que se requiera observar y estudiar sus diagramas de tiempos.
- 5) En cuanto al software debemos mencionar que es completamente amigable, el cual puede ejecutarse desde el prompt del DOS; no obstante es compatible con Windows, es decir podrá ejecutarse sin ningún problema bajo la plataforma Windows. El programa es basado totalmente en menús lo que entrega bastante facilidad en su manejo por parte del operador. Adicionalmente cuenta con un archivo de ayuda que aparece como una ventana emergente, entregando información del evento actual que esté sucediendo en el programa. Además cuenta con ciertas rutinas de utilidad como cambiar el drive actual o como salir al DOS temporalmente etc.
- 6) La manera de adquirir los datos e interactuar con el computador podría mantenerse, y en base a esto se le proporcionaría mejoras al presente proyecto, en lo que respecta a sus características. Así debemos mencionar que se podría utilizar memorias más rápidas cuyo tiempo de acceso sea menor a la de la memorias empleadas en el presente proyecto. Esto obligaría a que también todos los circuitos integrados debieran ser TTL de la serie Fast, con lo que se conseguiría una mayor frecuencia de muestreo, aumentando así su performance. También se podría hacer un aumento en el número de canales los que podrían llegar a sumar 16 ó 32 canales e incluso 64.
- 7) Al realizarse cambios en la tarjeta de Interfaz, también el software tendría que ser mejorado para poder atender a las nuevas características del hardware e inclusive se podría rediseñar el software para que sea un aplicativo en entorno Windows, es decir migrar al ambiente gráfico.
- 8) Debemos de indicar que el presente proyecto ha sido probado en diversas computadoras personales de diversas velocidades tales como AT 286/25Mhz, AT 386/40Mhz y AT 486/66Mhz, sin ningún reporte de error de hardware ni de software.

BIBLIOGRAFIA

- 1) Stephen O'Brien, *Turbo Pascal 6 Manual de Referencia*, Osborne/McGraw-Hill, 1991.
- 2) Robert Jourdain, *Solucionario del Programador para IBM PC, XT, AT y compatibles*, Anaya Multimedia, 1986.
- 3) Lewis C. Eggebrecht. *Interfacing to the IBM Personal Computer*, SAMS, 1990.
- 4) Thomson-CSF Sescosem, *Logic TTL Integrated Circuits LS series*, 1976.
- 5) RCA Corporation, *CMOS Microprocessors, Memories and Peripherals*, 1984.
- 6) Philips Components, *Fast TTL Logic Series Data Hand Book*, 1990.
- 7) Multitech Industrial Corp., *Microprofessor MPF-I, User's and Experiment Manual*, 1981.
- 8) Intel, *80386 Manual de Referencia Hardware*, Anaya Multimedia.
- 9) Michael Riezenman, *Electronic Design*, November 2, 1982, pp 91-96.
- 10) Bob Milne, *Electronic Design*, September 15, 1983, pp 93-108.
- 11) Dick Woods, *Electronic Design*, September 15, 1983, pp 147-152.
- 12) William Swan, *Electronic Design*, May 3, 1984, pp 341-348.
- 13) Herman van Eijkelenburg, *Electronic Design*, August 15, 1985, pp 139-147.

APENDICE A

COSTOS

COSTOS

En esta parte sólo se indicará los precios de los chips utilizados en el presente proyecto más no la de una tarjeta impresa lista para ser conectada en el computador. Tampoco se indicará el costo del software.

La tarjeta prototipo esta basada en los siguientes circuitos integrados en la cual se especifica la cantidad, así como su precio en nuevos soles.

Circuito Integrado	Cantidad	P. Unitario	Total
TTL 74F02	1	1.50	1.50
TTL 74S04	1	1.5	1.50
TTL 74F08	2	1.50	3.00
TTL 74F20	1	1.70	1.70
TTL 74F32	1	1.50	1.50
TTL 74S74	1	1.30	1.30
TTL 74LS75	1	0.95	0.95
TTL 74LS85	3	1.80	5.40
TTL 74LS95	8	5.30	42.40
TTL 74F112	2	2.00	2.00
TTL 74LS138	1	1.70	1.70
TTL 74F139	1	1.50	1.50
TTL 74LS193	3	2.50	7.50
TTL 74LS244	4	1.50	6.00
TTL 74LS245	1	1.50	1.50

TTL 74LS273	4	1.50	6.00
TTL 74LS283	3	1.70	5.10
TTL 74F373	1	1.70	1.70
TTL 74F374	4	1.70	6.80
TTL 74F521	2	3.00	6.00
CMOS TMM2016	4	5.00	20.00
			125.05
		Total:	S/. 125.05

Debemos mencionar que los precios señalados aquí podrían ser modificados con el transcurrir del tiempo. Sin embargo, observando el precio total de los circuitos integrados, podríamos conjeturar que el costo final de la tarjeta de interfaz debe ser sumamente módica.

APENDICE B

LISTADO DEL PROGRAMA FUENTE PARA EL COMPUTADOR PERSONAL

```

{.....}
{
{ Analizador Lógico 1.0
{ ALOG.EXE
{ Copyright (c) 1995 por José Manuel Rivera Vigo
{ Todos los Derechos Reservados
{
{.....}

```

Program LOGIC;

{\$X+,S-}

{\$M 16384,8192,655360}

uses

Dos, Objects, Drivers, Memory, Views, Menus, Dialogs, StdDlg, MsgBox, App,
 Crt, Gadgets, Calendar, Calc, HelpFile, Hcxxxx, Analyzer, AlogCmds;

type

{ TLOGIC }

PLOGIC = ^TLOGIC;

TLOGIC = object(TApplication)

Clock: PClockView;

constructor Init;

procedure GetEvent(var Event: TEvent); virtual;

function GetPalette: PPalette; virtual;

procedure HandleEvent(var Event: TEvent); virtual;

procedure Idle; virtual;

procedure InitMenuBar; virtual;

procedure InitStatusLine; virtual;

procedure OutOfMemory; virtual;

procedure About;

destructor Done; virtual;

end;

function ExecDialog(P: PDialog; Help: Word; Data: Pointer): Word;

var

Result: Word;

begin

Result := cmCancel;

P^.HelpCtx:= Help;

P := PDialog(Application^.ValidView(P));

if P <> nil then

begin

if Data <> nil then P^.SetData(Data^);

Result := DeskTop^.ExecView(P);

if (Result <> cmCancel) and (Data <> nil) then P^.GetData(Data^);

Dispose(P, Done);

end;

ExecDialog := Result;

end;

function CrearDialCambiar: PDialog;

var

D : PDialog;

R : TRect;

C : PView;

begin

R.Assign(0,0,35,17);

D := New(PDialog, Init(R, 'Canales'));

With D^ Do

Begin

Options := Options or ofcentered;

```
R.Assign(5,2,28,3);
Insert(New(PStaticText, Init(R, 'Ingrese los nombres')));
R.Assign(5,3,28,4);
Insert(New(PStaticText, Init(R, 'de los canales :')));
```

```
R.Assign(15,5,24,6);
C := New(PInputLine,Init(R, 7));
Insert(C);
R.Assign(4,5,15,6);
Insert(New(PLabel, Init(R, 'CANAL 0 :', C)));
R.Assign(24, 5, 27, 6);
Insert(New(PHistory, Init(R, PInputLine(C), 1)));
```

```
R.Assign(15,6,24,7);
C := New(PInputLine,Init(R, 7));
Insert(C);
R.Assign(4,6,15,7);
Insert(New(PLabel, Init(R, 'CANAL 1 :', C)));
R.Assign(24, 6, 27, 7);
Insert(New(PHistory, Init(R, PInputLine(C), 1)));
```

```
R.Assign(15,7,24,8);
C := New(PInputLine,Init(R, 7));
Insert(C);
R.Assign(4,7,15,8);
Insert(New(PLabel, Init(R, 'CANAL 2 :', C)));
R.Assign(24, 7, 27, 8);
Insert(New(PHistory, Init(R, PInputLine(C), 1)));
```

```
R.Assign(15,8,24,9);
C := New(PInputLine,Init(R, 7));
Insert(C);
R.Assign(4,8,15,9);
Insert(New(PLabel, Init(R, 'CANAL 3 :', C)));
R.Assign(24, 8, 27, 9);
Insert(New(PHistory, Init(R, PInputLine(C), 1)));
```

```
R.Assign(15,9,24,10);
C := New(PInputLine,Init(R, 7));
Insert(C);
R.Assign(4,9,15,10);
Insert(New(PLabel, Init(R, 'CANAL 4 :', C)));
R.Assign(24, 9, 27, 10);
Insert(New(PHistory, Init(R, PInputLine(C), 1)));
```

```
R.Assign(15,10,24,11);
C := New(PInputLine,Init(R, 7));
Insert(C);
R.Assign(4,10,15,11);
Insert(New(PLabel, Init(R, 'CANAL 5 :', C)));
R.Assign(24, 10, 27, 11);
Insert(New(PHistory, Init(R, PInputLine(C), 1)));
```

```
R.Assign(15,11,24,12);
C := New(PInputLine,Init(R, 7));
Insert(C);
R.Assign(4,11,15,12);
Insert(New(PLabel, Init(R, 'CANAL 6 :', C)));
R.Assign(24, 11, 27, 12);
Insert(New(PHistory, Init(R, PInputLine(C), 1)));
```

```
R.Assign(15,12,24,13);
C := New(PInputLine,Init(R, 7));
Insert(C);
R.Assign(4,12,15,13);
Insert(New(PLabel, Init(R, 'CANAL 7 :', C)));
```

```
R.Assign(24, 12, 27, 13);
Insert(New(PHistory, Init(R, PInputLine(C), 1)));
```

```
R.Assign(5, 14, 15, 16);
Insert(New(PButton, Init(R, 'O-K', cmOK, bfDefault)));
R.Assign(18, 14, 30, 16);
Insert(New(PButton, Init(R, 'Cancelar', cmCancel, bfNormal)));
```

```
end;
CrearDialCambiar := D;
```

```
end;
```

```
function CrearDialIntercambio: PDialog;
```

```
var
```

```
  D: PDialog;
  Control: PView;
  R: TRect;
```

```
begin
```

```
  R.Assign(0, 17, 45, 22);
  D := New(PDialog, Init(R, 'Intercambiar'));
  with D^ do
```

```
    begin
      Options := Options or ofCenterx;
```

```

      R.Assign(17, 1, 21, 2);
      Control := New(PInputLine, Init(R, 2));
      Insert(Control);
      R.Assign(1, 1, 16, 2);
      Insert(New(PLabel, Init(R, '~P~rimer Canal:', Control)));
```

```

      R.Assign(17, 3, 21, 4);
      Control := New(PInputLine, Init(R, 2));
      Insert(Control);
      R.Assign(1, 3, 16, 4);
      Insert(New(PLabel, Init(R, '~S~egundo Canal:', Control)));
```

```

      R.Assign(22, 2, 30, 4);
      Insert(New(PButton, Init(R, 'O-K~', cmOk, bfDefault)));
      R.Assign(31, 2, 43, 4);
      Insert(New(PButton, Init(R, 'Cancelar', cmCancel, bfNormal)));
```

```
    end;
  CrearDialIntercambio := D;
```

```
end;
```

```
function CrearDialBuscar: PDialog;
```

```
var
```

```
  D: PDialog;
  Control: PView;
  R: TRect;
```

```
begin
```

```
  R.Assign(0, 0, 38, 12);
  D := New(PDialog, Init(R, 'Buscar'));
  with D^ do
```

```
    begin
      Options := Options or ofCentered;
```

```

      R.Assign(3, 3, 32, 4);
      Control := New(PInputLine, Init(R, 8));
      Insert(Control);
      R.Assign(2, 2, 24, 3);
      Insert(New(PLabel, Init(R, '~B~inario a buscar:', Control)));
      R.Assign(32, 3, 35, 4);
      Insert(New(PHistory, Init(R, PInputLine(Control), 10)));
```

```

R.Assign(3, 5, 35, 7);
Insert(New(PRadioButtons, Init(R,
  NewSItem('-D~esde el inicio',
  NewSItem('-D~esde el dato actual', nil))));

R.Assign(11, 9, 21, 11);
Insert(New(PButton, Init(R, 'O~K~', cmOk, bfDefault)));
Inc(R.A.X, 10); Inc(R.B.X, 12);
Insert(New(PButton, Init(R, 'Cancelar', cmCancel, bfNormal)));

end;
CrearDialBuscar := D;
end;

function CrearDiallra: PDialog;
var
  D: PDialog;
  Control: PView;
  R: TRect;
begin
  R.Assign(0, 0, 45, 8);
  D := New(PDialog, Init(R, 'Ir a la muestra ?'));
  with D^ do
  begin
    Options := Options or ofCentered;

    R.Assign(1, 2, 25, 3);
    Insert(New(PLabel, Init(R, '-I~ngrese nueva posición:', Control)));

    R.Assign(3, 3, 25, 4);
    Control := New(PInputLine, Init(R, 4));
    Insert(Control);

    R.Assign(26, 3, 29, 4);
    Insert(New(PHistory, Init(R, PInputLine(Control), 11)));

    R.Assign(31, 1, 43, 3);
    Insert(New(PButton, Init(R, 'O~K~', cmOk, bfDefault)));

    R.Assign(31, 3, 43, 5);
    Insert(New(PButton, Init(R, 'Cancelar', cmCancel, bfNormal)));
  end;
  CrearDiallra := D;
end;

function CrearDialFuncion: PDialog;
var
  D : PDialog;
  Control: PView;
  R: TRect;

begin
  R.Assign(0,0,35,21);
  D := New(PDialog, Init(R, 'Función Booleana'));
  With D^ Do
  Begin
    Options := Options or ofcentered;
    R.Assign(5,1,33,2);
    Insert(New(PStaticText, Init(R,'Formato: A_OPERACION B')));
    R.Assign(5,3,28,4);
    R.Assign(13, 2, 16, 3);
    Control := New(PInputLine, Init(R, 2));
    Insert(Control);
  end;
end;

```

```

R.Assign(1, 2, 10, 3);
Insert(New(PLabel, Init(R, 'Canal -A-:', Control)));
R.Assign(28, 2, 31, 3);
Control := New(PInputLine, Init(R, 2));
Insert(Control);
R.Assign(17, 2, 26, 3);
Insert(New(PLabel, Init(R, 'Canal -B-:', Control)));
R.Assign(10, 4, 23, 5);
Insert(New(PCheckBoxes, Init(R, NewSItem('~V-visible', nil))));
R.Assign(5,6,33,7);
Insert(New(PStaticText, Init(R, 'Seleccione Operación:')));
R.Assign(5, 8, 23, 17);
  Insert(New(PRadioButtons, Init(R,
    NewSItem('~A-ND',
    NewSItem('~O-R',
    NewSItem('~E-XOR',
    NewSItem('~N-OT',
    NewSItem('~NA-ND',
    NewSItem('~NO-R',
    NewSItem('~NOR-EX',
    NewSItem('~I-NHIBICION',
    NewSItem('~IM-PLICACION', nil))))))))));
R.Assign(5,18,15,20);
Insert(New(PButton, Init(R, 'O-K', cmOK, bfDefault)));
R.Assign(18,18,30,20);
Insert(New(PButton, Init(R, 'Cancelar', cmCancel, bfNormal)));
end;
CrearDialFuncion := D;

```

end;

function CrearDialPalabra: PDialog;

var

```

D : PDialog;
R : TRect;
P : PView;

```

begin

```

R.Assign(0,0,39,14);
D:= New(PDialog, Init(R, 'Palabra de Disparo'));
With D^ Do
  Begin
    Options := Options or ofcentered;
    R.Assign(5,2,31,3);
    Insert(New(PStaticText, Init(R, 'Ingrese la Palabra de')));
    R.Assign(5,3,31,4);
    Insert(New(PStaticText, Init(R, 'Disparo en HEXADECIMAL')));
    R.Assign(12,7,18,8);
    P := New(PInputLine, Init(R, 4));
    Insert(P);
    R.Assign(4,5,19,6);
    Insert(New(PLabel, Init(R, 'Palabra HEX :', P)));
    R.Assign(5,11,15,13);
    Insert(New(PButton, Init(R, 'O-K', cmOK, bfDefault)));
    R.Assign(18,11,30,13);
    Insert(New(PButton, Init(R, 'Cancelar', cmCancel, bfNormal)));
  end;

```

CrearDialPalabra:= D;

end;

function CrearDialPretrigger: PDialog;

```

var
  D : PDialog;
  R : TRect;
  P : PView;

begin
  R.Assign(0,0,39,12);
  D:= New(PDialog, Init(R, 'Pre-Disparo'));
  With D^ Do
    Begin
      Options := Options or ofcentered;
      R.Assign(5,2,33,3);
      Insert(New(PStaticText, Init(R, 'Ingrese la Cantidad de')));
      R.Assign(5,3,33,4);
      Insert(New(PStaticText, Init(R, 'Muestras ANTES del Disparo')));
      R.Assign(26,5,32,6);
      P := New(PInputLine,Init(R, 4));
      Insert(P);
      R.Assign(4,5,25,6);
      Insert(New(PLabel, Init(R, 'Número de Muestras:', P)));
      R.Assign(5,8,15,10);
      Insert(New(PButton,Init(R, 'O~K', cmOK, bfDefault)));
      R.Assign(18,8,30,10);
      Insert(New(PButton,Init(R,'Cancelar',cmCancel, bfNormal)));
    end;
  CrearDialPreTrigger:= D;
end;

```

```
function CrearDialImpresion: PDialog;
```

```

var
  D: PDialog;
  Control: PView;
  R: TRect;
begin
  R.Assign(0, 17, 50, 24);
  D := New(PDialog, Init(R, 'Impresión'));
  with D^ do
    begin
      Options := Options or ofCentered;

      R.Assign(18, 2, 24, 3);
      Control := New(PInputLine, Init(R, 4));
      Insert(Control);
      R.Assign(1, 2, 17, 3);
      Insert(New(PLabel, Init(R, 'Dato de ~I~nicio:', Control)));

      R.Assign(18, 4, 24, 5);
      Control := New(PInputLine, Init(R, 4));
      Insert(Control);
      R.Assign(1, 4, 17, 5);
      Insert(New(PLabel, Init(R, 'Dato ~F~inal:', Control)));

      R.Assign(24, 3, 32, 5);
      Insert(New(PButton, Init(R, 'O~K~', cmOk, bfDefault)));
      R.Assign(34, 3, 48, 5);
      Insert(New(PButton, Init(R, 'Cancelar', cmCancel, bfNormal)));
    end;
  CrearDialImpresion:= D;
end;

```

```
function HacerDialogo(Dialog: Integer; Info: pointer): word; far;
```

```

begin
  case Dialog of

```



```

edReadError:
HacerDialogo := MessageBox('Error leyendo archivo %s.',@Info, mfError + mfOkButton);

edWriteError
HacerDialogo := MessageBox('Error escribiendo archivo %s.',@Info, mfError + mfOkButton);

edCreateError :
HacerDialogo := MessageBox('Error creando archivo %s.',@Info, mfError + mfOkButton);

edSaveModify :
HacerDialogo := MessageBox('^C'%s ha sido modificado. Salvarlo?',@Info, mflInformation + mfYesNoCancel);

edSaveUntitled:
HacerDialogo := MessageBox('^C'Salvar archivo sin ttulo?',nil, mflInformation + mfYesNoCancel);

edCambEtiquet :
HacerDialogo := ExecDialog(CrearDialCambiar, hcNomCanal, Info);

edIntercambio :
HacerDialogo := ExecDialog(CrearDialIntercambio, hcIntercambio, Info);

edBuscar
HacerDialogo := ExecDialog(CrearDialBuscar, hcEncontrar,Info);

edlra
HacerDialogo := ExecDialog(CrearDiallra, hcLra,Info);

edFuncion
HacerDialogo := ExecDialog(CrearDialFuncion, hcFuncion, Info);

edPalabra
HacerDialogo := ExecDialog(CrearDialPalabra, hcPalDis,Info);

edPreTrigger
HacerDialogo := ExecDialog(CrearDialPreTrigger, hcPreDisparo,Info);

edPrinter
HacerDialogo := ExecDialog(CrearDialImpresion, hcImpresion, Info);

edSaveAs
HacerDialogo := ExecDialog(New(PFileDialog, Init('*.DAL',
'Salvar Archivo Como',~N~ombre', fdOkButton,101)),
hcSalvarComoA,Info);

end;
end;

```

```
var
```

```

AnaLog: TLOGIC;
VectorAsinc : Pointer;

```

```
{ CalcHelpName }
```

```
function CalcHelpName: PathStr;
```

```
var
```

```

EXEName: PathStr;
Dir: DirStr;
Name: NameStr;
Ext: ExtStr;

```

```
begin
```

```

if Lo(DosVersion) >= 3 then EXEName := ParamStr(0)
else EXEName := FSearch('ALOG.EXE', GetEnv('PATH'));

```

```

FSplit(EXEName, Dir, Name, Ext);
if Dir[Length(Dir)] = '\' then Dec(Dir[0]);
CalcHelpName := FSearch('ALOG.HLP', Dir);
end;

```

```

function OpenEdicion(FileName: FNameStr): PVentanaPrin;
var

```

```

    P: PView;
    R: TRect;
begin
    Inc(WinCount);
    DeskTop^.GetExtent(R);
    P := Application^.ValidView(New(PVentanaPrin,
        Init(R, FileName, WinCount)));
    DeskTop^.Insert(P);
    OpenEdicion := PVentanaPrin(P);
end;

```

```
{ TLOGIC }
```

```
constructor TLOGIC.Init;
```

```
var
    R: TRect;
    I: Integer;
```

```
begin
```

```

    TApplication.Init;
    DialogPant := HacerDialogo;
    RegisterObjects;
    RegisterViews;
    RegisterMenus;
    RegisterDialogs;
    RegisterApp;
    RegisterHelpFile;
    RegisterCalendar;
    RegisterCalc;
    GetIntVec($0A, VectorAsinc);
    SetIntVec($0A, @ManejaInt);
    GetExtent(R);
    R.A.X := R.B.X - 9; R.B.Y := R.A.Y + 1;
    Clock := New(PClockView, Init(R));
    Insert(Clock);
    DisableCommands([cmSave, cmSaveAs, cmImpresion, cmPalabra, cmPreTrigger, cmReset,
cmCanales, cmConPre,
cmManual, cmAutoSin, cmMuestra, cmCambEtiquet, cmIntercambio, cmNormal, cmBinario, cmMovimiento,
cmBuscar, cmBuscarAgain, cmIra, cmFuncion, cmOrden]);
    About;
    Port[$2F1] := $20; (* 00100000 *)
    Port[$2F1] := $A0; (* 10100000 *)
    Delay(1);

```

```
if not Listo then
```

```

    begin
        MessageBox('^C'No existe comunicaci3n con la'#13 +
^C#13 +
^C'tarjeta de interfaz', nil, mfError + mfOkButton);
        Done;
        Halt(1);
    end;

```

```
end;
```

```

procedure TLOGIC.GetEvent(var Event: TEvent);
var
  W: PWindow;
  HFile: PHelpFile;
  HelpStrm: PDosStream;
const
  HelpInUse: Boolean = False;
begin
  TApplication.GetEvent(Event);
  case Event.What of
    evCommand:
      if (Event.Command = cmHelp) and not HelpInUse then
        begin
          HelpInUse := True;
          HelpStrm := New(PDosStream, Init(CalcHelpName, stOpenRead));
          HFile := New(PHelpFile, Init(HelpStrm));
          if HelpStrm^.Status <> stOk then
            begin
              MessageBox('No puedo abrir archivo de ayuda.'
                , nil, mfError + mfOkButton);
              Dispose(HFile, Done);
            end
          else
            begin
              W := New(PHelpWindow, Init(HFile, GetHelpCtx));
              if ValidView(W) <> nil then
                begin
                  ExecView(W);
                  Dispose(W, Done);
                end;
              ClearEvent(Event);
            end;
          HelpInUse := False;
        end;
    evMouseDown:
      if Event.Buttons <> 1 then Event.What := evNothing;
  end;
end;

function TLOGIC.GetPalette: PPalette;
const
  CNewColor = CColor + CHelpColor;
  CNewBlackWhite = CBlackWhite + CHelpBlackWhite;
  CNewMonochrome = CMonochrome + CHelpMonochrome;
  P: array[apColor..apMonochrome] of string[Length(CNewColor)] =
    (CNewColor, CNewBlackWhite, CNewMonochrome);
begin
  GetPalette := @P[AppPalette];
end;

procedure TLOGIC.HandleEvent(var Event: TEvent);

procedure FileOpen;
var
  FileName: FNameStr;
begin
  FileName := '*.DAL';
  if ExecDialog(New(PFileDialog, Init(*, *, 'Abrir un Archivo',
    '~N~ombre', fdOpenButton + fdHelpButton, 100)), hcFOpen, @FileName) <> cmCancel
  then OpenEdicion(FileName);
end;

procedure FileNew;
begin
  OpenEdicion('');
end;

```

```

procedure ChangeDir;
var
  D: PChDirDialog;
begin
  D := New(PChDirDialog, Init(cdNormal + cdHelpButton, 101));
  D^.HelpCtx := hcFCChDirDBox;
  if ValidView(D) <> nil then
  begin
    DeskTop^.ExecView(D);
    Dispose(D, Done);
  end;
end;

```

```

procedure Tile;
var
  R: TRect;
begin
  Desktop^.GetExtent(R);
  Desktop^.Tile(R);
end;

```

```

procedure Cascade;
var
  R: TRect;
begin
  Desktop^.GetExtent(R);
  Desktop^.Cascade(R);
end;

```

```

procedure Calendar;
var
  P: PCalendarWindow;
begin
  P := New(PCalendarWindow, Init);
  P^.HelpCtx := hcCalendar;
  Desktop^.Insert(ValidView(P));
end;

```

```

procedure Calculator;
var
  P: PCalculator;
begin
  P := New(PCalculator, Init);
  P^.HelpCtx := hcCalculator;
  if ValidView(P) <> nil then
    Desktop^.Insert(P);
end;

```

```

procedure DosShell;
begin
  DoneSysError;
  DoneEvents;
  DoneVideo;
  DoneMemory;
  SetMemTop(HeapPtr);
  PrintStr('Escriba EXIT para retornar...');
  SwapVectors;
  Exec(GetEnv('COMSPEC'), "");
  SwapVectors;
  SetMemTop(HeapEnd);
  InitMemory;
  InitVideo;
  InitEvents;
  InitSysError;
  Redraw;
end;

```

begin

```
TApplication.HandleEvent(Event);
case Event.What of
  evCommand:
    begin
      case Event.Command of
        cmFOpen      : FileOpen;
        cmChDir      : ChangeDir;
        cmCascade    : Cascade;
        cmNew        : FileNew;
        cmTile       : Tile;
        cmAbout      : About;
        cmCalendar   : Calendar;
        cmCalculator  : Calculator;
        cmDosShell   : DosShell;
      else
        Exit;
      end;
      ClearEvent(Event);
    end;
end;
end;
```

procedure TLOGIC.Idle;

```
function IsTileable(P: PView): Boolean; far;
begin
  IsTileable := P^.Options and ofTileable <> 0;
end;
```

begin

```
TApplication.Idle;
Clock^.Update;
if Desktop^.FirstThat(@ IsTileable) <> nil then
  EnableCommands([cmTile, cmCascade])
else
  DisableCommands([cmTile, cmCascade]);
```

end;

procedure TLOGIC.InitMenuBar;

```
var
  R: TRect;
begin
  GetExtent(R);
  R.B.Y := R.A.Y+1;
  MenuBar := New(PMenuBar, Init(R, NewMenu(
    NewSubMenu('~#240~', hcSystem, NewMenu(
      NewItem('~A~cerca', '', kbNoKey, cmAbout, hcSAbout,
        NewLine(
          NewItem('~Ca~l~endario', '', kbNoKey, cmCalendar, hcSCalendar,
            NewItem('~C~alculadora', 'Alt+C', kbAltC, cmCalculator, hcSCalculator, nil)))))),
    NewSubMenu('~A~rchivo', hcFile, NewMenu(
      NewItem('~A~brir...', 'F3', kbF3, cmFOpen, hcFOpen,
        NewItem('~N~uevo...', 'F4', kbF4, cmNew, hcNuevoA,
          NewItem('~S~alvar', 'F2', kbF2, cmSave, hcSalvarA,
            NewItem('~Salvar ~C~omo...', 'Alt+F2', kbAltF2, cmSaveAs, hcSalvarComoA,
              NewItem('~l~mpimir...', '', kbNoKey, cmImpresion, hcImpresion,
                NewItem('~l~ a Directorio', '', kbNoKey, cmChDir, hcFChangeDir,
                  NewLine(
                    NewItem('~Ingresar al ~D~OS', 'Alt+D', kbAltD, cmDosShell, hcFDosShell,
                      NewItem('~T~erminar Sesión', 'Alt+X', kbAltX, cmQuit, hcFExit, nil)))))))))),
    NewSubMenu('~B~uscar', hcBusqueda, NewMenu(
      NewItem('~B~uscar...', 'Ctrl+F7', kbCtrlF7, cmBuscar, hcEncontrar,
```

```

NewlItem('Buscar ~o~tra vez...', 'F7', kbF7, cmBuscarAgain, hcOtraVez,
NewLine(
NewlItem('~l~r a la muestra ?','Alt+l',kbAltI, cmlra, hclra,nil))),
NewSubMenu('A~s~ignar', hcAsignar, NewMenu(
NewlItem('~P~alabra de Disparo', 'Alt+W', kbAltW, cmPalabra, hcPaldis,
NewlItem('Pre~D~isparo', 'Alt+P', kbAltP, cmPretrigger, hcpredisparo,
NewLine(
NewlItem('...nombre a cada ~C~anal','Alt+N',kbAltN, cmCambEtiquet, hcNomcanal,nil))),
NewSubMenu('~E~jecución', hcEjecucion, NewMenu(
NewlItem('~C~on Pre-Disparo','Ctrl+Ins', kbCtrlIns, cmConpre, hcConpre,
NewLine(
NewSubMenu('~S~in Pre-Disparo',hcSinpre, NewMenu(
NewlItem('~M~anual', 'Ins',kblns, cmManual, hcManual,
NewlItem('~A~utomático', 'Shift+Ins',kbShiftIns, cmAutoSin, hcAutomatico, nil))),
NewLine(
NewlItem('~R~eset','F8', kbF8, cmReset, hcReset,nil))))),
NewSubMenu('~R~esultados', hcResultados, NewMenu(
NewlItem('~I~ntercambio','Tab',kbTab, cmlIntercambio, hclIntercambio,
NewlItem('~O~rden Original','Alt+O',kbAltO, cmOrden, hcOrden,
NewLine(
NewSubMenu('Modo ~D~isplay...',hcModoDisplay, NewMenu(
NewlItem('~N~ormal','Shift+F9',kbShiftF9, cmNormal, hcNormal,
NewlItem('~B~inario', 'F9',kbF9, cmBinario, hcBinario, nil))),
NewlItem('~F~unción Booleana','Alt+F',kbAltF, cmFuncion, hcFuncion,
NewlItem('~M~ovimiento', 'Alt+M',kbAltM, cmMovimiento, hcMovimiento,nil)))))),
NewSubMenu('~V~entana', hcWindows, NewMenu(
NewlItem('~R~edimensionar/Mover','Ctrl+F5', kbCtrlF5, cmResize, hcWSizeMove,
NewlItem('~Z~oom', 'F5', kbF5, cmZoom, hcWZoom,
NewlItem('~P~róximo', 'F6', kbF6, cmNext, hcWNext,
NewlItem('~A~nterior', 'Shift+F6', kbShiftF6, cmPrev, hcWPrevious,
NewlItem('~C~errar', 'Alt+F3', kbAltF3, cmClose, hcWCclose,
NewlItem('~M~osaico', 'Alt+T', kbAltT, cmTile, hcWTile,
NewlItem('C~a~scada', 'Alt+K', kbAltK, cmCascade, hcWCascade, nil)))))),
nil))))))));
end;

```

```

procedure TLOGIC.InitStatusLine;
var
R: TRect;
begin
GetExtent(R);
R.A.Y := R.B.Y - 1;
StatusLine := New(PStatusLine, Init(R,
NewStatusDef(0, $FFFF,
NewStatusKey('~F1~ Ayuda', kbF1, cmHelp,
NewStatusKey('~F2~ Salvar', kbF2, cmSave,
NewStatusKey('~F3~ Abrir', kbF3, cmFOpen,
NewStatusKey('~Alt+F3~ Cerrar', kbAltF3, cmClose,
NewStatusKey('~F5~ Zoom', kbF5, cmZoom,
NewStatusKey('~F8~ Reset', kbF8, cmReset,
NewStatusKey('~F10~ Menú', kbF10, cmMenu,
NewStatusKey("", kbCtrlF5, cmResize, nil)))))), nil));
end;

```

```

procedure TLOGIC.OutOfMemory;
begin
MessageBox('NO existe memoria disponible para completar la operación.',
nil, mfError + mfOkButton);
end;

```

```

procedure TLOGIC.About;
var
D: PDialog;
Control: PView;
R: TRect;
begin
R.Assign(0, 0, 40, 13);

```

```

D := New(PDialog, Init(R, 'Acerca'));
D^.HelpCtx := hcSAbout;
with D^ do
begin
  Options := Options or ofCentered;

  R.Grow(-1, -1);
  Dec(R.B.Y, 3);
  Insert(New(PStaticText, Init(R,
    #13 +
    ^C'Analizador Lógico de Tiempos'#13 +
    #13 +
    ^C'Versión 1.0'#13 +
    #13 +
    #13 +
    ^C'por J. Manuel Rivera Vigo')));

  R.Assign(15, 10, 25, 12);
  Insert(New(PButton, Init(R, 'O-K', cmOk, bfDefault)));
end;
if ValidView(D) <> nil then
begin
  DisableCommands([cmFOpen, cmMenu]);
  Desktop^.ExecView(D);
  Dispose(D, Done);
  EnableCommands([cmFOpen, cmMenu]);
end;
end;

destructor TLOGIC.Done;

begin
TApplication.Done;
SetIntVec($0A, VectorAsinc);
end;

begin
AnaLog.Init;
AnaLog.Run;
AnaLog.Done;
end.

```

```

{.....}
{
{   Unidad Analizador Lógico   }
{                               }
{   Copyright (c) 1995 por José Rivera Vigo }
{   Todos los Derechos reservados }
{                               }
{.....}

```

Unit Analyzer;

{ \$F+, I-, O+, S-, V-, X+, D- }

interface

uses Dos, Crt, Printer, Memory, Objects, Drivers, Views,
 Apoyo, MsgBox, AlogCmds, Hcxxxx;

const

orden: array[0..7] of Byte = (0, 1, 2, 3, 4, 5, 6, 7);

```
counter: integer=0;
```

```
const
```

```
edCambEtiquet    = 0;  
edIntercambio    = 1;  
edBuscar         = 2;  
edIra            = 3;  
edFuncion        = 4;  
edReseteo        = 5;  
edPalabra        = 6;  
edPreTrigger     = 7;  
edOutOfMemory    = 8;  
edReadError      = 9;  
edSaveAs         = 10;  
edCreateError    = 11;  
edWriteError     = 12;  
edSaveUntitled   = 13;  
edSaveModify     = 14;  
edPrinter        = 15;
```

```
const
```

```
Control          : byte    = $20;  
Listo            : boolean = false;  
WinCount         : integer = 0;  
Nmuestralni     : integer = 0;
```

```
Clave0          : boolean = false;  
Clave1          : boolean = false;
```

```
Fb              : string  = 'Func';
```

```
type
```

```
MatrizNombres = Array[0..7] of string[7];  
Data          = Array [0..1023] of Byte;  
AllData       = Array[0..8] of record  
    D : Data;  
end;
```

```
Registro = Record
```

```
    Nombre          : MatrizNombres;  
    Canal           : AllData;  
    SPalabra        : string[4];  
    SMuestras       : string[4];  
    Wpalabra        : word;  
    WMuestras       : word;  
    KeyA            : boolean;  
    keyB            : boolean;  
end;
```

```
AToma = Record
```

```
    Canal          : AllData;  
end;
```

```
type
```

```
TRegDialIntercambio = record
```

```
    A : string[2];  
    B : string[2];  
end;
```

```
TRegDialBuscar = record
```

```
    Buscado : string[8];  
    Opcion  : byte;  
end;
```



```
TRegDiallra = record
  Equis : string[4];
end;
```

```
TRegDialFuncion = record
  CanalA : string[2];
  CanalB : string[2];
  flag : boolean;
  Opera : word;
end;
```

```
TRegDialReset = record
  Check : Word;
end;
```

```
TRegDialPalabra = record
  Palabra : string[4];
end;
```

```
TRegDialPreTrigger = record
  Muestras : string[4];
end;
```

```
TRegDialCanales = record
  Nombre : MatrizNombres;
end;
```

```
TRegPrinter = record
  Arranque : string[4];
  Fin : string[4];
end;
```

```
var
  Line : array[0..17] of String;
  ix,jx : byte;
  Toma : AToma;
```

type

```
TDialogPant = function(Dialog: Integer; Info: Pointer): word;
```

type

```
PEtiquetas = ^TEtiquetas;
TEtiquetas = object(TView)
  Etiqueta : MatrizNombres;
  constructor Init(var Bounds: TRect);
  procedure Draw; virtual;
  procedure Actualizar(AEtiqueta: MatrizNombres);
end;
```

```
PDisplay = ^TDisplay;
TDisplay = object(TScroller)
  Adquisicion : Registro;
  EditCanales : TRegDialCanales;
  IsValid : Boolean;
  Modified : Boolean;
  BanderBusq : Boolean;
  Bytebuscado : Byte;
  Cubrir : Byte;
  Bandera : boolean;
```

```

Etiquetas      : PEtiquetas;
constructor Init(var Bounds: TRect; AHScrollBar,
AVScrollBar: PScrollBar; AEtiquetas: PEtiquetas);
procedure Intercambio;
procedure Ordenar;
procedure CambEtiquet;
procedure Normal;
procedure Binario;
procedure Movimlento;
procedure Busqueda(Nmuestra: integer; buscado: byte);
procedure Buscar;
procedure BuscarAgain;
procedure Ira;
procedure Funcion;
procedure Palabra;
procedure PreTrigger;
procedure Reseteo;
procedure ConPre;
procedure Manual;
procedure AutoSin;
procedure SetState(AState: Word; Enable: Boolean); virtual;
procedure SetCmdState(Command: Word; Enable: Boolean);
procedure UpdateCommands; virtual;
procedure HandleEvent(var Event: TEvent); virtual;
procedure Draw; virtual;
procedure ChangeBounds(var Bounds: TRect); virtual;
procedure Actual;
end;

```

type

```

PFileAlog = ^TFileAlog;
TFileAlog = object(TDisplay)
  FileName: FNameStr;
  constructor Init(var Bounds: TRect;
    AHScrollBar, AVScrollBar: PScrollBar;
    AEtiquetas: PEtiquetas; AFileName: FNameStr);
  procedure HandleEvent(var Event: TEvent); virtual;
  function LoadFile: Boolean;
  function Save: Boolean;
  function SaveAs: Boolean;
  function SaveFile: Boolean;
  procedure Impresion;
  procedure UpdateCommands; virtual;
  function Valid(Command: Word): Boolean; virtual;
end;

```

```

PVentanaPrin = ^TVentanaPrin;
TVentanaPrin = object(TWindow)
  ANumberStr : String[3];
  ArchMuestra: PFileAlog;
  constructor Init(var Bounds: TRect;FileName: FNameStr;
  ANumber: Integer);
  function  GetTitle(MaxSize: Integer): TTitleStr; virtual;
  procedure HandleEvent(var Event: TEvent); virtual;

```

end; ----

```

function DefDialogPant(Dialog: Integer; Info: Pointer): word;
procedure ConvertLine(Channel:data; Inicio, Ancho: Integer;
  L: Byte; Normal: Boolean; var hexa: integer);

procedure Manejalnt(Flags,CS,IP,AX,BX,CX,DX,SI,DI,DS,ES,BP: Word); Interrupt;

```

const

```

DialogPant: TDialogPant = DefDialogPant;
A0:         string[2] = "";
B0:         string[2] = "";
buscado0:   string[8] = '00000000';
Equis0:     string[4] = '0';
CanalA0:    string[2] = "";
CanalB0:    string[2] = "";
flag0:      boolean = True;
opera0:     word = 0;
Check0:     word = 0;

```

Implementation

```

function DefDialogPant(Dialog: Integer; Info: Pointer): word;
begin
DefDialogPant := cmCancel;
end;

```

procedure Manejalnt;

```

var
P, j : word;
A, B : byte;
  procedure Ingrese(var cnli , cnls : Data);
  var
  i : word;
  begin
  for i := 0 to 1023 do
  begin
  A := Port[P];
  Inline($90/$90/$90/$90/$90/$90/$90/$90/$90/$90/$90/$90/$90);
  B := Port[P];
  cnli[i] :=(A and $0F) or (B shl 4);
  cnls[i] :=(B and $F0) or (A shr 4);
  end;
  end;

```

Begin

```

Inline($FA); {CLI}
with Toma do for j:=0 to 3 do
  begin
  P:= $2F0 + j;
  Ingrese(canal[2*j].D,canal[2*j+1].D);
  end;
Inline($FB); {STI}
Port[$0A0] := $20; {BORRA EL REGISTRO DE INTERRUPCIONES EN ACTIVO 2}
Port[$020] := $20; {BORRA EL REGISTRO DE INTERRUPCIONES EN ACTIVO 1}
control := (control and $0F) or $20;
Port[$2F1] := Control; {Volver a resetear la IRQ 9}
Listo := True;
Sound(1000);
Delay(10);
NoSound;
end;

```

```

procedure ConvertLine(Channel:data; Inicio, Ancho: Integer;
  L: Byte; Normal: Boolean; var Hexa: Integer);

```

```
var
```

```

BitActual,
BitAnterior,
operacion : Boolean;
SA, SB, SC, SD : String;

```

```

orden[ent1] := orden[ent2];
orden[ent2] := i;
i := 2*ent1; A0:= A;
j := 2*ent2; B0:= B;
With adquisicion Do
begin
trans := canal[ent1].D;
canal[ent1].D := canal[ent2].D;
canal[ent2].D:= trans;
y := nombre[ent1];
nombre[ent1]:=nombre[ent2];
nombre[ent2] := y;
EditCanales.Nombre:= Nombre;
Clave1:= false;
end;
Actual;
Etiquetas^.Actualizar(Adquisicion.Nombre);
end;
end;
end;
end;

```

```

procedure TDisplay.Ordenar;

```

```

var x,i,j :byte;
trans : data;
y : string;

```

```

begin
j:= 0;
for i:= 0 to 6 do
begin
if orden[i]<>i then
begin
for j:=i+1 to 7 do
begin
if orden[j] = i then
With adquisicion Do
begin
x:= orden[i];
orden[i] := orden[j];
orden[j] := x;
trans := canal[i].D;
canal[i].D := canal[j].D;
canal[j].D:= trans;
y := nombre[i];
nombre[i]:=nombre[j];
nombre[j] := y;
EditCanales.Nombre:= Nombre;
Clave1:= false;
end;
end;
end;
end;
if j <> 0 then
begin
Actual;
Etiquetas^.Actualizar(Adquisicion.Nombre);
end;
end;

```

```

procedure TDisplay.CambEtiquet;

```

```

begin

```

```

if DialogPant(edCambEtiquet, @EditCanales) <> cmCancel then

```

```
Adquisicion.Nombre:= EditCanales.Nombre;  
Etiquetas^.Actualizar(Adquisicion.nombre);  
modified:= true;  
end;
```

```
procedure TDisplay.Normal;
```

```
begin  
if Bandera = False then  
  Bandera := not Bandera  
else exit;  
Actual;  
End;
```

```
procedure TDisplay.Binario;
```

```
begin  
if Bandera = True then  
  Bandera := not Bandera  
else exit;  
Actual;  
End;
```

```
procedure TDisplay.Movimiento;
```

```
var  
x : integer;  
begin  
repeat  
x:= Delta.x + 1;  
ScrollTo(x, Delta.y);  
until keypressed or (x = 8191);  
end;
```

```
procedure TDisplay.Busqueda(Nmuestra: integer; buscado: byte);
```

```
const  
byteformado : byte = $00;  
var  
sombra: byte;
```

```
function bit(Numero: integer; channel: data): boolean;
```

```
var  
mask: byte;
```

```
begin  
mask:= $01 shl (Numero mod 8);  
bit := (channel[(Numero div 8) and mask <> 0]);  
end;
```

```
procedure Octeto;
```

```
var  
i : byte;  
begin  
with Adquisicion Do for I := 7 downto 0 do  
begin  
byteformado := byteformado shl 1;  
if bit(Nmuestra, canal[i].D) = True then  
Inc(byteformado);  
end;  
end;  
end;
```

```
begin  
Repeat  
Octeto;  
inc(Nmuestra);
```

```

sombra:= byteformado and not Cubrir;
until ((sombra = buscado) or (Nmuestra = 8192));
if Nmuestra = 8192 then MessageBox(^C'No se encontró al dato:'+buscado0,
nil, mfError + mfokButton) else Scrollto(Nmuestra-1, delta.y);
end;

```

```

procedure TDisplay.Buscar;

```

```

const

```

```

    inicial: integer = 0;

```

```

var

```

```

i      : byte;

```

```

chr    : char;

```

```

RegDialBuscar: TRegDialBuscar;

```

```

begin

```

```

with RegDialBuscar do

```

```

    begin

```

```

        Buscado := buscado0;

```

```

        Opcion := 0;

```

```

        bytebuscado := 0;

```

```

        cubrir := 0;

```

```

        if DialogPant(edbuscar, @RegDialBuscar) <> cmCancel then

```

```

            begin

```

```

                for i:= 1 to 8 do

```

```

                    begin

```

```

                        chr:=upcase(Buscado[i]);

```

```

                        case chr of

```

```

                            '0','1','X': begin

```

```

                                bytebuscado := bytebuscado shl 1;

```

```

                                cubrir := cubrir shl 1;

```

```

                                if chr='X' then inc(cubrir);

```

```

                                if chr='1' then inc(bytebuscado);

```

```

                                end;

```

```

                            else MessageBox(^C'Caracteres válidos: 0, 1 y X',

```

```

                                nil, mfError + mfokButton);

```

```

                    i:= 0;

```

```

                    exit;

```

```

                end;

```

```

            end;

```

```

        if i <> 0 then

```

```

            begin

```

```

                BanderBusq := True;

```

```

                buscado0:= buscado;

```

```

                if RegDialBuscar.Opcion = 0 then Busqueda(inicial, bytebuscado)

```

```

                else BuscarAgain

```

```

            end;

```

```

        end;

```

```

    end;

```

```

end;

```

```

procedure TDisplay.BuscarAgain;

```

```

begin

```

```

if BanderBusq = true then Busqueda(delta.x+1, bytebuscado);

```

```

end;

```

```

procedure TDisplay.lra;

```

```

var

```

```

Intira, codigo      : integer;

```

```

RegDiallra: TRegDiallra;

```

```

begin

```

```

with RegDiallra do

```

```

begin
  Equis:= Equis0;
if DialogPant(edIra, @RegDialIra) <> cmCancel then
  begin
  val(Equis, Intira, codigo);
  if ((codigo <> 0) or (Intira > 8191)) then
    MessageBox('^C'Ingrese un número menor que 8192',
    nil, mfError + mfokButton)
  else begin
    Scrollto(Intira, Delta.y);
    Equis0:= Equis;
    end;
  end;
end;
end;

procedure TDisplay.Funcion;
var
k: integer;
cod1, cod2 : integer;
Ent:set of byte;
RegDialFuncion: TRegDialFuncion;

begin
ent:=[0..7];
  With RegDialFuncion do
  begin
  CanalA := CanalA0;
  CanalB := CanalB0;
  flag := flag0;
  opera := 00;
  if DialogPant(edFuncion, @RegDialFuncion) <> cmCancel then
  begin
  val(canalA,ix,cod1);
  val(canalB,jx,cod2);
  if ((not (ix in ent)) or (not (jx in ent)) or
(cod1+cod2 <> 0)) then MessageBox(
^C'El Número es Errado. Intente nuevamente',nil,mferror+ mfokbutton)
  else
  begin
  CanalA0:=CanalA;
  CanalB0:=canalB;
  flag0:= flag;
  if flag = true then
  begin
  clave1:= true;
  with Adquisicion do
  begin
  case opera of
    $000 :begin
      for k:=0 to 1023 do
        canal[8].D[k]:= canal[ix].D[k] AND canal[jx].D[k];
        Fb:='AND';
        end;
    $100 :begin
      for k:=0 to 1023 do
        canal[8].D[k]:= canal[ix].D[k] OR canal[jx].D[k];
        Fb:='OR';
        end;
    $200 :begin
      for k:=0 to 1023 do
        canal[8].D[k]:= canal[ix].D[k] XOR canal[jx].D[k];
        Fb:='XOR';
        end;
    $300 :begin
      for k:=0 to 1023 do
        canal[8].D[k]:= not(canal[ix].D[k]);

```

```

        Fb:='NOT';
        end;
$400 :begin
        for k:=0 to 1023 do
        canal[8].D[k]:=NOT(canal[ix].D[k] AND canal[jx].D[k]);
        Fb:='NAND';
        end;
$500 :begin
        for k:=0 to 1023 do
        canal[8].D[k]:=NOT(canal[ix].D[k] OR canal[jx].D[k]);
        Fb:='NOR';
        end;
$600 :begin
        for k:=0 to 1023 do
        canal[8].D[k]:=NOT(canal[ix].D[k] XOR canal[jx].D[k]);
        Fb:='NOREX';
        end;
$700 :begin
        for k:=0 to 1023 do
        canal[8].D[k]:= canal[ix].D[k] AND not canal[jx].D[k];
        Fb:='/' ;
        end;
$800 :begin
        for k:=0 to 1023 do
        canal[8].D[k]:=not canal[ix].D[k] OR canal[jx].D[k];
        Fb:='f>';
        end;
end;

```

```

end;
end
else clave1:= false;
Actual;
Etiquetas^.Actualizar(Adquisicion.Nombre);
end;

```

```

end;
end;
end;

```

```

procedure TDisplay.Palabra;

```

```

var
RegDialPalabra: TRegDialPalabra;

```

```

begin
With RegDialPalabra do
begin
palabra:= Adquisicion.Spalabra;
if DialogPant(edPalabra, @RegDialPalabra) <> cmCancel then
begin
if Valido(palabra) then
begin
hexaDec(palabra,Adquisicion.Wpalabra);
Adquisicion.SPalabra := palabra;
end
else messagebox(
^C'El número que ha ingresado no es HEXADECIMAL. Intente nuevamente!',
nil,mfError + mfOkButton);
end;
end;
end;
end;

```

```

procedure TDisplay.Pretrigger;
var
RegDialPreTrigger: TRegDialPreTrigger;

```



```

Begin
With RegDialPreTrigger do
begin
Muestras:= Adquisicion.SMuestras;
if DialogPant(edPreTrigger, @RegDialPreTrigger) <> cmCancel then
begin
if CadADec(Muestras, Adquisicion.WMuestras) then
begin
Str(Adquisicion.WMuestras, Muestras);
Adquisicion.SMuestras := Muestras;
end
else messagebox(^C'Ingrese un Número entero'#13+^C'entre 4 y 8188',
nil,mfError + mfOkButton);
end;
end;
end;

```

```

procedure TDisplay.Reseteo;

```

```

begin
Port[$2F1]:= $00;
end;

```

```

procedure TDisplay.ConPre;
var

```

```

OffSet : word;
{Control: byte;}

```

```

begin
Adquisicion.KeyA:= false;
Adquisicion.KeyB:= false;
Listo := false;
OffSet := $800 - (Adquisicion.WMuestras div 4);
Control:= $60 or ($07 and Hi(OffSet));
Port[$2F0]:= Lo(OffSet);
Port[$2F1]:= Control;
Port[$2F2]:= Hi(Adquisicion.WPalabra); {byte mas significativo}
Port[$2F3]:= Lo(Adquisicion.WPalabra); {byte menos significativo}
Control:= $E0 or Control;
Port[$2F1]:= Control;
Clave0:= True;
Actual;
Repeat
if keypressed then
begin
clave0:= false;
Actual;
Reseteo;
Exit;
end
until Listo;
clave0:= False;
Adquisicion.KeyA:= True;
Adquisicion.KeyB:= True;
Adquisicion.Canal:= Toma.Canal;
Actual;
Modified:= True;
end;

```

```

procedure TDisplay.Manual; {Rutina para Manual}

```

```

begin

```

```

Adquisicion.KeyA:= false;
Adquisicion.KeyB:= false;
Listo := false;
Port[$2F1]:=$20; (* 00100000 *)
Port[$2F1]:=$A0; (* 10100000 *)
Repeat until Listo;
Adquisicion.Canal:= Toma.Canal;
Actual;
modified:= true;
end;

```

```

procedure TDisplay.AutoSin;

```

```

begin
Adquisicion.KeyA:= false;
Adquisicion.KeyB:= false;
Listo := false;
Port[$2F2]:= Hi(Adquisicion.WPalabra); (byte mas significativo)
Port[$2F3]:= Lo(Adquisicion.WPalabra); (byte menos significativo)
Port[$2F1]:= $00;
Port[$2F1]:= $80;
clave0:= True;
Actual;
Repeat
  if keypressed then
    begin
      clave0:= false;
      Actual;
      Reseteo;
      Exit;
    end
  until Listo;
clave0:= False;
Adquisicion.KeyA:= True;
Adquisicion.Canal:= Toma.Canal;
Actual;
Modified:= true;

end;

```

```

procedure TDisplay.SetState(AState: Word; Enable: Boolean);

```

```

begin
  TScroller.SetState(AState, Enable);
  case AState of
    sfActive:
      UpdateCommands;
  end;
end;

```

```

procedure TDisplay.SetCmdState(Command: Word; Enable: Boolean);

```

```

var
  S: TCommandSet;
begin
  S := [Command];
  if Enable and (State and sfActive <> 0) then
    EnableCommands(S) else DisableCommands(S);
end;

```

```

procedure TDisplay.UpdateCommands;

```

```

begin
  SetCmdState(cmPalabra, True);
  SetCmdState(cmPreTrigger, True);
  SetCmdState(cmReset, True);
  SetCmdState(cmCanales, True);
  SetCmdState(cmConPre, True);
  SetCmdState(cmManual, True);
  SetCmdState(cmAutoSin, True);

```

```

SetCmdState(cmMuestra, True);
SetCmdState(cmCambEtiquet, True);
SetCmdState(cmIntercambio, True);
SetCmdState(cmNormal, True);
SetCmdState(cmBinario, True);
SetCmdState(cmMovimiento, True);
SetCmdState(cmBuscar, True);
SetCmdState(cmBuscarAgain, True);
SetCmdState(cmlra, True);
SetCmdState(cmFuncion, True);
SetCmdState(cmOrden, True);

end;

```

```

procedure TDisplay.HandleEvent(var Event: TEvent);
begin

```

```

TScroller.HandleEvent(Event);
case Event.What of

```

```

    evCommand:
    case Event.Command of
        cmCambEtiquet : CambEtiquet;
        cmIntercambio : Intercambio;
        cmNormal      : Normal;
        cmBinario     : Binario;
        cmMovimiento  : Movimiento;
        cmBuscar      : Buscar;
        cmBuscarAgain : BuscarAgain;
        cmlra         : lra;
        cmFuncion     : Funcion;
        cmPalabra     : Palabra;
        cmPreTrigger  : PreTrigger;
        cmReset       : Reseteo;
        cmConPre      : ConPre;
        cmManual      : Manual;
        cmAutoSin     : AutoSin;
        cmOrden       : Ordenar;
    end;

```

```

end;
end;

```

```

procedure TDisplay.Draw;
var

```

```

    Color: Byte;
    I, Y: Integer;
    B: TDrawBuffer;
    L: array[0..1] of Longint;
    S: String[15];
    Datohex : integer;
    X: byte;
begin
    Datohex:= 0;
    Color := GetColor(1);
    if Clave1 then X:= 8 else X:= 7;
    with Adquisicion do for I := 0 to X do
        ConvertLine(Canal[I].D, Delta.x, Size.x, 2*I, Bandera, Datohex);
        for Y := 0 to Size.Y - 1 do
            begin
                MoveChar(B, ' ', Color, Size.X);
                I := Delta.Y + Y;
                if (I < 2*x+2) then
                    MoveStr(B, Line[I], Color);
                WriteLine(0, Y, Size.X, 1, B);
            end;
        end;
    end;

```

```

L[0] := Delta.X;
L[1] := Datohex;
FormatStr(S, '%d:%x', L);
MoveStr(B[5 - Pos(':', S)], S, Color);
Writeln(0, size.y-1, 9, 1, B);
if clave0 then
begin
MoveStr(B, 'Presione cualquier tecla para detener el Muestreo', Color);
Writeln(10, Size.y-1, 50, 1, B);
end;
if Adquisicion.KeyA then
begin
MoveStr(B, 'Palabra de Disparo:' + Adquisicion.SPalabra, Color);
Writeln(8, Size.y-1, 25, 1, B);
end;
if Adquisicion.KeyB then
begin
MoveStr(B, 'Muestras antes del Disparo = ' + Adquisicion.SMuestras, Color);
Writeln(32, Size.y-1, 50, 1, B);
end;
end;

```

```

procedure TDisplay.ChangeBounds(var Bounds: TRect);
begin
TScroller.ChangeBounds(Bounds);
SetLimit(8191 + Size.x, 16);
end;

```

```

procedure TDisplay.Actual;

```

```

begin
UpDateCommands;
DrawView;
end;

```

{ TFileAlog }

```

constructor TFileAlog.Init(var Bounds: TRect;
    AHScrollBar, AVScrollBar: PScrollBar;
    AEtiquetas: PEtiquetas; AFileName: FNameStr);
begin
TDisplay.Init(Bounds, AHScrollBar, AVScrollBar, AEtiquetas);
if AFileName <> '' then
begin
FileName := FExpand(AFileName);
if IsValid then IsValid := LoadFile;
end;
end;

```

```

procedure TFileAlog.HandleEvent(var Event: TEvent);
begin
TDisplay.HandleEvent(Event);
case Event.What of
evCommand:
case Event.Command of
cmSave: Save;
cmSaveAs: SaveAs;
cmImpresion: Impresion;
else
Exit;
end;
else
end;

```

```
    Exit;
end;
ClearEvent(Event);
end;
```

```
function TFileAlog.LoadFile: Boolean;
```

```
var
```

```
    Length: Word;
    FSize: Longint;
    F: File of Registro;
```

```
begin
```

```
    LoadFile := False;
```

```
    Length := 0;
```

```
    Assign(F, FileName);
```

```
    Reset(F);
```

```
    if IOResult <> 0 then LoadFile := True else
```

```
begin
```

```
    FSize := FileSize(F);
```

```
    if (FSize > $FFFF0) then
```

```
        DialogPant(edOutOfMemory, nil) else
```

```
begin
```

```
    Read(F, Adquisicion);
```

```
    if IOResult <> 0 then DialogPant(edReadError, @FileName) else
```

```
begin
```

```
    LoadFile := True;
```

```
    Length := FSize;
```

```
    EditCanales.Nombre := Adquisicion.Nombre;
```

```
    Etiquetas^.Etiqueta:= Adquisicion.Nombre;
```

```
end;
```

```
end;
```

```
Close(F);
```

```
end;
```

```
end;
```

```
function TFileAlog.Save: Boolean;
```

```
begin
```

```
    if FileName = "" then Save := SaveAs else Save := SaveFile;
```

```
end;
```

```
function TFileAlog.SaveAs: Boolean;
```

```
begin
```

```
    SaveAs := False;
```

```
    if DialogPant(edSaveAs, @FileName) <> cmCancel then
```

```
begin
```

```
    FileName := FExpand(FileName);
```

```
    Message(Owner, evBroadcast, cmUpdateTitle, nil);
```

```
    SaveAs := SaveFile;
```

```
end;
```

```
end;
```

```
function TFileAlog.SaveFile: Boolean;
```

```
var
```

```
    F: File of Registro;
```

```
    BackupName: FNameStr;
```

```
    D: DirStr;
```

```
    N: NameStr;
```

```
    E: ExtStr;
```

```
begin
```

```
    SaveFile := False;
```

```
    FSplit(FileName, D, N, E);
```

```
    BackupName := D + N + '.BAK';
```

```

Assign(F, BackupName);
Erase(F);
Assign(F, FileName);
Rename(F, BackupName);
InOutRes := 0;
Assign(F, FileName);
Rewrite(F);
if IOResult <> 0 then DialogPant(edCreateError, @FileName) else
begin
Write(F, adquisicion);
if IOResult <> 0 then DialogPant(edWriteError, @FileName) else
begin
Modified := False;
SaveFile := True;
end;
Close(F);
end;
end;
end;

```

```

procedure TFileAlog.Impresion;
var

```

```

Dif, Caras,Z : integer;
RegPrinter: TRegPrinter;
IntArraque,IntFin, cod1,cod2 : integer;

```

```

const
days : array [0..6] of String[9] =
('Domingo','Lunes','Martes',
'Miércoles','Jueves','Viernes',
'Sábado');

```

```

var
y, m, d, dow : Word;

```

```

procedure PrinterCara(Comienzo:integer);

```

```

var
l, datoHex : integer;
x : byte;

```

```

begin

```

```

if Clave1 then X:= 8 else X:= 7;
write(lst, ' ');
writeln(lst, ' ');
with Adquisicion do for l := 0 to X do
ConvertLine(Canal[l].D, Comienzo, 120 , 2*l, Bandera, DatoHex);
for l:= 0 to 2*X+1 do
begin
write(lst, '|', #13);
if i=17 then if Fb='NOT' then write(lst, Fb+' C'+chr(ix+48)) else write(lst, ' C'+chr(ix+48)+' '+Fb+'
C'+chr(jx+48),#13);
if (i mod 2 <> 0) and (i<>17) then write(lst, ' ,i div 2,': ',Adquisicion.nombre[i div 2],#13);
Write(lst, ' ',Line[l]);writeln(lst, '|')
end;
write(lst, '|');
write(lst, '|');
writeln(lst,#13,' ',Comienzo);
write(lst, '|');
writeln(lst, '|');
end;

```

```

begin
With RegPrinter do
begin
Str(Delta.x,Arranque);
Str(Delta.x+120,Fin);
if DialogPant(edPrinter, @RegPrinter) <> cmCancel then
begin
val(Arranque, IntArranque, cod1);
val(Fin, IntFin, cod2);
if ((cod1+ cod2 <> 0) or (IntArranque > 8067)or
(IntFin < IntArranque) or (IntFin > 8192))
then
MessageBox(^C'Ingrese correctamente el inicio y fin de impresión',
nil, mfError + mfokButton)
else begin
write(1st,#15,#11,#27,#71);(SETEA IMPRESION COMPRIMIDA Y DOBLE PASADA)
dif := IntFin-IntArranque+1;
Caras:= Dif div 120;
if dif Mod 120 <> 0 then Inc(Caras);
GetDate(y,m,d,dow);
writeln(1st,'DATOS DEL ARCHIVO: ',FileName);
WriteLn(1st,#13,'Fecha de Impresión: ', days[dow],', ',d:0,'/',m:0,'/',y:0);
if Adquisicion.KeyA then writeln(1st,' Palabra de Disparo: '+Adquisicion.SPalabra,'H');
if Adquisicion.KeyB then writeln(1st,' Muestras antes del Disparo = '+Adquisicion.SMuestras);
for Z := 1 to Caras do
begin
PrinterCara(IntArranque);
IntArranque:= IntArranque + 120;
end;
end;
end;
end;
write(1st,#18);
end;

```

```

procedure TFileAlog.UpdateCommands;
begin
TDisplay.UpdateCommands;
SetCmdState(cmSave, True);
SetCmdState(cmSaveAs, True);
SetCmdState(cmlImpresion, True);
end;

```

```

function TFileAlog.Valid(Command: Word): Boolean;
var
D: Integer;
begin
if Command = cmValid then Valid := IsValid else
begin
Valid := True;
if Modified then
begin
if FileName = "" then D := edSaveUntitled else D := edSaveModify;
case DialogPant(D, @FileName) of
cmYes: Valid := Save;
cmNo: Modified := False;
cmCancel: Valid := False;
end;
end;
end;
end;
end;

```

```

{ TVentanaPrin }
constructor TVentanaPrin.Init(var Bounds: TRect;

```

FileName: FNameStr; ANumber: Integer);

var

R: TRect;
HScrollBar, VScrollBar: PScrollBar;
Etiquetas: PEtiquetas;

function MakeInterior(Bounds: TRect): PFileAlog;

begin

R.Assign(Bounds.A.X+2, Bounds.B.Y-1, Bounds.B.X-2, Bounds.B.Y);
HScrollBar := New(PScrollBar, Init(R));
HScrollBar^.Options := HScrollBar^.Options or ofPostProcess;
Insert(HScrollBar);
Bounds.Grow(-1,-1);
MakeInterior := New(PFileAlog, Init(Bounds, HScrollBar, nil,
Etiquetas, FileName));

end;

begin

TWindow.Init(Bounds, "", ANumber);
Options:=Options or OfTileable;
R.Assign(Bounds.A.X, Bounds.A.Y, Bounds.B.X div 5-2, Bounds.B.Y);
R.Grow(-1,-1);
Etiquetas := New(PEtiquetas,Init(R));
Etiquetas^.GrowMode := gfGrowHiY;
Insert(Etiquetas);
R.Assign(Bounds.B.X div 5-3, Bounds.A.Y div 3, Bounds.B.X, Bounds.B.Y);
ArchMuestra := MakeInterior(R);
ArchMuestra^.GrowMode := gfGrowHiX + gfGrowHiY;
Insert(ArchMuestra);
Str(ANumber, ANumberStr);
end;

function TVentanaPrin.GetTitle(MaxSize: Integer): TTitleStr;

begin

if ArchMuestra^.FileName = ""
then GetTitle := 'Sin Título'+ ' ('+ ANumberStr+ ')'
else GetTitle := ArchMuestra^.FileName + ' ('+ ANumberStr+ ')';
end;

procedure TVentanaPrin.HandleEvent(var Event: TEvent);

begin

TWindow.HandleEvent(Event);
if (Event.What = evBroadcast) and (Event.Command = cmUpdateTitle) then
begin
Frame^.DrawView;
ClearEvent(Event);
end;
end;

end.

```
{*****}  
{  
{ Unidad que contiene las cmxxxx }  
{ }  
{ Copyright (c) 1995 por José Rivera Vigo }  
{ Todos los Derechos reservados }  
{ }  
{*****}
```

unit AlogCmds;

{ \$D- }

interface

const

cmNew = 100;
cmFOpen = 101;
cmChDir = 102;
cmAbout = 103;
cmCalendar = 104;
cmCalculator = 105;
cmDosShell = 106;

const

cmSave = 107;
cmSaveAs = 108;
cmPalabra = 109;
cmPreTrigger = 110;
cmReset = 111;
cmCanales = 112;
cmConPre = 113;
cmManual = 114;
cmAutoSin = 115;
cmMuestra = 116;
cmCambEtiquet = 117;
cmIntercambio = 118;
cmNormal = 119;
cmBinario = 120;
cmMovimiento = 121;
cmBuscar = 122;
cmBuscarAgain = 123;
cmIra = 124;
cmFuncion = 125;
cmUpDateTitle = 126;
cmImpresion = 127;
cmOrden = 128;

implementation

end.

```
{ ..... }  
{  
  Unidad que contiene las hcxxxx  
}  
{  
  Copyright (c) 1995 por José Rivera Vigo  
  Todos los Derechos reservados  
}  
{ ..... }
```

unit hcxxxx;

interface

const

hcAnalyzer = 1;
hcasignar = 27;
hcautomatico = 35;
hcbinario = 42;
hcbusqueda = 23;
hcCalculator = 2;
hcCalendar = 3;
hcCancelBtn = 21;
hcConpre = 32;
hcejecucion = 31;
hcEncontrar = 24;

```

hcFCChDirDBox      = 22;
hcFChangeDir      = 14;
hcFDosShell       = 15;
hcFExit           = 16;
hcFile            = 8;
hcFOFileOpenDBox  = 17;
hcFOFiles         = 19;
hcFOName          = 18;
hcFOOpenBtn       = 20;
hcFOpen           = 9;
hcfuncion         = 43;
hcImpresion       = 13;
hcintercambio     = 38;
hclra             = 26;
hcManual          = 34;
hcmododisplay     = 40;
hcMovimiento      = 44;
hcNoContext       = 0;
hcnomcanal        = 30;
hcnormal          = 41;
hcNuevoA         = 10;
hcorden           = 39;
hcOtraVez         = 25;
hcpaldis         = 28;
hcpreDisparo      = 29;
hcReset           = 36;
hcresultados     = 37;
hcSAbout          = 5;
hcSalvarA         = 11;
hcSalvarComoA     = 12;
hcSCalculator     = 7;
hcSCalendar       = 6;
hcsinpre         = 33;
hcSystem          = 4;
hcWCascade        = 49;
hcWClose          = 52;
hcWindows         = 45;
hcWNext           = 50;
hcWPrevious       = 51;
hcWSizeMove       = 46;
hcWTile           = 48;
hcWZoom           = 47;

```

implementation

end.

```

{.....}
{
{   Unidad de Apoyo para Alog   }
{                               }
{                               }
{   Copyright (c) 1993 por José Rivera Vigo   }
{                               }
{.....}

```

Unit Apoyo;

interface
type

conjcar = set of char;

```

function Valido(var x: string): boolean;
function CadADec(x: string; var i: word): boolean;
procedure HexaDec(x: string; var Decimal: word);

```

implementation

```
function Valido(var x: string): boolean;
```

```
var
```

```
  num : conjcar;  
  v,w : boolean;  
  i   : byte;
```

```
begin
```

```
  num := ['0'..'9','A'..'F'];  
  valido := false;  
  v := true;  
  w := false;
```

```
for i := 1 to Length(x) do
```

```
  begin  
    x[i] := UpCase(x[i]);  
    v := v AND (x[i] in num);  
    valido := v;  
    w := v;  
  end;
```

```
if Length(x) < 4 then for i := 1 to (4-Length(x)) do x := '0'+ x ;  
end;
```

```
procedure HexaDec(x: string;var Decimal: word);
```

```
var
```

```
  hex   : integer;  
  i,k   : word;
```

```
  begin
```

```
    hex := ord(x[4]);  
    if hex < 58 then hex := hex - 48 else hex := hex - 55;  
    decimal := hex;  
    k:=1;  
    for i := 3 downto 1 do  
      begin  
        hex := ord(x[i]);  
        k:= k * 16;  
        if hex < 58 then hex := hex - 48 else hex := hex - 55;  
        decimal := decimal + hex * k;  
      end;  
    end;
```

```
function CadADec(x: string; var i: word): boolean;
```

```
var
```

```
  code : integer;
```

```
begin
```

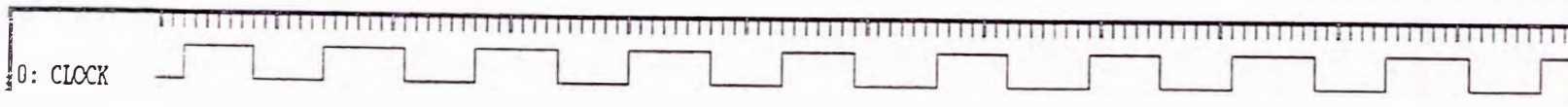
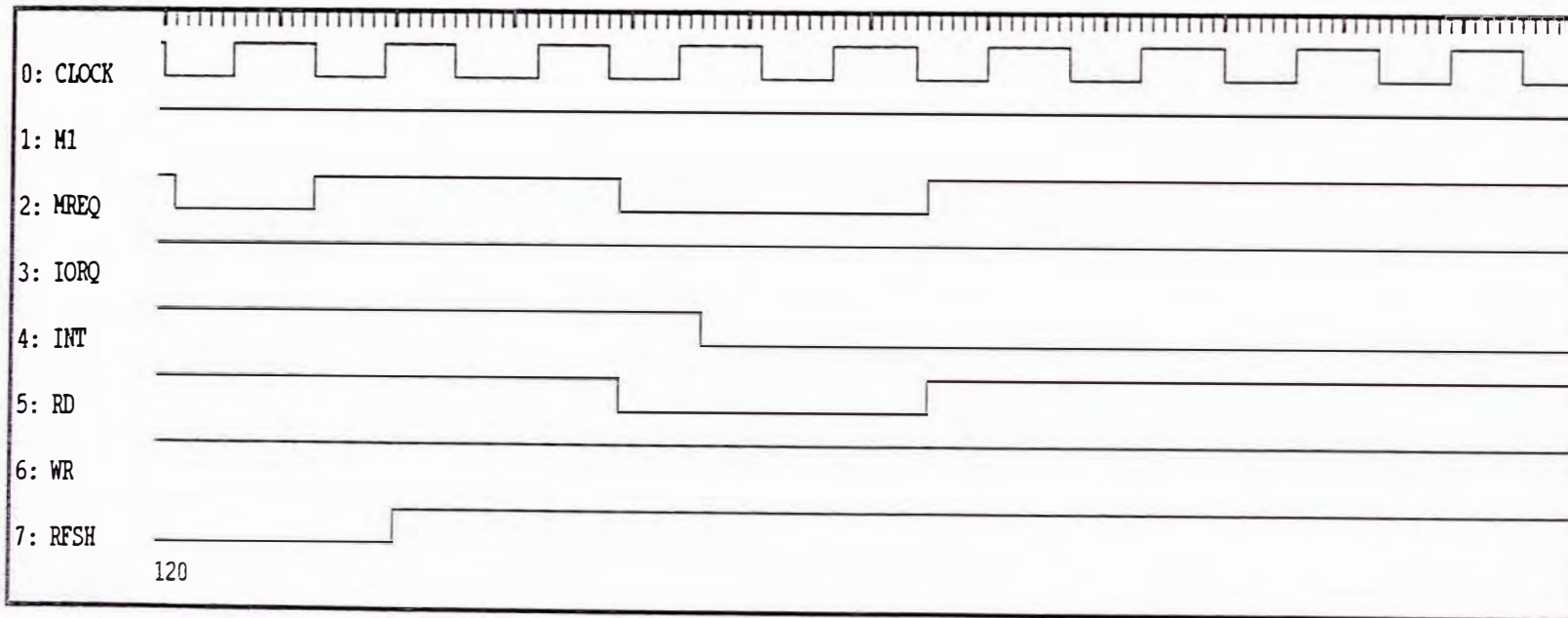
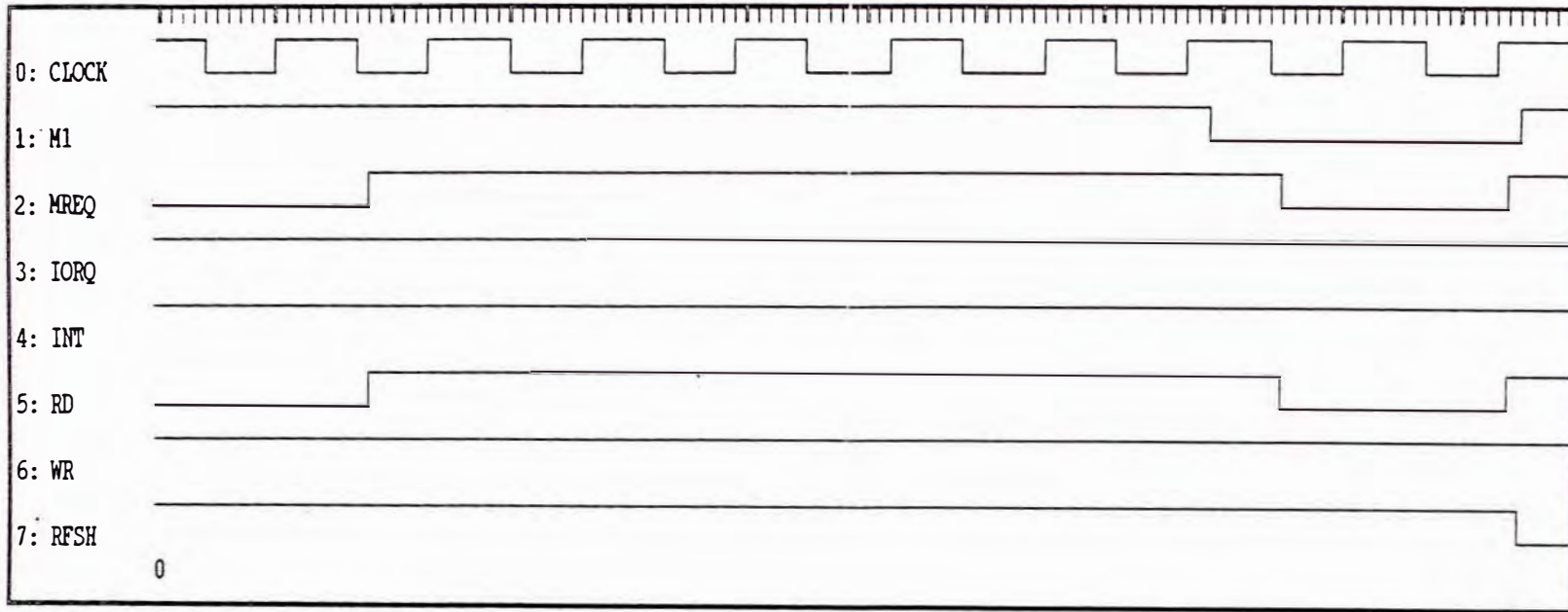
```
  val(x,i,code);  
  if (code = 0) and (i>=4) and (i<=8188) then  
    begin  
      CadADec:= true;  
      i:= 4 * (i div 4);  
    end  
  else CadADec := false;  
end;
```

```
end.
```

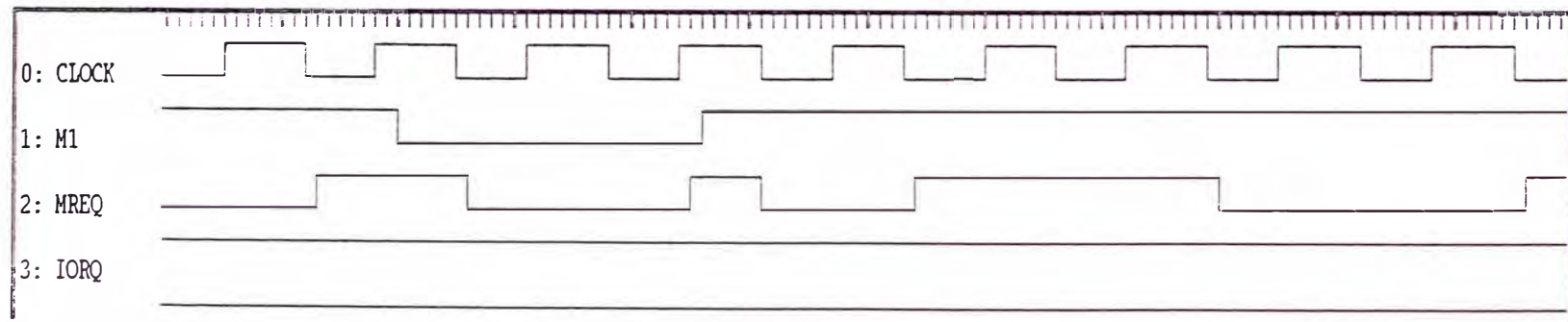
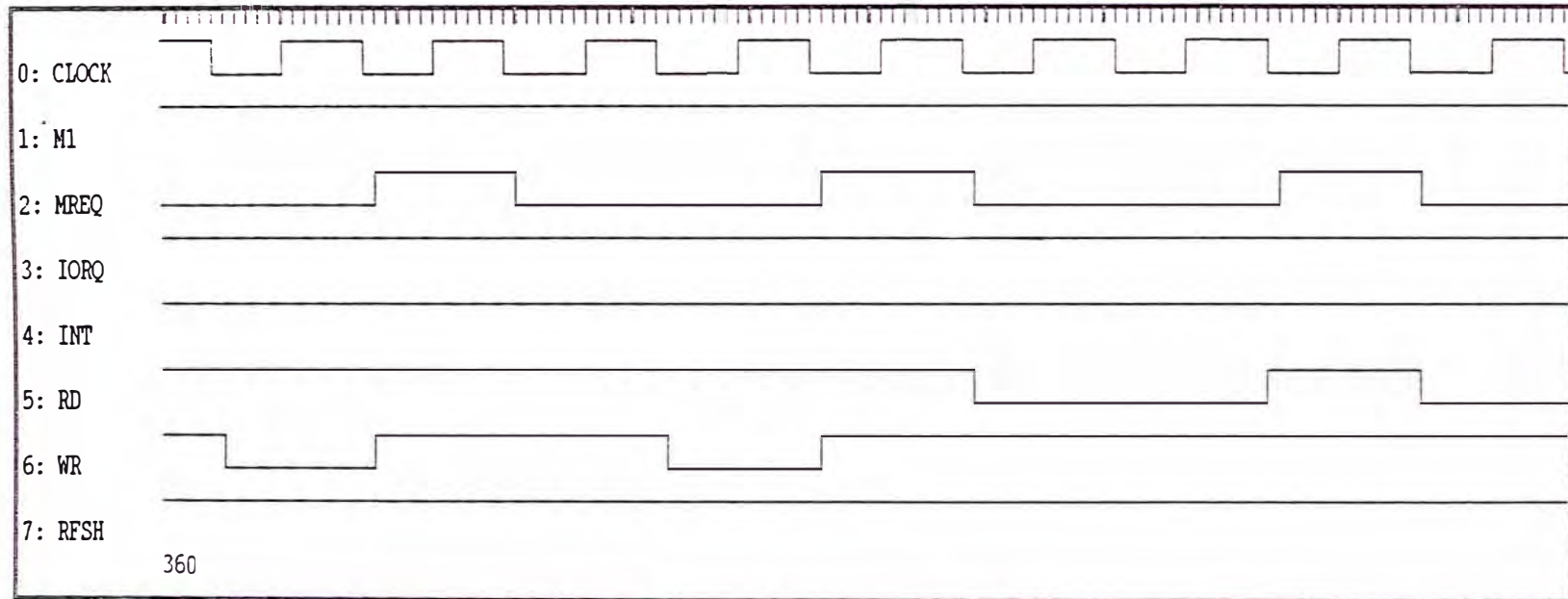
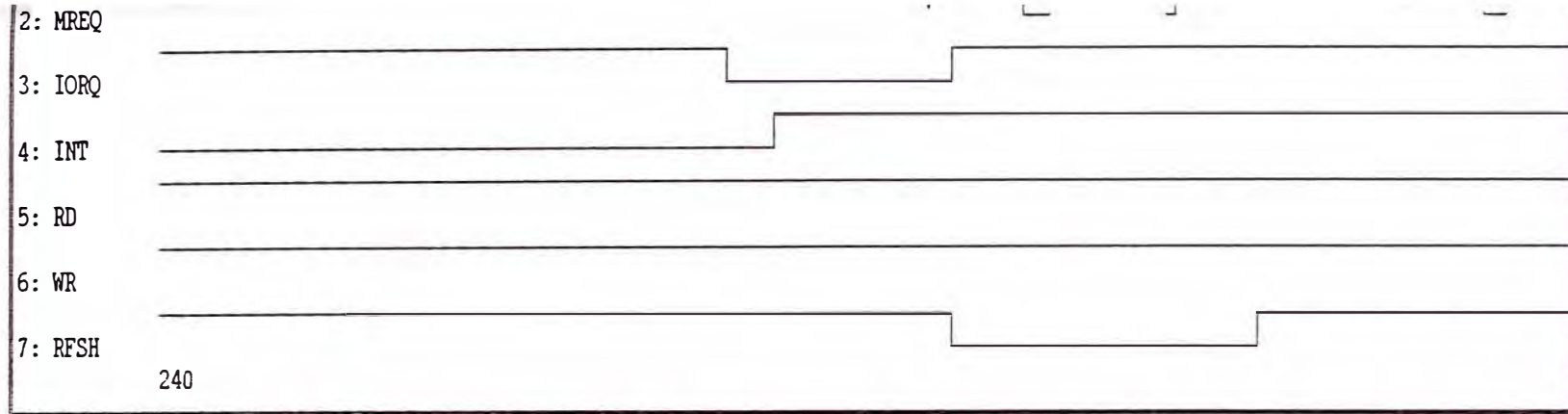
```
??
```

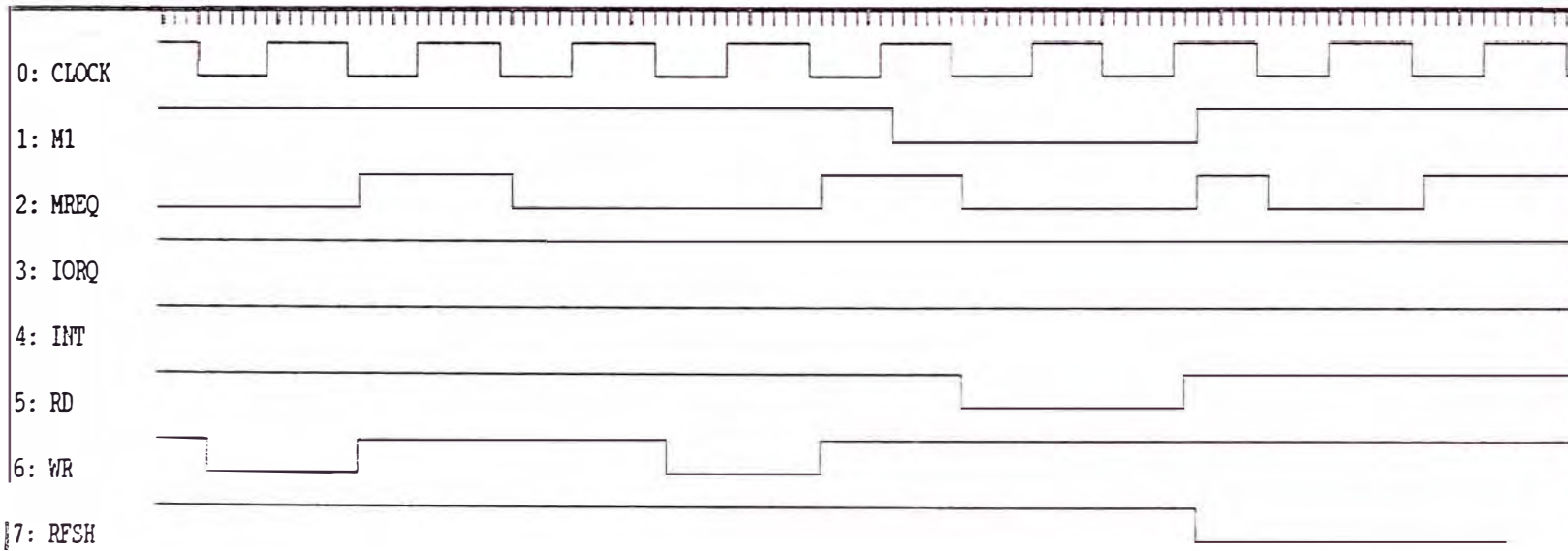
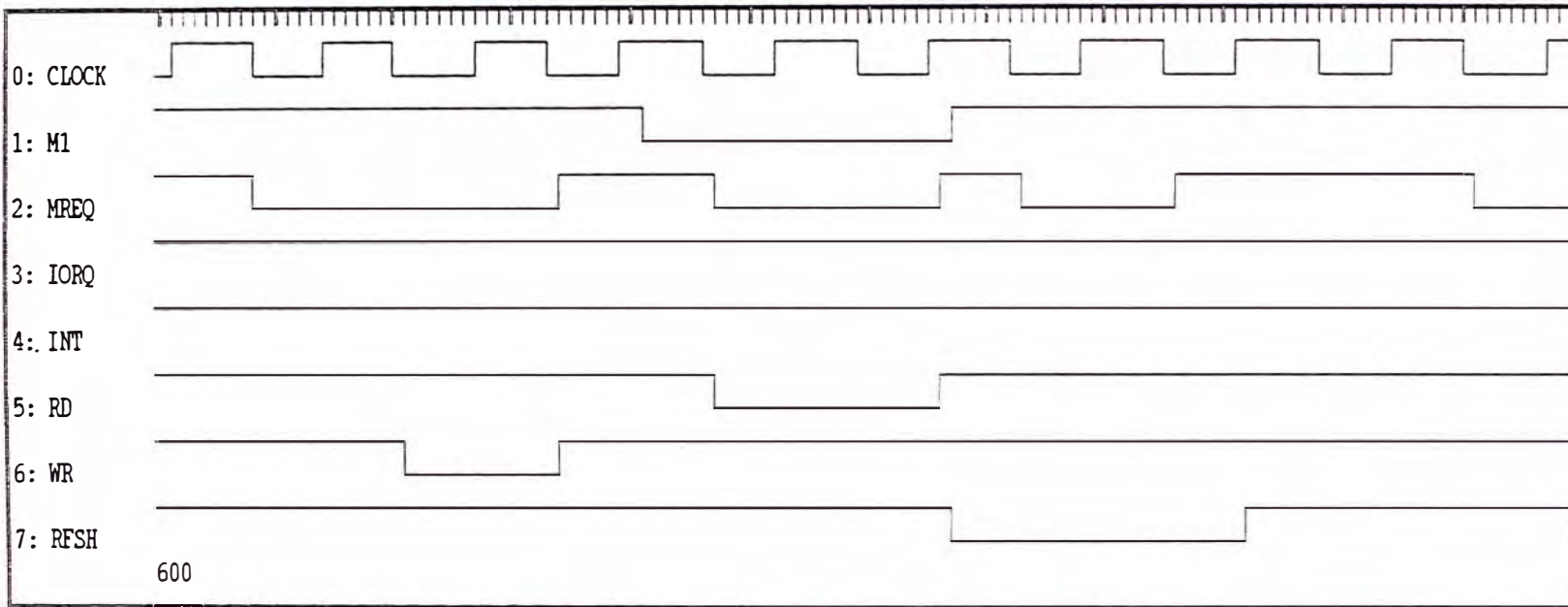
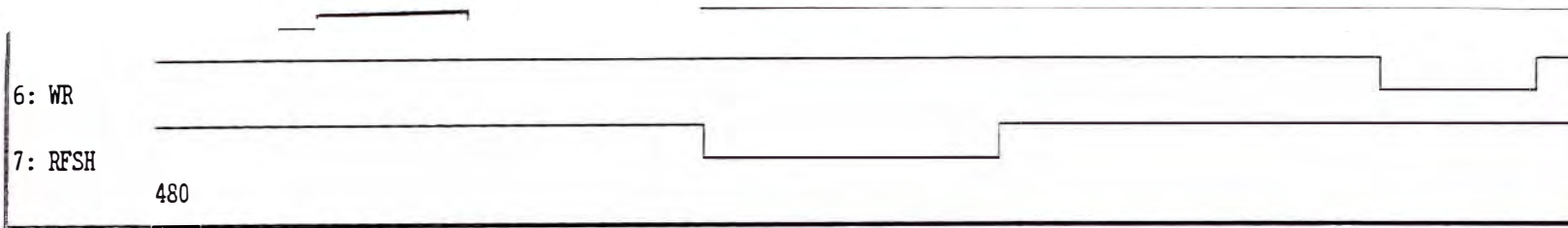
APENDICE C

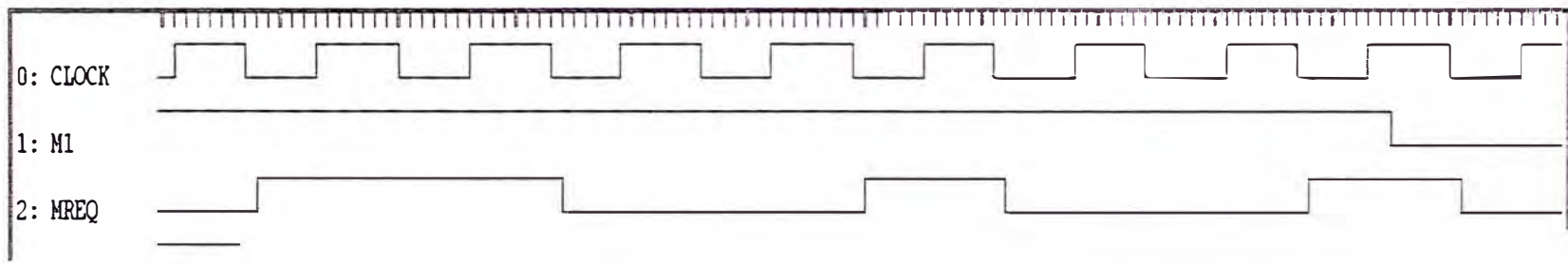
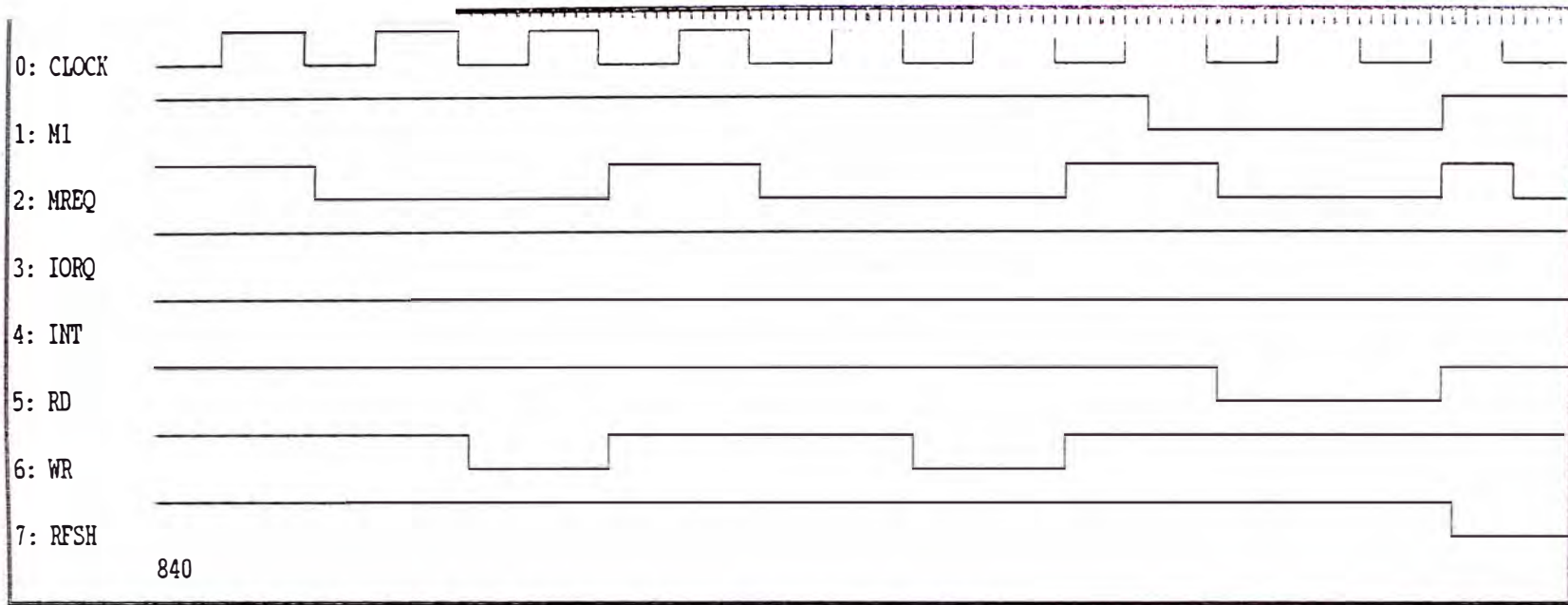
**RESULTADOS EXPERIMENTALES DE LA PRUEBA 3 PARA EL SISTEMA
BASADO EN EL μ P Z80.**

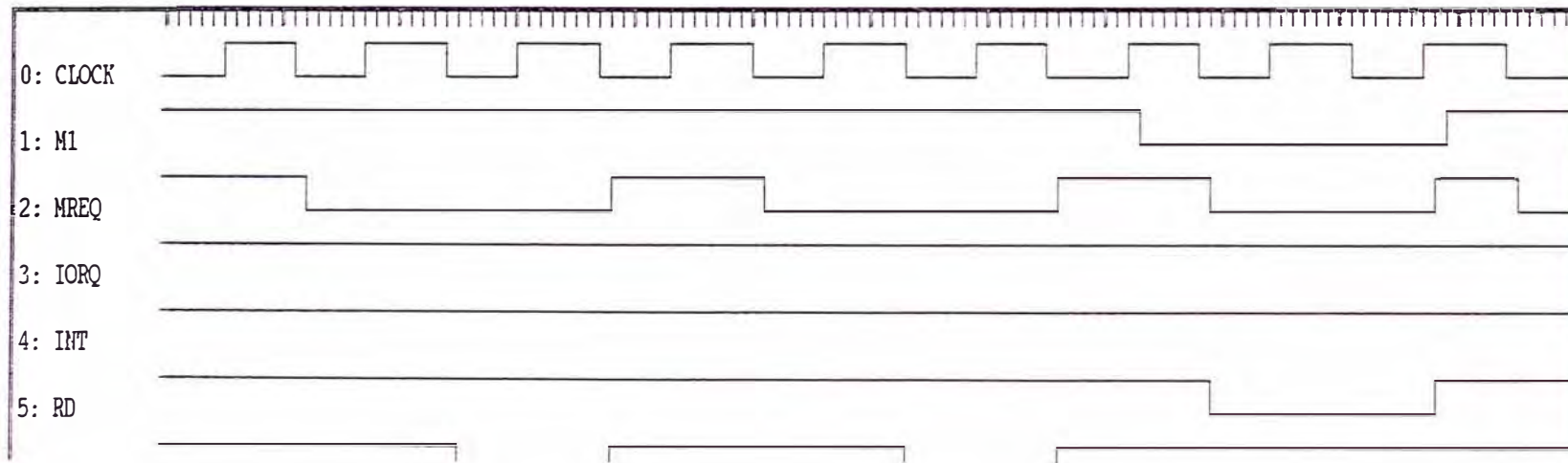
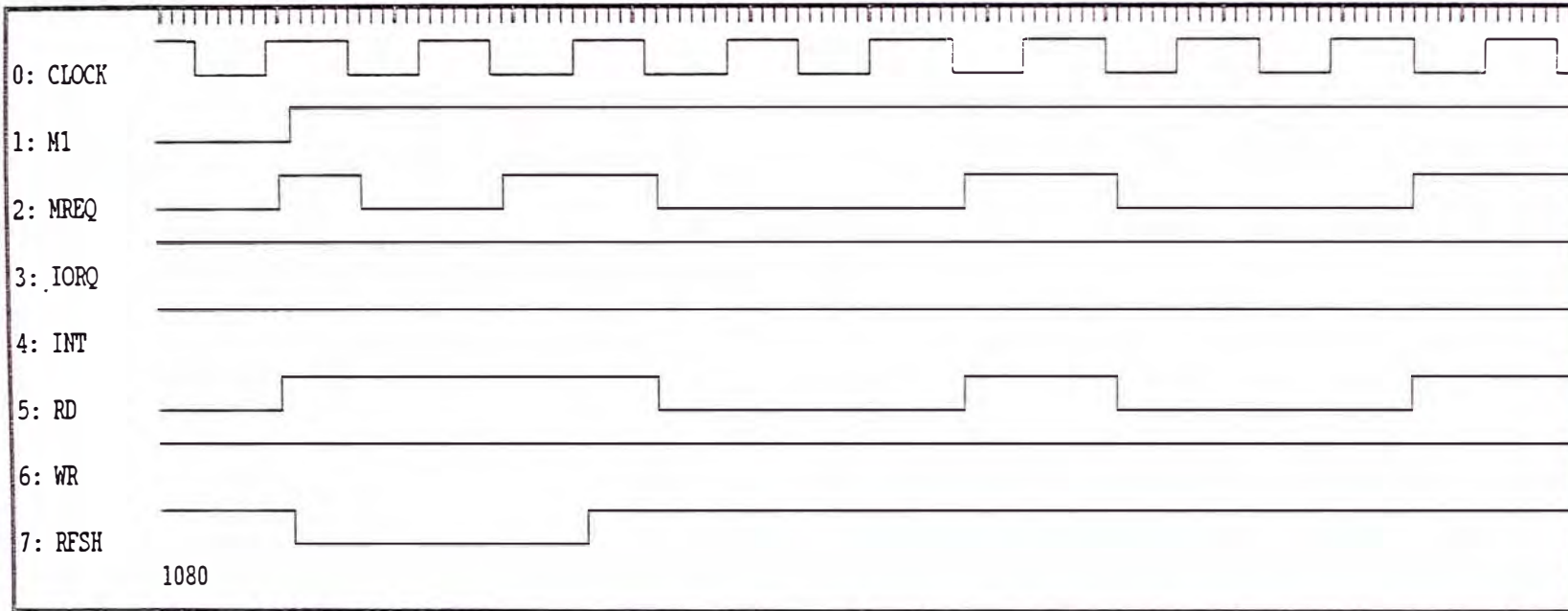
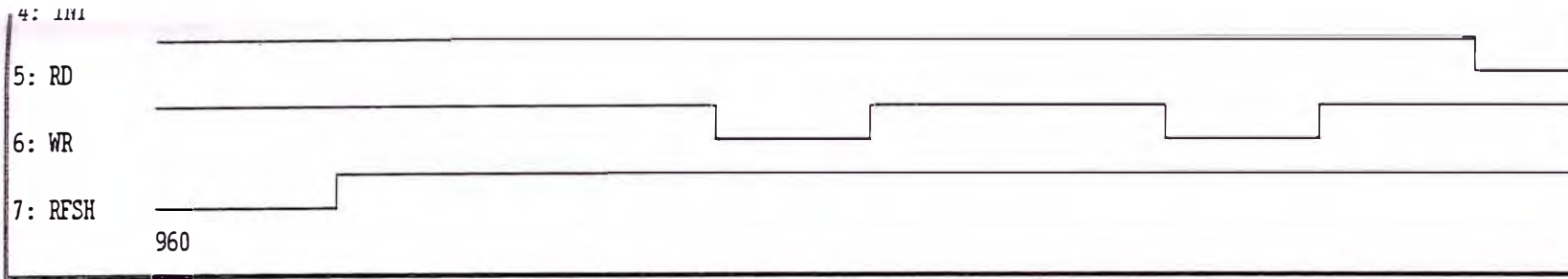


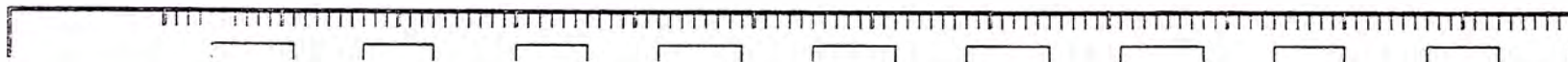
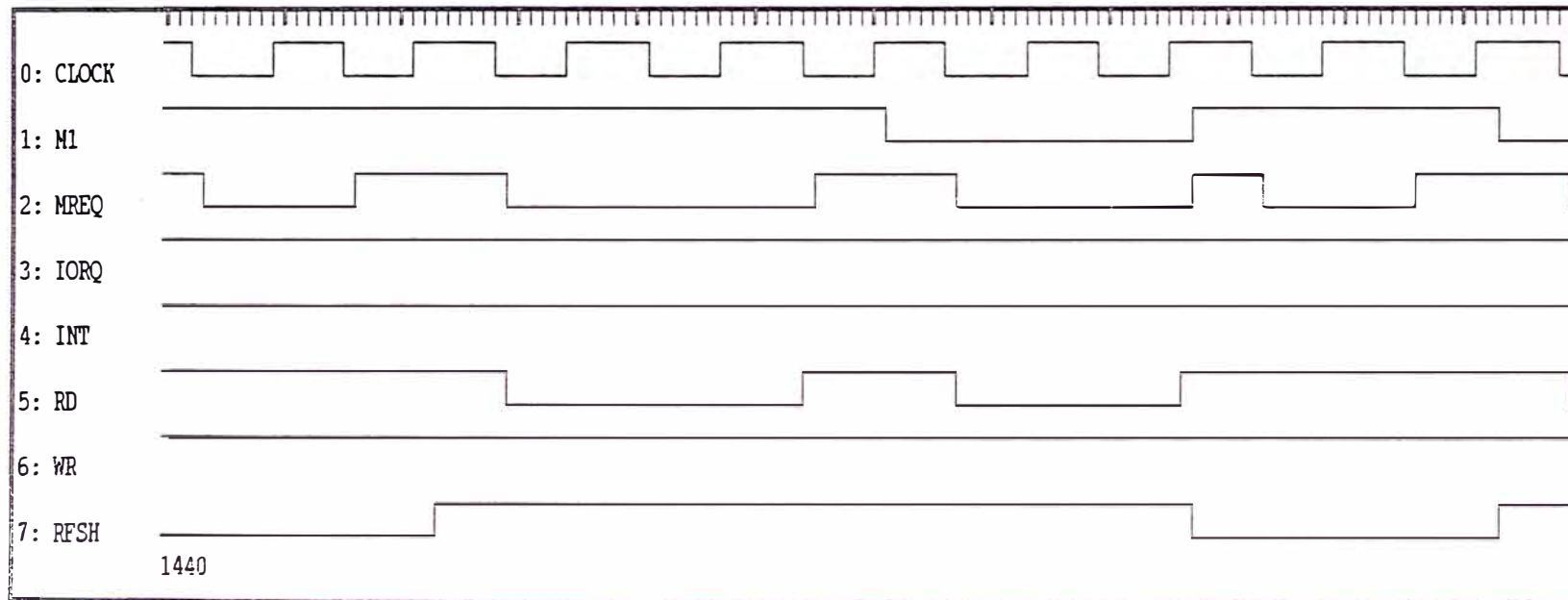
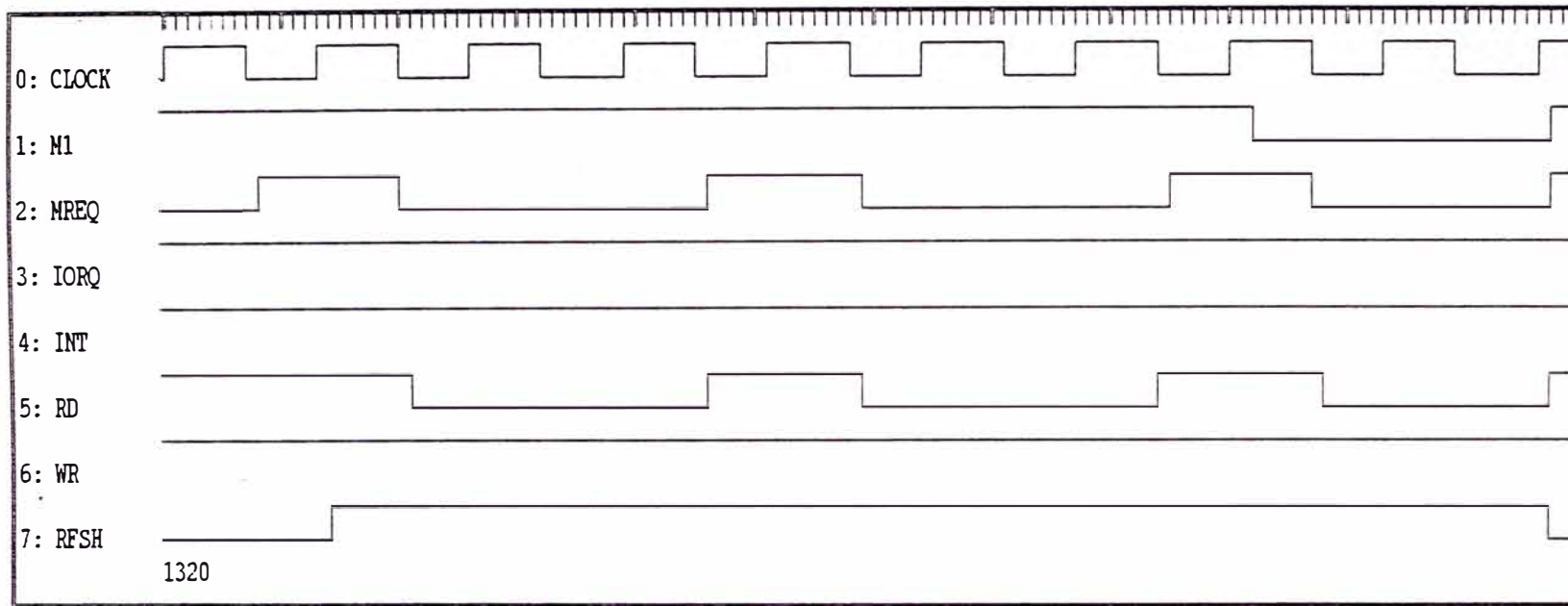
1. M1

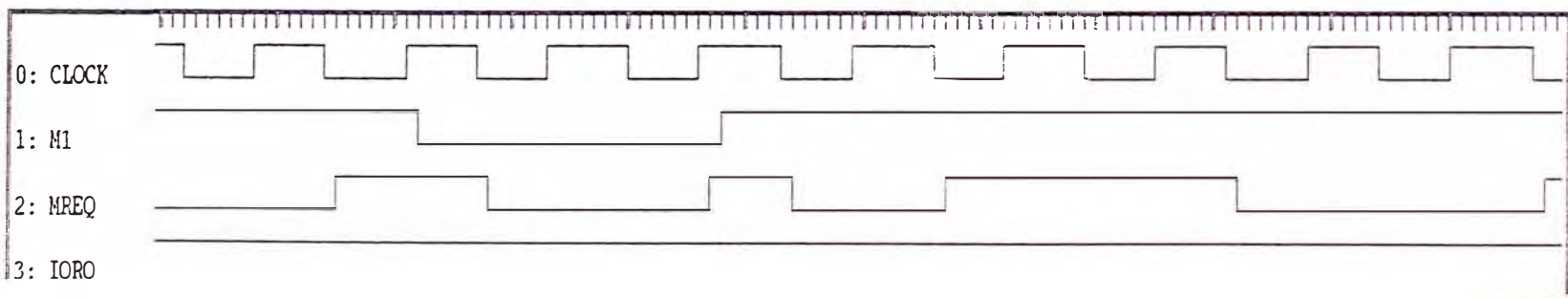
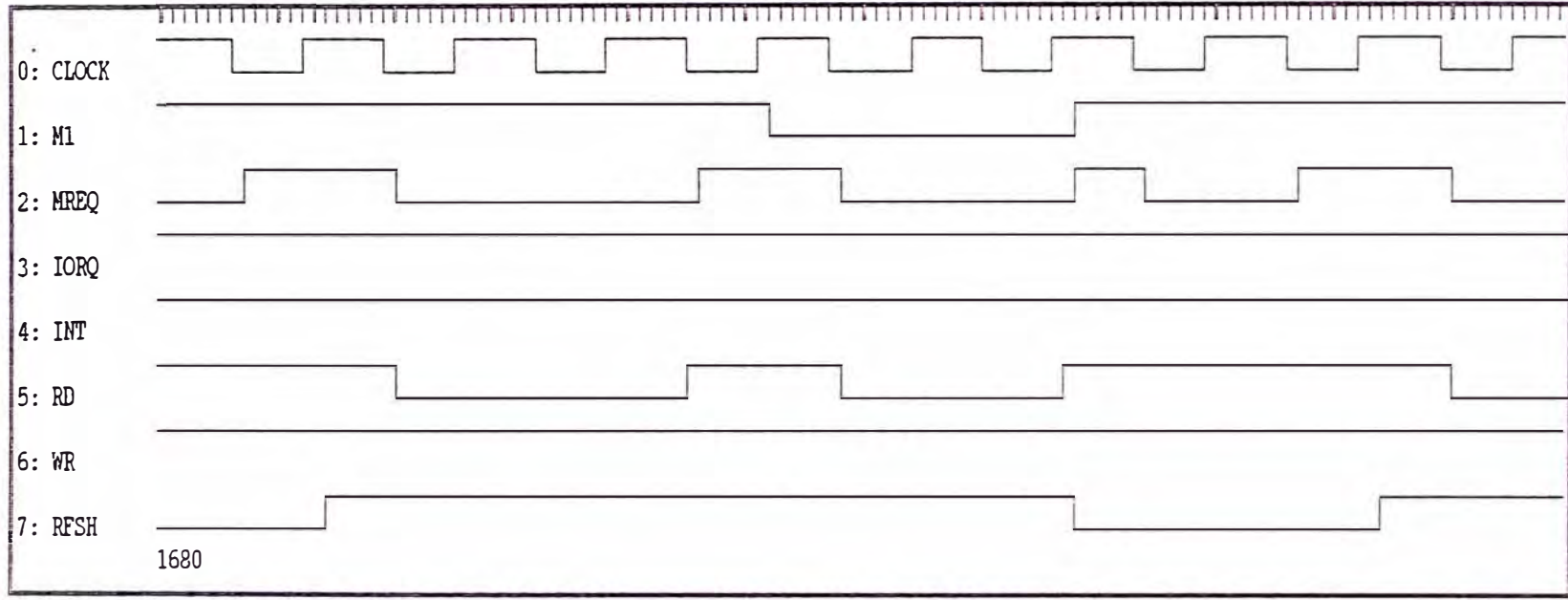
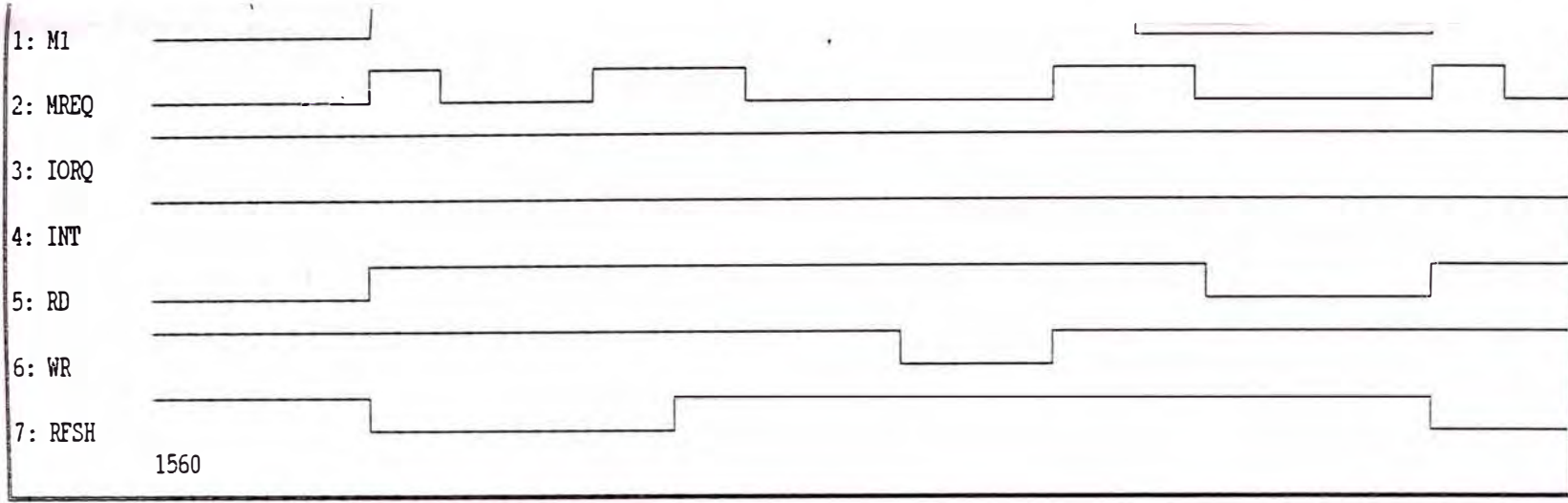


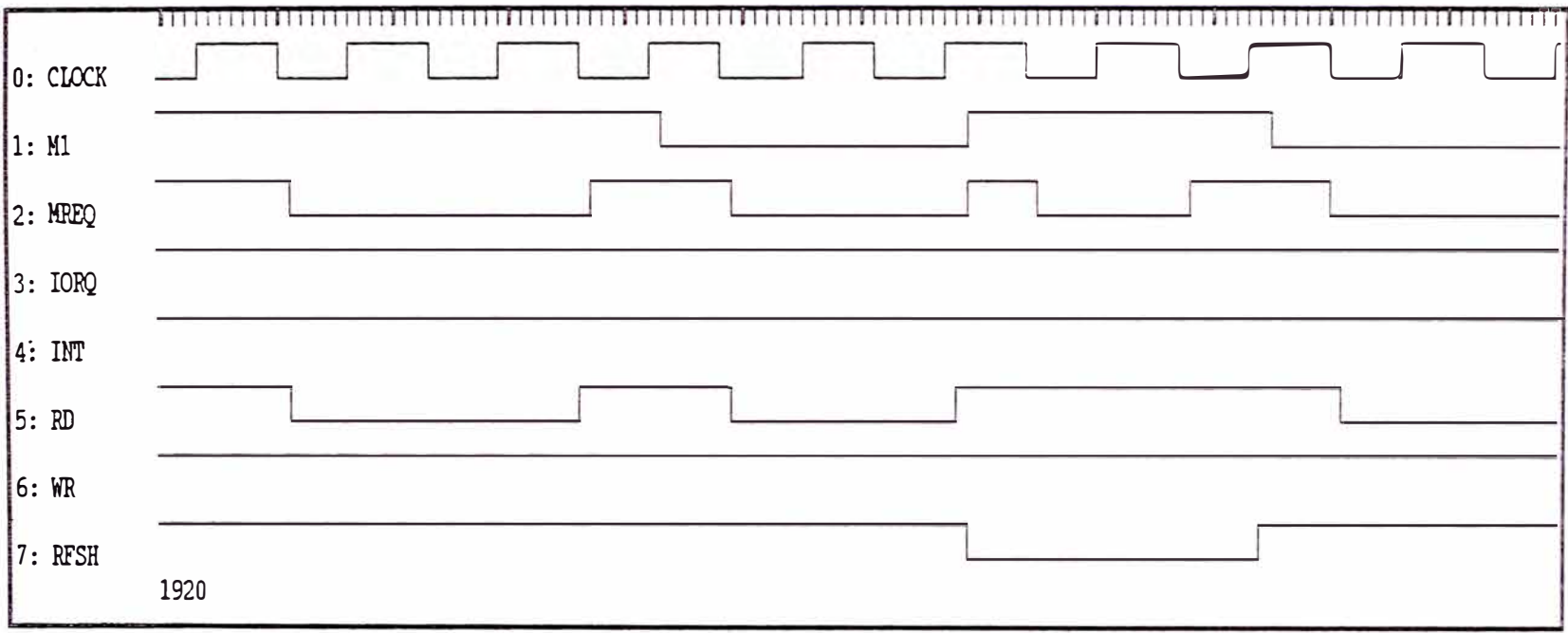
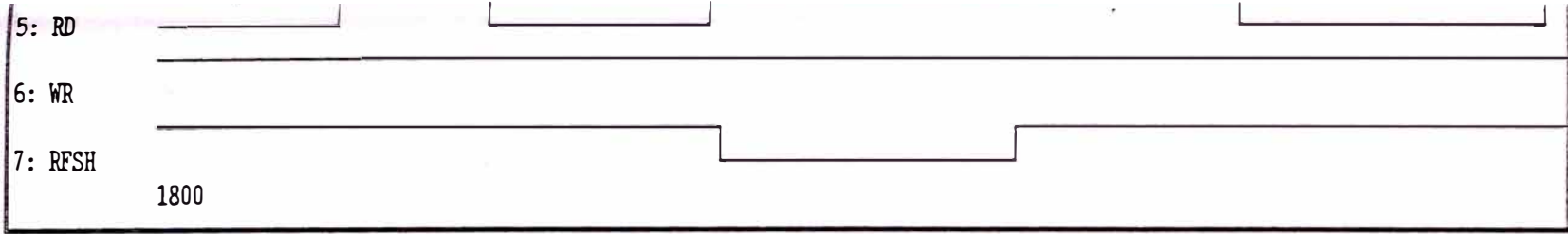








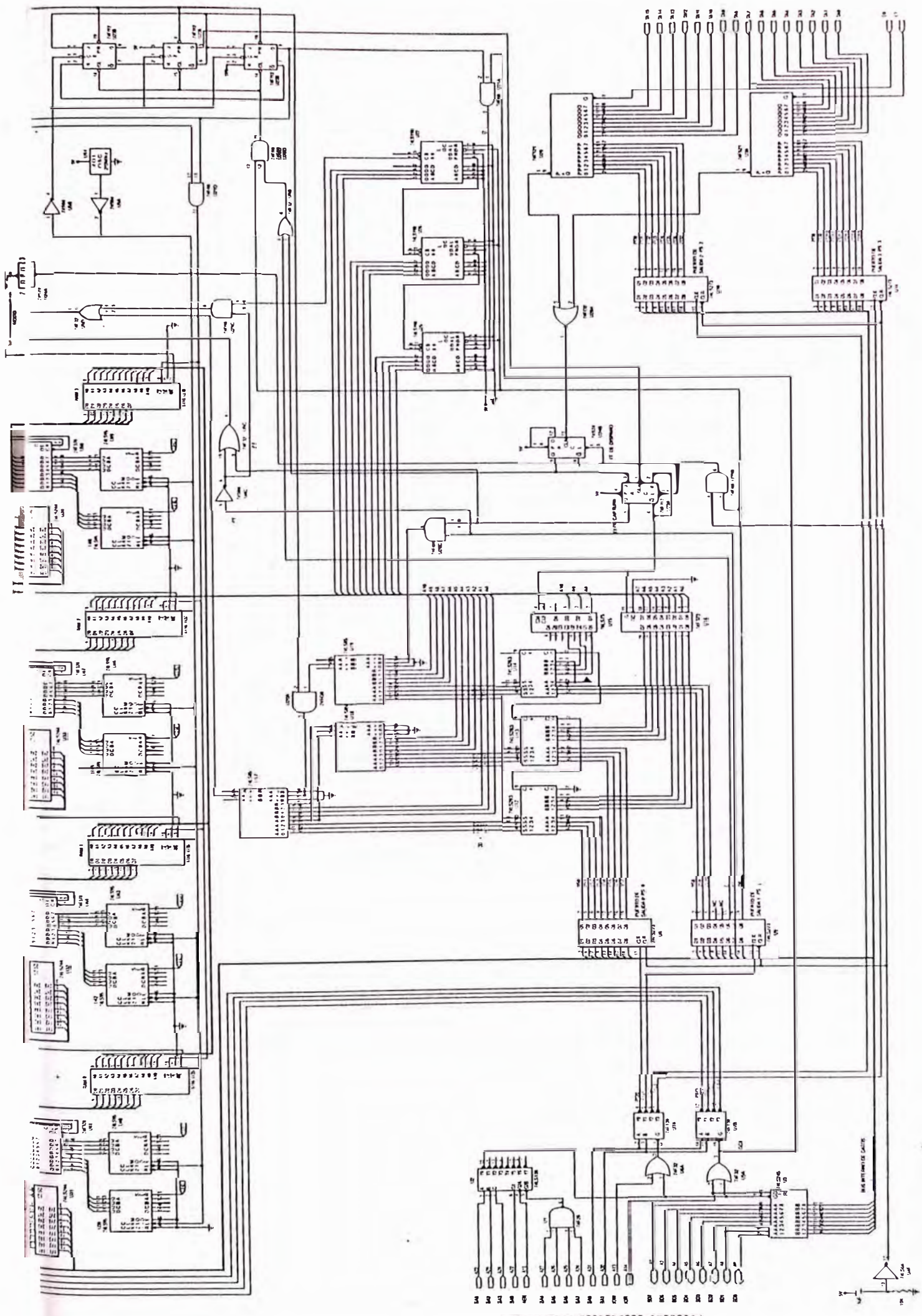




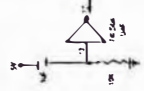
↑

APENDICE D

DIAGRAMA DEL CIRCUITO COMPLETO DEL INTERFAZ



PROCEEDING FROM THE BOARD



APENDICE E

HOJAS DE DATOS DE LOS CIRCUITOS INTEGRADOS USADOS EN EL PROYECTO

FAST 74F02

Gate

Quad Two-Input NOR Gate

Products

Product Specification

FUNCTION TABLE

INPUTS		OUTPUT
D _{nb}	D _{na}	\bar{Q}_n
L	L	H
H	L	L
L	H	L
H	H	L

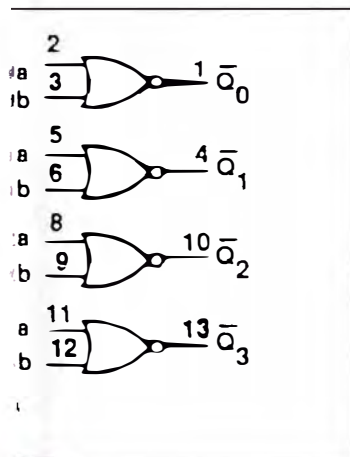
voltage level
 voltage level

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F02	3.4 ns	4.4 mA

ORDERING INFORMATION

PACKAGES	COMMERCIAL RANGE V _{CC} = 5V ± 10%; T _A = 0°C to +70°C
14-Pin Plastic DIP	N74F02N
14-Pin Plastic SO	N74F02D

DIAGRAM



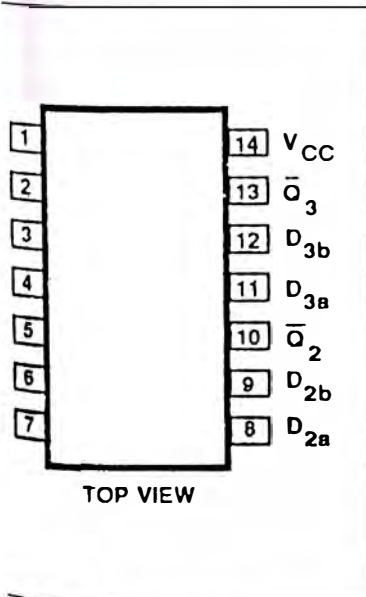
INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74F(U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
D _{na} , D _{nb}	Data inputs	10/10	20µA/0.6mA
\bar{Q}_n	Data output	50/33	1.0mA/20mA

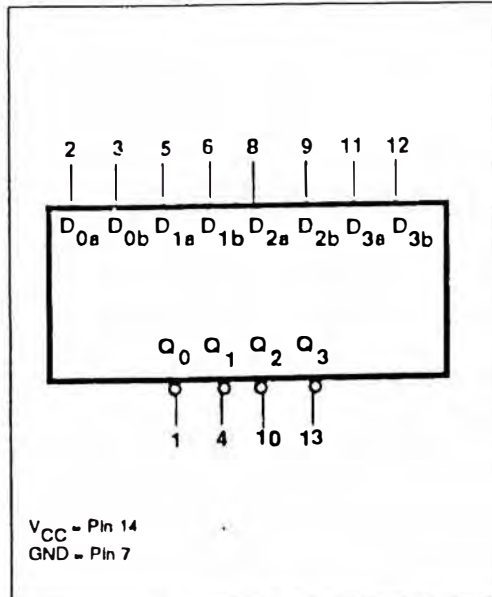
NOTE:

One (1.0) FAST Unit Load is defined as: 20µA in the High state and 0.6mA in the Low state

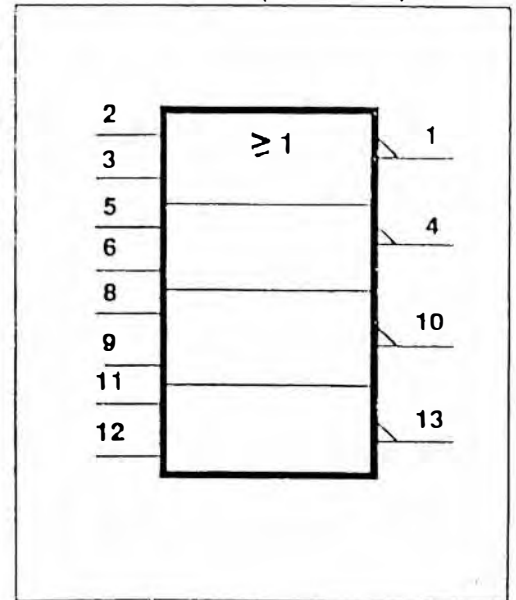
CONFIGURATION



LOGIC SYMBOL



LOGIC SYMBOL (IEEE/IEC)



ABSOLUTE MAXIMUM RATINGS (Operation beyond the limits set forth in this table may impair the useful life of the device. Unless otherwise noted these limits are over the operating free-air temperature range.)

PARAMETER	RATING	UNIT
Supply voltage	-0.5 to +7.0	V
Input voltage	-0.5 to +7.0	V
Input current	-30 to +5	mA
Voltage applied to output in High output state	-0.5 to +V _{CC}	V
Current applied to output in Low output state	40	mA
Operating free-air temperature range	0 to +70	°C
Storage temperature	-65 to +150	°C

RECOMMENDED OPERATING CONDITIONS

PARAMETER	LIMITS			UNIT
	Min	Norm	Max	
Supply voltage	4.5	5.0	5.5	V
High-level input voltage	2.0			V
Low-level input voltage			0.8	V
Input clamp current			-18	mA
High-level output current			-1	mA
Low-level output current			20	mA
Operating free-air temperature range	0		70	°C

ELECTRICAL CHARACTERISTICS (Over recommended operating free-air temperature range unless otherwise noted.)

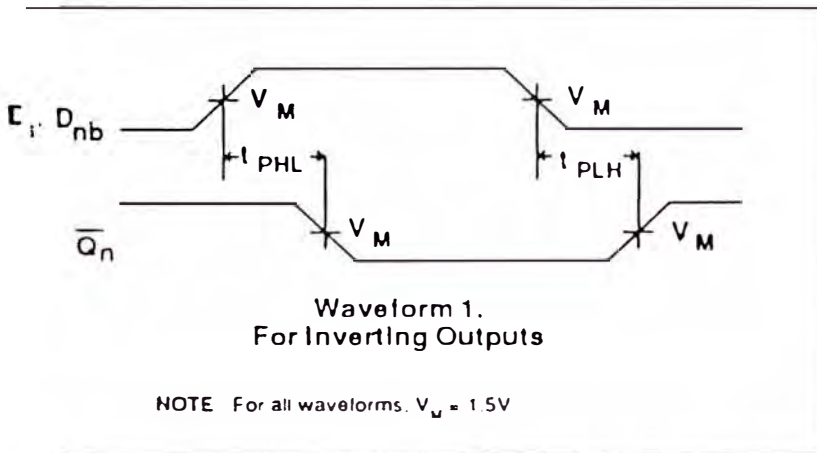
PARAMETER	TEST CONDITIONS ¹	LIMITS			UNIT
		Min	Typ ²	Max	
High-level output voltage	V _{CC} = MIN, V _{IL} = MAX	±10%V _{CC}	2.5		V
	V _{IH} = MIN, I _{OH} = MAX	±5%V _{CC}	2.7	3.4	V
Low-level output voltage	V _{CC} = MIN, V _{IL} = MAX	±10%V _{CC}		0.30 0.50	V
	V _{IH} = MIN, I _{OL} = MAX	±5%V _{CC}		0.30 0.50	V
Input clamp voltage	V _{CC} = MIN, I _I = I _{IK}		-0.73	-1.2	V
Input current at maximum input voltage	V _{CC} = MAX, V _I = 7.0V			100	µA
High-level input current	V _{CC} = MAX, V _I = 2.7V			20	µA
Low-level input current	V _{CC} = MAX, V _I = 0.5V			-0.6	mA
Short circuit output current ³	V _{CC} = MAX		-60	-150	mA
Supply current (total) ⁴	I _{CCH}	V _{CC} = MAX		3.0 5.6	mA
	I _{CCL}			7.0 13.0	mA

NOTES:
 1. Conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.
 2. Typical values are at V_{CC} = 5V, T_A = 25°C.
 3. More than one output should be shorted at a time. For testing I_{OS}, the use of high-speed test apparatus and/or sample-and-hold techniques are preferable in order to minimize internal heating and more accurately reflect operational values. Otherwise, prolonged shorting of a High output may raise the chip temperature above the normal and thereby cause invalid readings in other parameter tests. In any sequence of parameter test, I_{OS} tests should be performed last.
 4. Measured with outputs open.

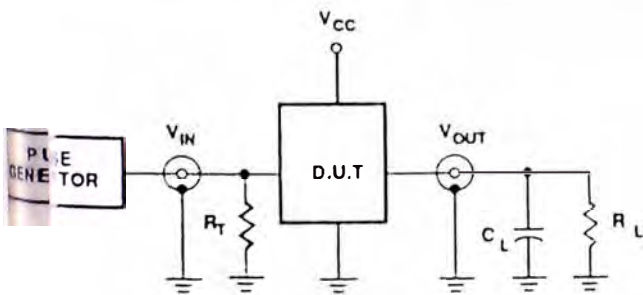
ELECTRICAL CHARACTERISTICS

SYMBOL	PARAMETER	TEST CONDITION	LIMITS					UNIT
			$T_A = +25^\circ\text{C}$ $V_{CC} = 5\text{V}$ $C_L = 50\text{pF}$ $R_L = 500\Omega$			$T_A = 0^\circ\text{C to } +70^\circ\text{C}$ $V_{CC} = 5\text{V} \pm 10\%$ $C_L = 50\text{pF}$ $R_L = 500\Omega$		
			Min	Typ	Max	Min	Max	
t_{pd} D_{na}, D_{nb} to \bar{Q}_n	Propagation delay	Waveform 1	2.5	4.4	5.5	2.5	6.5	ns
			2.0	3.2	4.3	2.0	5.3	

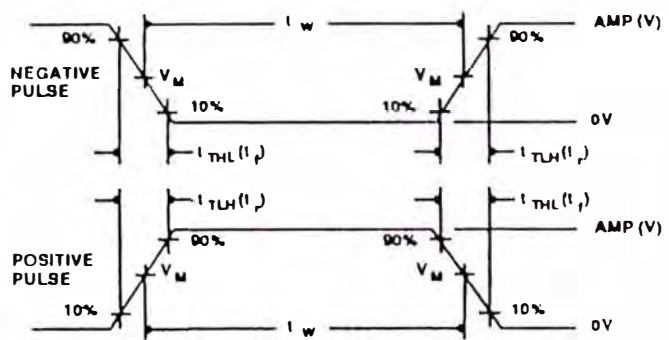
WAVEFORMS



CIRCUIT AND WAVEFORMS



Test Circuit For Totem-Pole Outputs



$V_M = 1.5\text{V}$
Input Pulse Definition

DEFINITIONS

- R_T = load resistor; see AC CHARACTERISTICS for value.
- C_L = load capacitance includes jig and probe capacitance; see AC CHARACTERISTICS for value.
- R_T = termination resistance should be equal to Z_{OUT} of pulse generators.

FAMILY	INPUT PULSE REQUIREMENTS				
	Amplitude	Rep. Rate	t_w	t_{TLH}	t_{THL}
74F	3.0V	1MHz	500ns	2.5ns	2.5ns

Sylvania ECG[®] Semiconductors

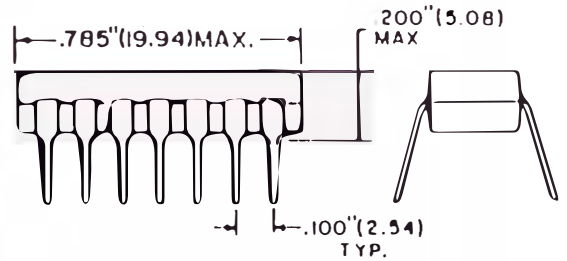
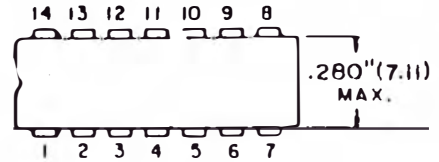
ECG7404, 74H04, 74LS04 74S04

Hex Inverter

Operating Conditions

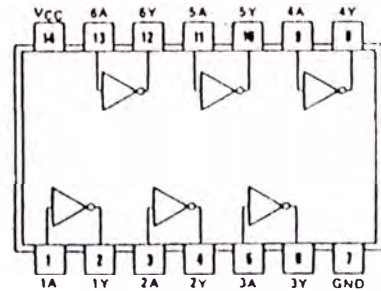
RATINGS	74 SERIES	74H SERIES	74LS SERIES		74S SERIES	UNITS
			DIODE INPUTS	EMITTER INPUTS		
Maximum Allowable Supply Voltage	7	7	7	7	7	V
Minimum Operating Supply Voltage Range	4.75 to 5.25					V
Maximum Input Voltage	5.5	5.5	7	5.5	5.5	V
Maximum Voltage to Open Circuit Outputs*	7	7	7	7	7	V
Operating Free-Air Temperature Range	0 to +70					°C
Storage Temperature Range	-65 to +150					°C

*All selected high voltage types, as specified in electrical tables.



Supply Currents

DEVICE	I _{CCH} (mA) Total With Outputs High		I _{CCL} (mA) Total With Outputs Low	
	TYP	MAX	TYP	MAX
EC74	6	12	18	33
H04	16	26	40	58
LS04	1.2	2.4	3.6	6.6
S04	15	24	30	54



Timing Characteristics at V_{CC} = 5V, T_A = 25°C

DEVICE	CONDITIONS	t _{PLH} (ns) Propagation Delay Time, Low-To-High Level Output			t _{PHL} (ns) Propagation Delay Time, High-To-Low Level Output		
		MIN	TYP	MAX	MIN	TYP	MAX
		04	C _L = 15 pF, R _L = 400Ω		12	22	8
H04	C _L = 25 pF, R _L = 280Ω		6	10	6.5	10	
LS04	C _L = 15 pF, R _L = 2 kΩ		9	15	10	15	
S04	C _L = 15 pF, R _L = 280Ω	2	3	4.5	2	3	5

PARAMETER		CONDITIONS		ECG74												UNITS			
				04			H04			LS04			S04						
				MIN	TYP(1)	MAX	MIN	TYP(1)	MAX	MIN	TYP(1)	MAX	MIN	TYP(1)	MAX				
V_{IH}	High Level Input Voltage			2			2			2			2			V			
V_{IL}	Low Level Input Voltage					0.8			0.8			0.8			0.8	V			
$-V_I$	Input Clamp Voltage	$V_{CC} = \text{Min}$	$I_I = -8 \text{ mA}$						-1.5							V			
			$I_I = -12 \text{ mA}$			-1.5													
			$I_I = -18 \text{ mA}$							-1.5			-1.2						
I_{OH}	High Level Output Current					-400			-500			-400			-1000	μA			
V_{OH}	High Level Output Voltage	$V_{CC} = \text{Min}, V_{IL} = \text{Max}$ $I_{OH} = \text{Max}$		2.4	3.4		2.4	3.5		2.7	3.4		2.7	3.4		V			
I_{OL}	Low Level Output Current					16			20			4			N/A	mA			
						16			20			8			20				
V_{OL}	Low Level Output Voltage	$V_{CC} = \text{Min}$ $V_{IH} = 2\text{V}$	$I_{OL} = \text{Max}$			0.2	0.4		0.2	0.4		0.35	0.5		0.5	V			
			$I_{OL} = 4 \text{ mA}$									0.4							
I_I	Input Current at Maximum Input Voltage	$V_{CC} = \text{Max}$	$V_I = 5.5\text{V}$			1			1						1	mA			
			$V_I = 7\text{V}$								0.1								
I_{IH}	High Level Input Current	$V_{CC} = \text{Max}$	$V_I = 2.4\text{V}$			40			50							μA			
			$V_I = 2.7\text{V}$								20			50					
I_{IL}	Low Level Input Current	$V_{CC} = \text{Max}$	$V_I = 0.3\text{V}$													mA			
			$V_I = 0.4\text{V}$										-0.36						
			$V_I = 0.5\text{V}$					-1.6			-2				-2				
I_{OS}	Short Circuit Output Current	$V_{CC} = \text{Max}(2)$		-18		-55		-40		-100		-30		-130		-40		-100	mA
I_{CC}	Supply Current	$V_{CC} = \text{Max}$		See Table															

Notes

- (1) All typical values are at $V_{CC} = 5\text{V}$, $T_A = 25^\circ\text{C}$.
- (2) Not more than one output should be shorted at a time, and for ECG74H, ECG74LS and ECG74S, duration of a short circuit should not exceed one second.

FAST 74F08

Gate

Quad Two-Input AND Gate

IS Products

Product Specification

FUNCTION TABLE

INPUTS		OUTPUT
D_{na}	D_{nb}	Q_n
L	L	L
L	H	L
H	L	L
H	H	H

L = high voltage level
/ = low voltage level

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F08	4.1 ns	7.1 mA

ORDERING INFORMATION

PACKAGES	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$; $T_A = 0^\circ C$ to $+70^\circ C$
14-Pin Plastic DIP	N74F08N
14-Pin Plastic SO	N74F08D

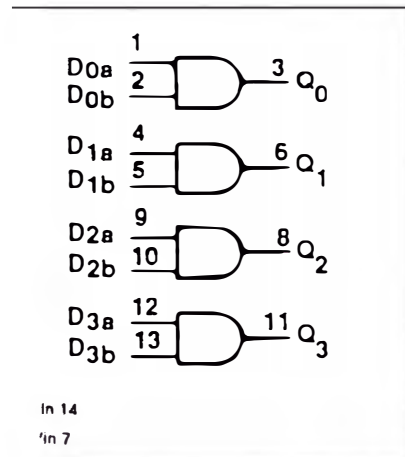
INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74F(U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
D_{na}, D_{nb}	Data inputs	1.0/1.0	20 μ A/0.6mA
Q_n	Data output	50:33	1.0mA/20mA

NOTE:

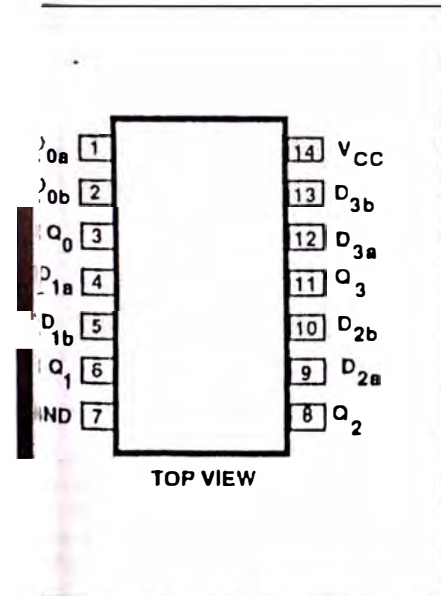
One (1.0) FAST Unit Load is defined as: 20 μ A in the High state and 0.6mA in the Low state.

DIAGRAM

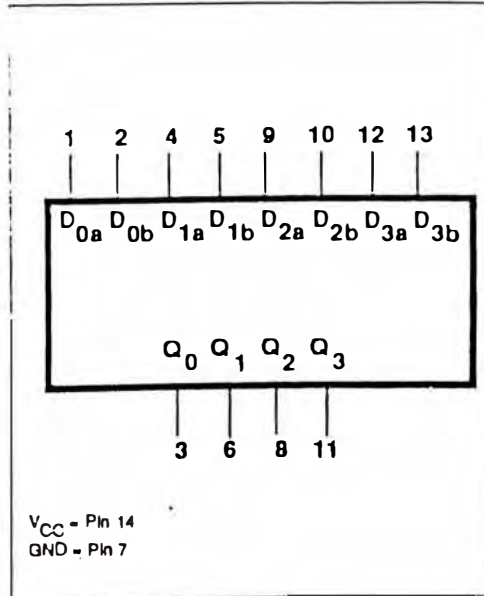


In 14
Pin 7

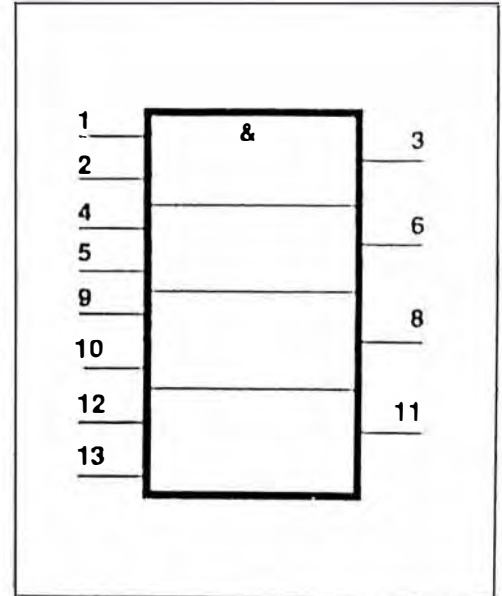
CONFIGURATION



LOGIC SYMBOL



LOGIC SYMBOL (IEEE/IEC)



G te

FAST 74F08

ABSOLUTE MAXIMUM RATINGS (Operation beyond the limits set forth in this table may impair the useful life of the device. Unless otherwise noted these limits are over the operating free-air temperature range.)

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Supply voltage	-0.5 to +7.0	V
V_I	Input voltage	-0.5 to +7.0	V
I_I	Input current	-30 to +5	mA
V_{OC}	Voltage applied to output in High output state	-0.5 to + V_{CC}	V
I_C	Current applied to output in Low output state	40	mA
T_A	Operating free-air temperature range	0 to +70	°C
T_{STG}	Storage temperature	-65 to +150	°C

RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	LIMITS			UNIT
		Min	Nom	Max	
V_{CC}	Supply voltage	4.5	5.0	5.5	V
V_I	High-level input voltage	2.0			V
V_I	Low-level input voltage			0.8	V
I_{IB}	Input clamp current			-18	mA
I_C	High-level output current			-1	mA
I_C	Low-level output current			20	mA
T_A	Operating free-air temperature range	0		70	°C

DC ELECTRICAL CHARACTERISTICS (Over recommended operating free-air temperature range unless otherwise noted.)

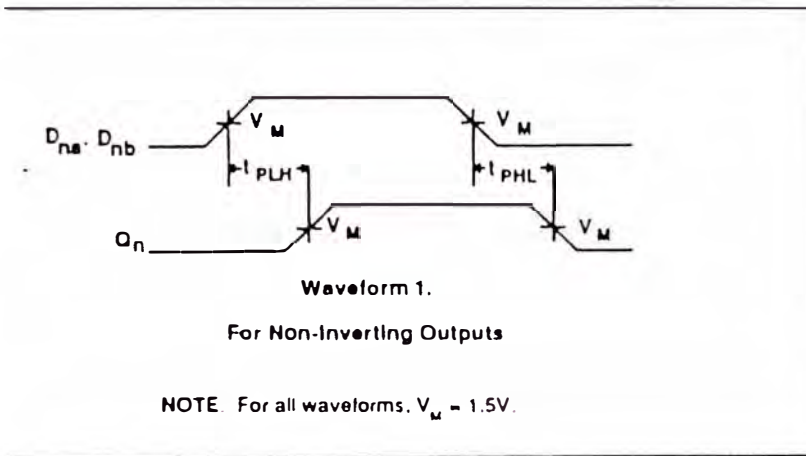
SYMBOL	PARAMETER	TEST CONDITIONS ¹	LIMITS			UNIT	
			Min	Typ ²	Max		
V_O	High-level output voltage	$V_{CC} = \text{MIN}, V_{IL} = \text{MAX}$	$\pm 10\%V_{CC}$	2.5		V	
		$V_{IH} = \text{MIN}, I_{OH} = \text{MAX}$	$\pm 5\%V_{CC}$	2.7	3.4	V	
V_O	Low-level output voltage	$V_{CC} = \text{MIN}, V_{IL} = \text{MAX}$	$\pm 10\%V_{CC}$		0.30	0.50	V
		$V_{IH} = \text{MIN}, I_{OL} = \text{MAX}$	$\pm 5\%V_{CC}$		0.30	0.50	V
V_{IC}	Input clamp voltage	$V_{CC} = \text{MIN}, I_I = I_{IK}$		-0.73	-1.2	V	
I_I	Input current at maximum input voltage	$V_{CC} = \text{MAX}, V_I = 7.0\text{V}$			100	μA	
I_I	High-level input current	$V_{CC} = \text{MAX}, V_I = 2.7\text{V}$			20	μA	
I_I	Low-level input current	$V_{CC} = \text{MAX}, V_I = 0.5\text{V}$			-0.6	mA	
I_C	Short circuit output current ³	$V_{CC} = \text{MAX}$		-60	-150	mA	
I_{CC}	Supply current (total)	$V_{CC} = \text{MAX}$	$V_{IN} = 4.5\text{V}$		5.5	8.3	mA
			$V_{IN} = \text{GND}$		8.6	12.9	mA

¹ Conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type. Typical values are at $V_{CC} = 5\text{V}, T_A = 25^\circ\text{C}$.
² More than one output should be shorted at a time. For testing I_{OS} , the use of high-speed test apparatus and/or sample-and-hold techniques are preferable in order to minimize internal heating and more accurately reflect operational values. Otherwise, prolonged shorting of a High output may raise the chip temperature above normal and thereby cause invalid readings in other parameter tests. In any sequence of parameter tests, I_{OS} tests should be performed last.

ELECTRICAL CHARACTERISTICS

PARAMETER	TEST CONDITION	LIMITS					UNIT
		$T_A = +25^\circ\text{C}$ $V_{CC} = 5\text{V}$ $C_L = 50\text{pF}$ $R_L = 500\Omega$			$T_A = 0^\circ\text{C to } +70^\circ\text{C}$ $V_{CC} = 5\text{V} \pm 10\%$ $C_L = 50\text{pF}$ $R_L = 500\Omega$		
		Min	Typ	Max	Min	Max	
Propagation delay D_{na}, D_{nb} to Q_n	Waveform 1	3.0 2.5	4.2 4.0	5.6 5.3	3.0 2.5	6.6 6.3	ns

WAVEFORMS



CIRCUIT AND WAVEFORMS

Test Circuit For Totem-Pole Outputs

$V_M = 1.5\text{V}$
Input Pulse Definition

DEFINITIONS

- = Load resistor; see AC CHARACTERISTICS for value.
- = Load capacitance includes jig and probe capacitance; see AC CHARACTERISTICS for value.
- = Termination resistance should be equal to Z_{OUT} of pulse generators.

FAMILY	INPUT PULSE REQUIREMENTS				
	Amplitude	Rep. Rate	t_w	t_{TLH}	t_{THL}
74F	3.0V	1MHz	500ns	2.5ns	2.5ns

FAST 74F20

Gate

Dual 4-Input NAND Gate

ST Products

Product Specification

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F20	3.5 ns	2.2 mA

ORDERING INFORMATION

PACKAGES	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$; $T_A = 0^\circ C$ to $+70^\circ C$
14-Pin Plastic DIP	N74F20N
14-Pin Plastic SO	N74F20D

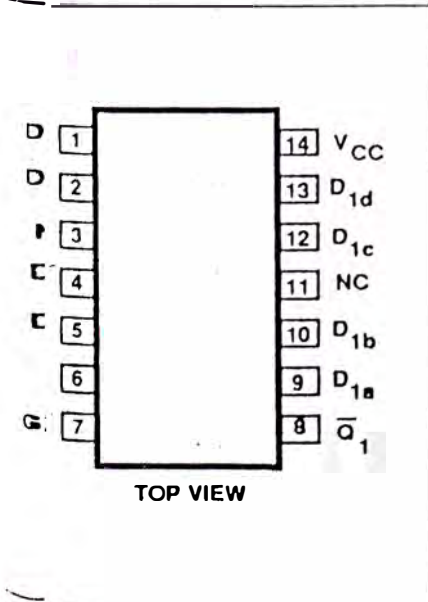
INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74F(U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
$D_{na}, D_{nb}, D_{nc}, D_{nd}$	Data inputs	10/10	20 μ A/0.6mA
\bar{Q}_0, \bar{Q}_1	Data outputs	50/33	10mA/20mA

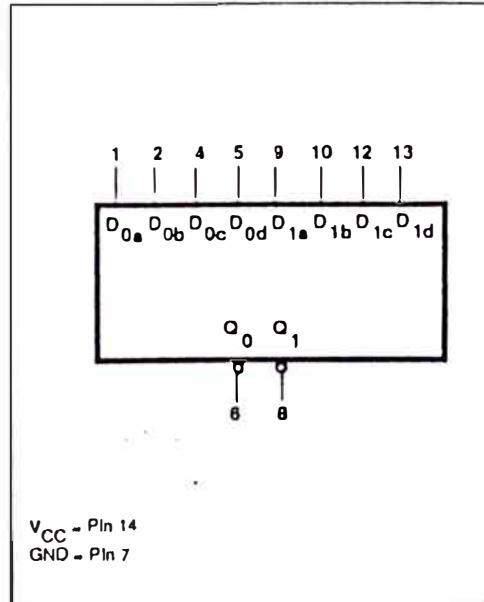
NOTE:

One (1 0) FAST Unit Load is defined as: 20 μ A in the High state and 0.6mA in the Low state.

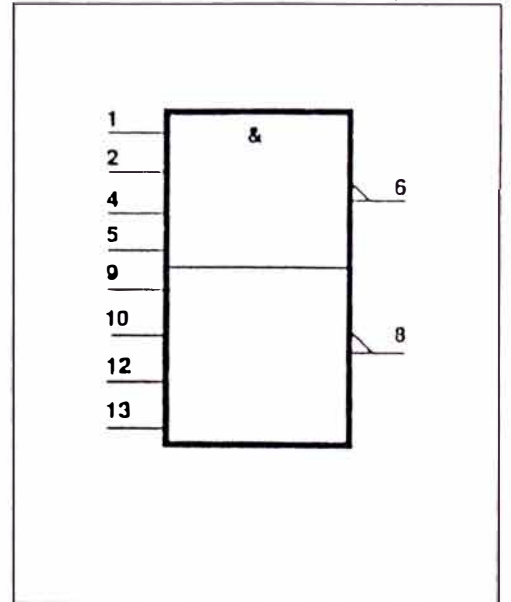
CONFIGURATION

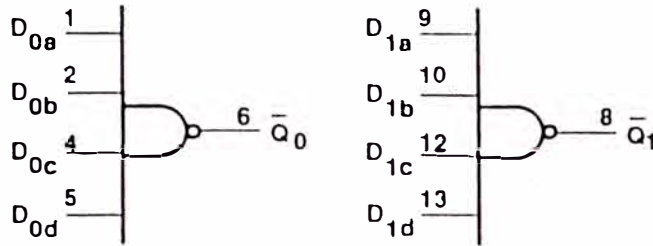


LOGIC SYMBOL



LOGIC SYMBOL (IEEE/IEC)





D = 14
D = 17

FUNCTION TABLE

INPUTS			OUTPUT
D_{nb}	D_{nc}	D_{nd}	\bar{Q}_n
X	X	X	H
L	X	X	H
X	L	X	H
X	X	L	H
H	H	H	L

X: voltage level
L: voltage level
H: voltage level

MAXIMUM RATINGS (Operation beyond the limits set forth in this table may impair the useful life of the device. Unless otherwise noted these limits are over the operating free-air temperature range.)

PARAMETER	RATING	UNIT
Supply voltage	-0.5 to +7.0	V
Input voltage	-0.5 to +7.0	V
Input current	-30 to +5	mA
Voltage applied to output in High output state	-0.5 to + V_{CC}	V
Current applied to output in Low output state	40	mA
Operating free-air temperature range	0 to +70	°C
Storage temperature	-65 to +150	°C

RECOMMENDED OPERATING CONDITIONS

PARAMETER	LIMITS			UNIT
	Min	Nom	Max	
Supply voltage	4.5	5.0	5.5	V
High-level input voltage	2.0			V
Low-level input voltage			0.8	V
Input clamp current			-18	mA
High-level output current			-1	mA
Low-level output current			20	mA
Operating free-air temperature range	0		70	°C

DC ELECTRICAL CHARACTERISTICS (Over recommended operating free-air temperature range unless otherwise noted.)

SYMBOL	PARAMETER	TEST CONDITIONS ¹		LIMITS			UNIT
				Min	Typ ²	Max	
V _O	High-level output voltage	V _{CC} = MIN, V _{IL} = MAX	±10%V _{CC}	2.5			V
		V _{IH} = MIN, I _{OH} = MAX	±5%V _{CC}	2.7	3.4		V
V _O	Low-level output voltage	V _{CC} = MIN, V _{IL} = MAX	±10%V _{CC}		0.30	0.50	V
		V _{IH} = MIN, I _{OL} = MAX	±5%V _{CC}		0.30	0.50	V
V _{IH}	Input clamp voltage	V _{CC} = MIN, I _I = I _{IK}			-0.73	-1.2	V
I _I	Input current at maximum input voltage	V _{CC} = MAX, V _I = 7.0V				100	μA
I _{IH}	High-level input current	V _{CC} = MAX, V _I = 2.7V				20	μA
I _{IL}	Low-level input current	V _{CC} = MAX, V _I = 0.5V				-0.6	mA
I _O	Short circuit output current ³	V _{CC} = MAX		-60		-150	mA
I _C	Supply current (total)	V _{CC} = MAX	V _{IN} = GND		0.9	1.4	mA
			V _{IN} = 4.5V		3.4	5.1	mA

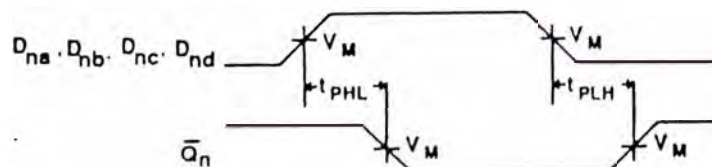
NOTE

- ¹ Conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.
- ² All typical values are at V_{CC} = 5V, T_A = 25°C.
- ³ No more than one output should be shorted at a time. For testing I_{OS}, the use of high-speed test apparatus and/or sample-and-hold techniques are preferable in order to minimize internal heating and more accurately reflect operational values. Otherwise, prolonged shorting of a High output may raise the chip temperature well above normal and thereby cause invalid readings in other parameter tests. In any sequence of parameter tests, I_{OS} tests should be performed last.

AC ELECTRICAL CHARACTERISTICS

SYMBOL	PARAMETER	TEST CONDITION	LIMITS					UNIT
			T _A = +25°C V _{CC} = 5V C _L = 50pF R _L = 500Ω			T _A = 0°C to +70°C V _{CC} = 5V ±10% C _L = 50pF R _L = 500Ω		
			Min	Typ	Max	Min	Max	
t _H t _L	Propagation delay D _{na} , D _{nb} , D _{nc} , D _{nd} to \bar{Q}_n	Waveform 1	2.4 2.0	3.7 3.2	5.0 4.3	2.4 2.0	6.0 5.3	ns

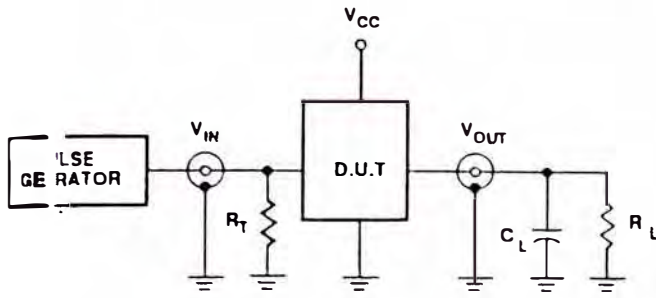
AC WAVEFORMS



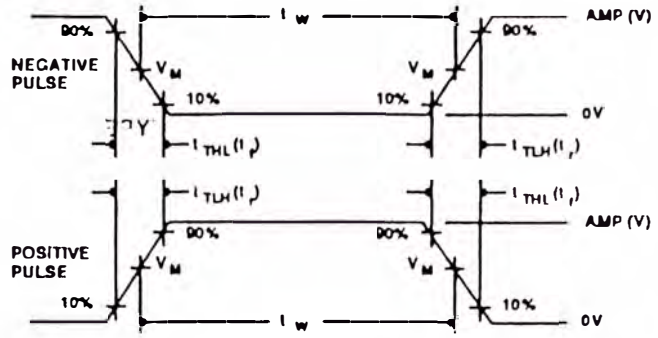
Waveform 1. For Inverting Outputs

NOTE: For all waveforms, V_M = 1.5V.

TEST CIRCUIT AND WAVEFORMS



Test Circuit For Totem-Pole Outputs



$V_M = 1.5V$
Input Pulse Definition

DEFINITIONS

- R_L Load resistor; see AC CHARACTERISTICS for value.
- C_L Load capacitance includes jig and probe capacitance; see AC CHARACTERISTICS for value.
- R_T Termination resistance should be equal to Z_{OUT} of pulse generators.

FAMILY	INPUT PULSE REQUIREMENTS				
	Amplitude	Rep. Rate	t_w	$t_{TLH}(I_p)$	t_{THL}
74F	3.0V	1MHz	500ns	2.5ns	2.5ns

FAST 74F32

Gate

Quad Two-Input OR Gate

FAST Products

Product Specification

FUNCTION TABLE

INPUTS		OUTPUT
D _{nb}	D _{na}	Q _n
L	L	L
H	L	H
L	H	H
H	H	H

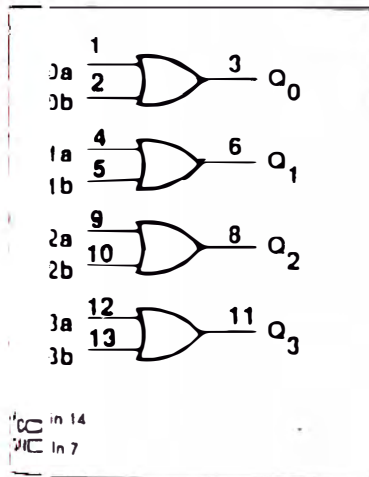
H = high voltage level
L = low voltage level

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F32	4.1 ns	8.2 mA

ORDERING INFORMATION

PACKAGES	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$; $T_A = 0^\circ C$ to $+70^\circ C$
14-Pin Plastic DIP	N74F32N
14-Pin Plastic SO	N74F32D

LOGIC DIAGRAM



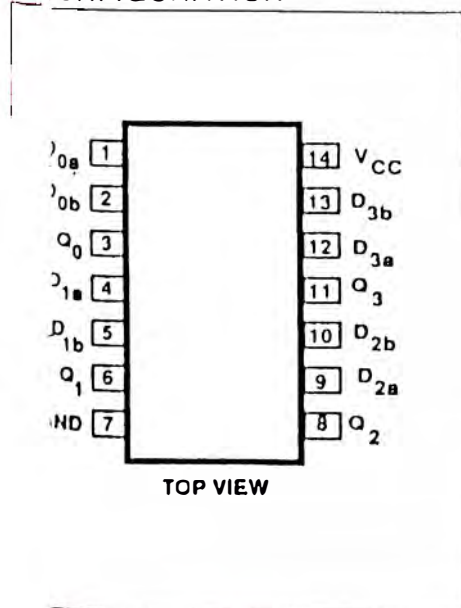
INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74F(U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
D _{na} , D _{nb}	Data inputs	1.0/1.0	20 μ A/0.6mA
Q _n	Data output	50/33	1.0mA/20mA

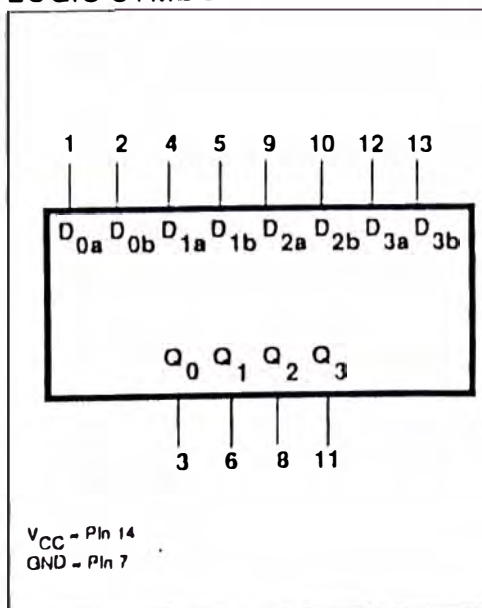
NOTE:

One (1.0) FAST Unit Load is defined as: 20 μ A in the High state and 0.6mA in the Low state.

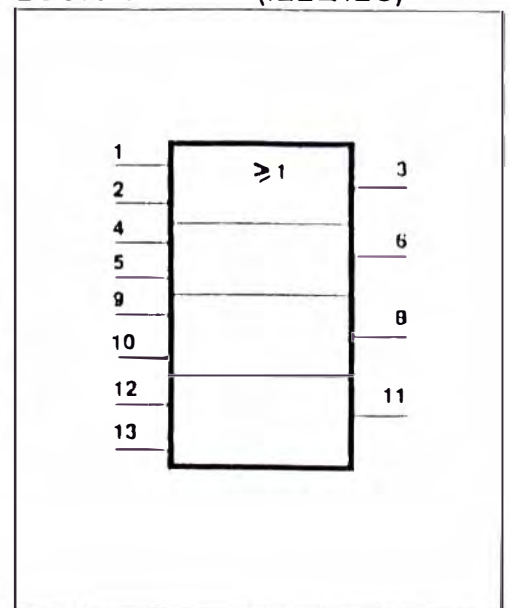
PACKAGE CONFIGURATION



LOGIC SYMBOL



LOGIC SYMBOL (IEEE/IEC)



ABSOLUTE MAXIMUM RATINGS (Operation beyond the limits set forth in this table may impair the useful life of the device. Unless otherwise noted these limits are over the operating free-air temperature range.)

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Supply voltage	-0.5 to +7.0	V
V_{IH}	Input voltage	-0.5 to +7.0	V
I_{IN}	Input current	-30 to +5	mA
V_{OH}	Voltage applied to output in High output state	-0.5 to + V_{CC}	V
I_{OL}	Current applied to output in Low output state	40	mA
T_A	Operating free-air temperature range	0 to +70	°C
T_S	Storage temperature	-65 to +150	°C

RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	LIMITS			UNIT
		Min	Nom	Max	
V_{CC}	Supply voltage	4.5	5.0	5.5	V
V_{IH}	High-level input voltage	2.0			V
V_{IL}	Low-level input voltage			0.8	V
I_{IK}	input clamp current			-18	mA
I_{OH}	High-level output current			-1	mA
I_{OL}	Low-level output current			20	mA
T_A	Operating free-air temperature range	0		70	°C

ELECTRICAL CHARACTERISTICS (Over recommended operating free-air temperature range unless otherwise noted.)

SYMBOL	PARAMETER	TEST CONDITIONS ¹	LIMITS			UNIT
			Min	Typ ²	Max	
V_{OH}	High-level output voltage	$V_{CC} = \text{MIN}, V_{IL} = \text{MAX}$	$\pm 10\% V_{CC}$	2.5		V
		$V_{IH} = \text{MIN}, I_{OH} = \text{MAX}$	$\pm 5\% V_{CC}$	2.7	3.4	V
V_{OL}	Low-level output voltage	$V_{CC} = \text{MIN}, V_{IL} = \text{MAX}$	$\pm 10\% V_{CC}$	0.35	0.50	V
		$V_{IH} = \text{MIN}, I_{OL} = \text{MAX}$	$\pm 5\% V_{CC}$	0.35	0.50	V
V_{IF}	input clamp voltage	$V_{CC} = \text{MIN}, I_I = I_{IK}$		-0.73	-1.2	V
I_I	Input current at maximum input voltage	$V_{CC} = \text{MAX}, V_I = 7.0\text{V}$			100	μA
I_{IH}	High-level input current	$V_{CC} = \text{MAX}, V_I = 2.7\text{V}$			20	μA
I_{IL}	Low-level input current	$V_{CC} = \text{MAX}, V_I = 0.5\text{V}$			-0.6	mA
I_{OS}	Short circuit output current ³	$V_{CC} = \text{MAX}$		-60	-150	mA
I_{CC}	Supply current (total)	$V_{CC} = \text{MAX}$	$V_{IN} = 4.5\text{V}$	6.1	9.2	mA
			$V_{IN} = \text{GND}$	10.3	15.5	mA

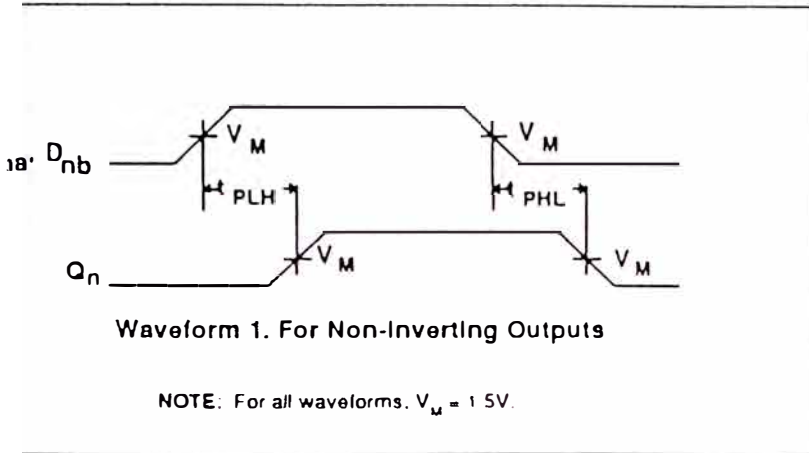
NOTE:

1. Conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.
 2. Typical values are at $V_{CC} = 5\text{V}, T_A = 25^\circ\text{C}$.
 3. More than one output should be shorted at a time. For testing I_{OS} , the use of high-speed test apparatus and/or sample-and-hold techniques are preferable in order to minimize internal heating and more accurately reflect operational values. Otherwise, prolonged shorting of a High output may raise the chip temperature above normal and thereby cause invalid readings in other parameter tests. In any sequence of parameter test, I_{OS} tests should be performed last.

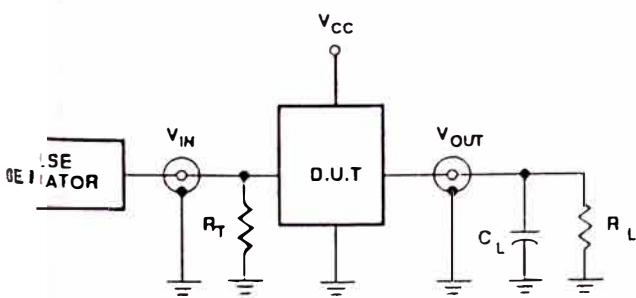
AC ELECTRICAL CHARACTERISTICS

PARAMETER	TEST CONDITION	LIMITS					UNIT
		$T_A = +25^\circ\text{C}$ $V_{CC} = 5\text{V}$ $C_L = 50\text{pF}$ $R_L = 500\Omega$			$T_A = 0^\circ\text{C to } +70^\circ\text{C}$ $V_{CC} = 5\text{V} \pm 10\%$ $C_L = 50\text{pF}$ $R_L = 500\Omega$		
		Min	Typ	Max	Min	Max	
Propagation delay D_{na}, D_{nb} to Q_n	Waveform 1	3.0	4.2	5.6	3.0	6.6	ns
		3.0	4.0	5.3	3.0	6.3	

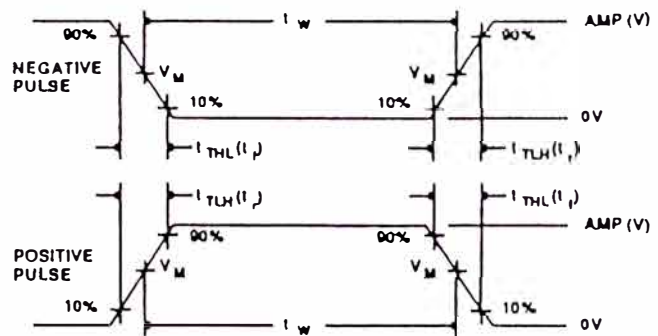
WAVEFORMS



TEST CIRCUIT AND WAVEFORMS



Test Circuit For Totem-Pole Outputs



$V_M = 1.5\text{V}$

Input Pulse Definition

DEFINITIONS

- R_L Load resistor; see AC CHARACTERISTICS for value.
- C_L Load capacitance includes jig and probe capacitance; see AC CHARACTERISTICS for value.
- R_T Termination resistance should be equal to Z_{OUT} of pulse generators.

FAMILY	INPUT PULSE REQUIREMENTS				
	Amplitude	Rep. Rate	t_w	t_{TLH}	t_{THL}
74F	3.0V	1MHz	500ns	2.5ns	2.5ns

Sylvania ECG[®] Semiconductors

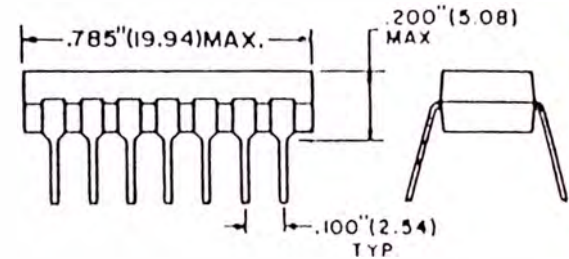
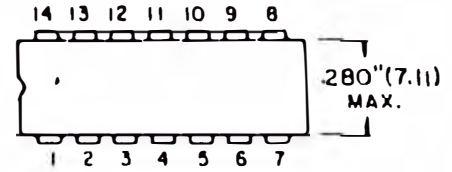
ECG74S74, 74S112, 74S113, 74S114

Flip/Flops

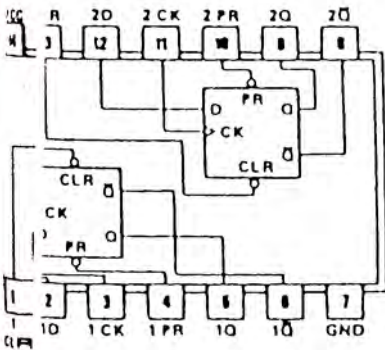
Ratings/Operating Conditions

RATINGS	74 SERIES	74H SERIES	74LS SERIES		74S SERIES	UNITS
			DIODE INPUTS	EMITTER INPUTS		
Maximum Allowable Supply Voltage	7	7	7	7	7	V
Minimum Operating Supply Voltage Range	4.75 to 5.25					V
Input Voltage	5.5	5.5	7	5.5	5.5	V
Minimum Voltage to Open Multiple Outputs*	7	7	7	7	7	V
Maximum Free-Air Operating Temperature Range	0 to +70					°C
Storage Temperature Range	-65 to +150					°C

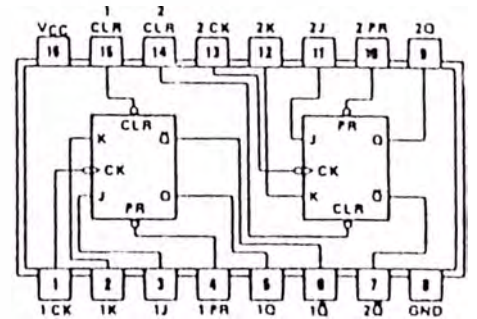
*For selected high voltage types, as specified in electrical tables



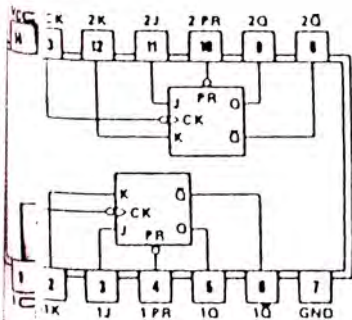
ECG74S74-"D-Type" Edge Triggered



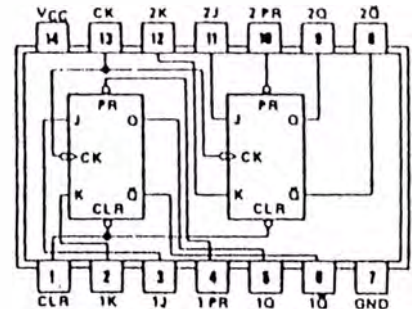
ECG74S112-J-K Negative Edge Triggered



ECG74S113-J-K Negative Edge Triggered



ECG74S114-J-K Negative Edge Triggered, Common Clear and Common Clock



Electrical Characteristics over recommended operating free-air temperature range (unless otherwise noted).

PARAMETER		CONDITIONS	ECG74												UNITS
			S74			S112			S113			S114			
			MIN	TYP(1)	MAX	MIN	TYP(1)	MAX	MIN	TYP(1)	MAX	MIN	TYP(1)	MAX	
V_{IH}	High Level Input Voltage		2			2			2			2			V
V_{IL}	Low Level Input Voltage		0.8			0.8			0.8			0.8			V
V_I	Input Clamp Voltage	$V_{CC} = \text{Min}, I_I = -18 \text{ mA}$	-1.2			-1.2			-1.2			-1.2			V
I_{OH}	High Level Output Current		-1			-1			-1			-1			mA
V_{OH}	High Level Output Voltage	$V_{CC} = \text{Min}, V_{IH} = 2\text{V}$ $V_{IL} = 0.8\text{V}, I_{OH} = -1 \text{ mA}$	2.7	3.4		2.7	3.4		2.7	3.4		2.7	3.4		V
I_{OL}	Low Level Output Current		20			20			20			20			mA
V_{OL}	Low Level Output Voltage	$V_{CC} = \text{Min}, V_{IH} = 2\text{V}$ $V_{IL} = 0.8\text{V}, I_{OL} = 20 \text{ mA}$	0.5			0.5			0.5			0.5			V
I_I	Input Current at Maximum Input Voltage	$V_{CC} = \text{Max}, V_I = 5.5\text{V}$	1			1			1			1			mA
I_{IH}	High Level Input Current	J, K, or D	50			50			50			50			μA
		Clear	150			100			N/A			200			
		Preset	100			100			100			100			
		Clock	100			100			100			200			
I_{IL}	Low Level Input Current	J, K, or D	-2			-1.6			-1.6			-1.6			mA
		Clear	-6			-7			N/A			-14			
		Preset	-4			-7			-7			-7			
		Clock	-4			-4			-4			-8			
I_{OS}	Short Circuit Output Current	$V_{CC} = \text{Max}(2)$	-40		-100		-40		-100		-40		-100		mA
I_{CC}	Supply Current	$V_{CC} = \text{Max}(3)$	30		50		30		50		30		50		mA

Notes

- (1) All typical values are at $V_{CC} = 5\text{V}, T_A = 25^\circ\text{C}$.
- (2) Not more than one output should be shorted at a time, and duration of short circuit should not exceed one second.
- (3) With all outputs open, I_{CC} is measured with the Q and \bar{Q} outputs high in turn. At the time of measurement, the clock input is grounded.

Switching Characteristics at $V_{CC} = 5V$, $T_A = 25^\circ C$

PARAMETER	FROM (INPUT)	TO (OUTPUT)	CONDITIONS	ECG74						UNITS
				S74			S112, S113, S114			
				MIN	TYP	MAX	MIN	TYP	MAX	
f_{MAX}	Maximum Clock Frequency			75	110		80	125		MHz
t_{PLH}	Propagation Delay Time, Low-to-High Level Output	Preset or Clear	Q or \bar{Q}		4	6	2	4	7	ns
t_{PHL}	Propagation Delay Time, High-to-Low Level Output	Preset or Clear	(Clock High)	\bar{Q} or Q	9	13.5	2	5	7	ns
			(Clock Low)		5	8	2	5	7	
t_{PLH}	Propagation Delay Time, Low-to-High Level Output	Clock	Q or \bar{Q}		6	9	2	4	7	ns
t_{PHL}	Propagation Delay Time, High-to-Low Level Output				6	9	2	5	7	
t_W	Pulse Width	Clock High		6			6			ns
		Clock Low		7.3			6.5			
		Clear or Preset Low		7			8			
SETUP	Input Setup Time(4)	High Level Data		3 \uparrow			3 \downarrow			ns
		Low Level Data		3 \uparrow			3 \downarrow			
HOLD	Input Hold Time(4)			2 \uparrow			0 \downarrow			ns

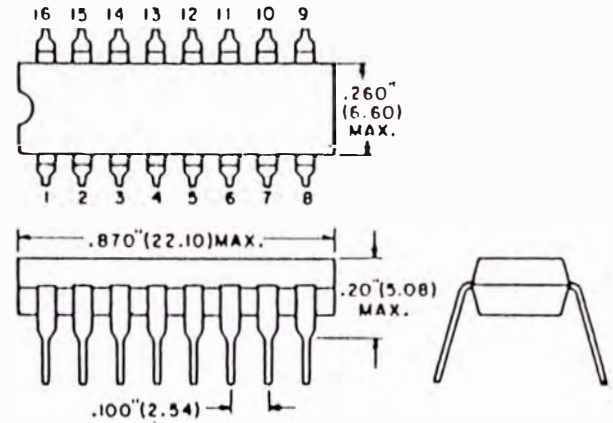
Notes

(4) \uparrow : The arrow indicates the edge of the clock pulse used for reference: \uparrow for the rising edge, \downarrow for the falling edge.

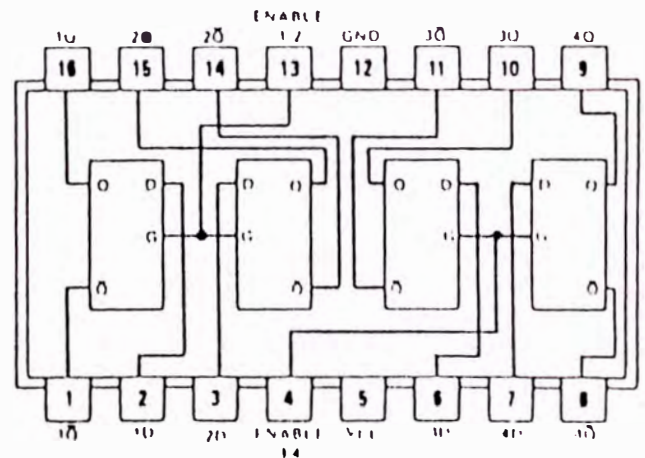
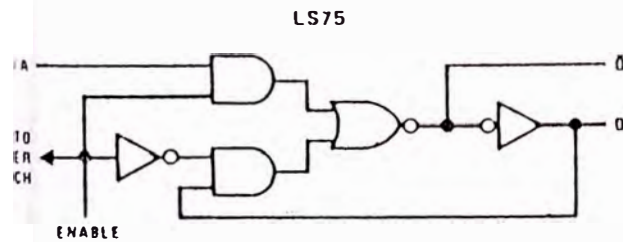
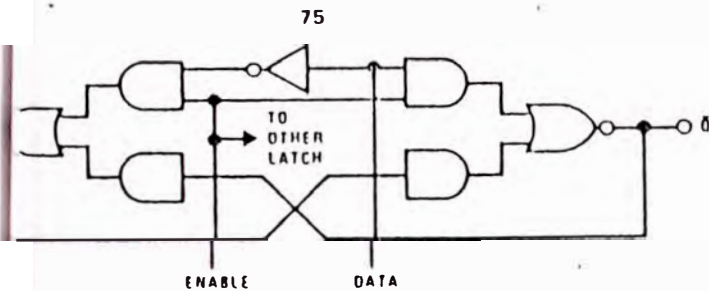
Stable Latch

are ideally suited for use as temporary storage for information between processing units and input/output or information present at a data (D) input is transferred to the Q output when the enable (G) is high, and the Q output will remain at that input as long as the enable remains high. When the enable goes low, the information (that was present at the data input at the time the transition occurred) is retained at the Q output until the enable is permitted to go high again.

The ECG7475 and ECG74LS75 feature complementary Q and \bar{Q} outputs and a 1-bit latch.



Logic Diagrams (Each Latch)



Operating Conditions

PARAMETER	74 SERIES	74H SERIES	74LS SERIES		74S SERIES	UNITS
			DIODE INPUTS	EMITTER INPUTS		
Allowable Storage Temperature Range	7	7	7	7	7	V
Operating Temperature Range	4.75 to 5.25					V
Input Voltage	5.5	5.5	7	5.5	5.5	V
Voltage to Open Outputs*	7	7	7	7	7	V
Free-Air Storage Temperature Range	0 to +70					°C
Operating Temperature Range	-65 to +150					°C

*Not for high voltage types, as specified in electrical tables.

Truth Table (Each Latch)

INPUTS		OUTPUTS	
D	G	Q	\bar{Q}
L	H	L	H
H	H	H	L
X	L	Q_0	\bar{Q}_0

H = High Level, L = Low Level, X = Don't Care

Q_0 = The Level of Q Before the High-to-Low Transition of G

PARAMETER		CONDITIONS		ECG74			ECG74			UNITS		
				75			LS75					
				MIN	TYP(1)	MAX	MIN	TYP(1)	MAX			
V_{IH}	High Level Input Voltage									V		
V_{IL}	Low Level Input Voltage									V		
V_I	Input Clamp Voltage	$V_{CC} = \text{Min}$	$I_I = -12 \text{ mA}$								V	
			$I_I = -18 \text{ mA}$									V
I_{OH}	High Level Output Current									μA		
V_{OH}	High Level Output Voltage	$V_{CC} = \text{Min}, V_{IH} = 2\text{V}$ $V_{IL} = \text{Max}, I_{OH} = \text{Max}$									V	
I_{OL}	Low Level Output Current									mA		
V_{OL}	Low Level Output Voltage	$V_{CC} = \text{Min}, V_{IH} = 2\text{V},$ $V_{IL} = \text{Max}$	$I_{OL} = 4 \text{ mA}$								V	
			$I_{OL} = \text{Max}$									V
I_I	Input Current at Maximum Input Voltage	$V_{CC} = \text{Max}$	$V_I = 5.5\text{V}$	D Input							mA	
				G Input								
			$V_I = 7\text{V}$	D Input								
				G Input								
I_{IH}	High Level Input Current	$V_{CC} = \text{Max}$	$V_I = 2.4\text{V}$	D Input							μA	
				G Input								
			$V_I = 2.7\text{V}$	D Input								
				G Input								
I_{IL}	Low Level Input Current	$V_{CC} = \text{Max}$	$V_I = 0.4\text{V}$	D Input							mA	
				G Input								
I_{OS}	Short Circuit Output Current	$V_{CC} = \text{Max}(2)$									mA	
I_{CC}	Supply Current	$V_{CC} = \text{Max}(3)$									mA	
											LS75	
											75	

Characteristics $V_{CC} = 5V, T_A = 25^\circ C$

METER	FROM (INPUT)	TO (OUTPUT)	ECG74			CONDITIONS	ECG74			UNITS
			75				LS75			
			CONDITIONS	MIN	TYP		MAX	MIN	TYP	
Propagation Delay Time, High Level Output	D	O	$C_L = 15\text{ pF}$ $R_L = 400\Omega$	$C_L = 15\text{ pF}$ $R_L = 2\text{ k}\Omega$		16	30	15	27	ns
						14	25	9	17	
Propagation Delay Time, Low Level Output	D	\bar{O}			24	40	12	20	ns	
					7	15	7	15		
Propagation Delay Time, High Level Output	G	O			16	30	15	27	ns	
					7	15	14	25		
Propagation Delay Time, Low Level Output	G	\bar{O}			16	30	16	30	ns	
					7	15	7	15		
Setup Time of Enabling Pulse					20			20		ns
Hold Time					20			20		ns
Output Delay Time			5			0		ns		

Values are at $V_{CC} = 5V, T_A = 25^\circ C$.

For the ECG74LS duration of short circuit should not exceed one second. Load with all inputs grounded and all outputs open.

Ivania ECG[®] microconductors

ECG7485, 74LS85

Magnitude Comparators

4-bit magnitude comparators perform comparison of binary or BCD codes. Three fully-decoded decisions about words (A, B) are made and are externally available at three separate outputs. These devices are fully expandable to any number of bits using external gates. Words of greater length may be compared by using comparators in cascade. The A₃ = B₃, A₃ > B₃, and A₃ < B₃ outputs of a stage handling less-significant bits are connected to the corresponding inputs of the next stage handling more-significant bits. The stage handling the least-significant bits must have a reference level voltage applied to the A₀ = B₀ input.

The output paths of the 85 and LS85 are implemented with only one gate delay to reduce overall comparison times for long words.

TYPE	TYPICAL POWER DISSIPATION	TYPICAL DELAY (4-BIT WORDS)
85	275 mW	23 ns
LS85	52 mW	24 ns

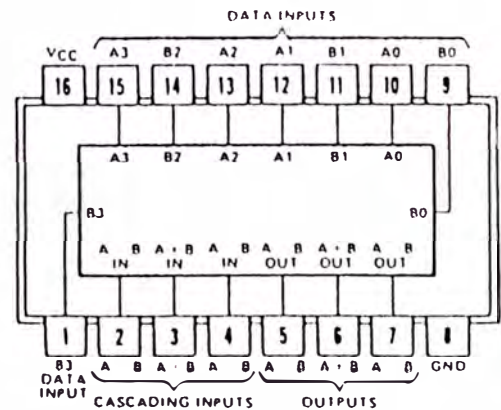
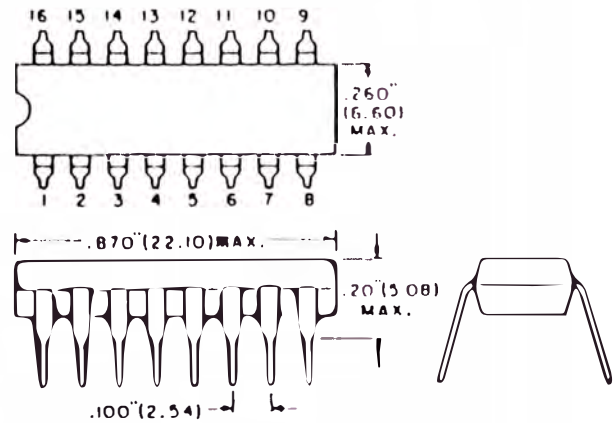
Truth Tables

COMPARING INPUTS			CASCADING INPUTS			OUTPUTS		
A ₂ , B ₂	A ₁ , B ₁	A ₀ , B ₀	A > B	A < B	A = B	A > B	A < B	A = B
X	X	X	X	X	X	H	L	L
X	X	X	X	X	X	L	H	L
A ₂ > B ₂	X	X	X	X	X	H	L	L
A ₂ < B ₂	X	X	X	X	X	L	H	L
A ₂ = B ₂	A ₁ > B ₁	X	X	X	X	H	L	L
A ₂ = B ₂	A ₁ < B ₁	X	X	X	X	L	H	L
A ₂ = B ₂	A ₁ = B ₁	A ₀ > B ₀	X	X	X	H	L	L
A ₂ = B ₂	A ₁ = B ₁	A ₀ < B ₀	X	X	X	L	H	L
A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	H	L	L	H	L	L
A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	L	H	L	L	H	L
A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	L	L	H	L	L	H

85, LS85

A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	X	X	H	L	L	H
A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	H	H	L	L	L	L
A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀	L	L	L	H	H	L

H = High Level, L = Low Level, X = Don't Care



Operating Conditions

PARAMETERS	74 SERIES	74H SERIES	74LS SERIES		74S SERIES	UNITS
			DIODE INPUTS	EMITTER INPUTS		
Power Supply Voltage	7	7	7	7	7	V
Operating Voltage Range	4.75 to 5.25					V
Output Voltage	5.5	5.5	7	5.5	5.5	V
Voltage to Open-Circuit Outputs*	7	7	7	7	7	V
Storage Temperature Range	0 to +70					°C
Operating Temperature Range	-65 to +150					°C

*For high voltage types, as specified in electrical tables.

Electrical Characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		CONDITIONS		ECG74						UNITS	
				85			LS85				
				MIN	TYP(1)	MAX	MIN	TYP(1)	MAX		
V _{IH}	High Level Input Voltage									V	
V _{IL}	Low Level Input Voltage									V	
V _I	Input Clamp Voltage	V _{CC} = Min	I _I = -12 mA							V	
			I _I = -18 mA								
I _{OH}	High Level Output Current									μA	
V _{OH}	High Level Output Voltage	V _{CC} = Min, V _{IH} = 2V V _{IL} = Max, I _{OH} = Max								V	
I _{OL}	Low Level Output Current									mA	
V _{OL}	Low Level Output Voltage	V _{CC} = Min V _{IH} = 2V V _{IL} = Max	I _{OL} = Max							V	
			I _{OL} = 4 mA								
I _I	Input Current at Maximum Input Voltage	V _{CC} = Max	A < B, A > B, All Other Inputs	V _I = 5.5V						mA	
				V _I = 7V							
				V _I = 5.5V							
				V _I = 7V							
I _{IH}	High Level Input Current	V _{CC} = Max	A < B, A > B, All Other Inputs	V _I = 2.4V						μA	
				V _I = 2.7V							
				V _I = 2.4V							
				V _I = 2.7V							
I _{IL}	Low Level Input Current	V _{CC} = Max	A < B, A > B, All Other Inputs	V _I = 0.3V						mA	
				V _I = 0.4V							
				V _I = 0.3V							
				V _I = 0.4V							
I _{OS}	Short Circuit Output Current	V _{CC} = Max(2)							mA		
I _{CC}	Supply Current	V _{CC} = Max	(3)							mA	

Notes

(1) All typical values are at V_{CC} = 5V, T_A = 25°C.

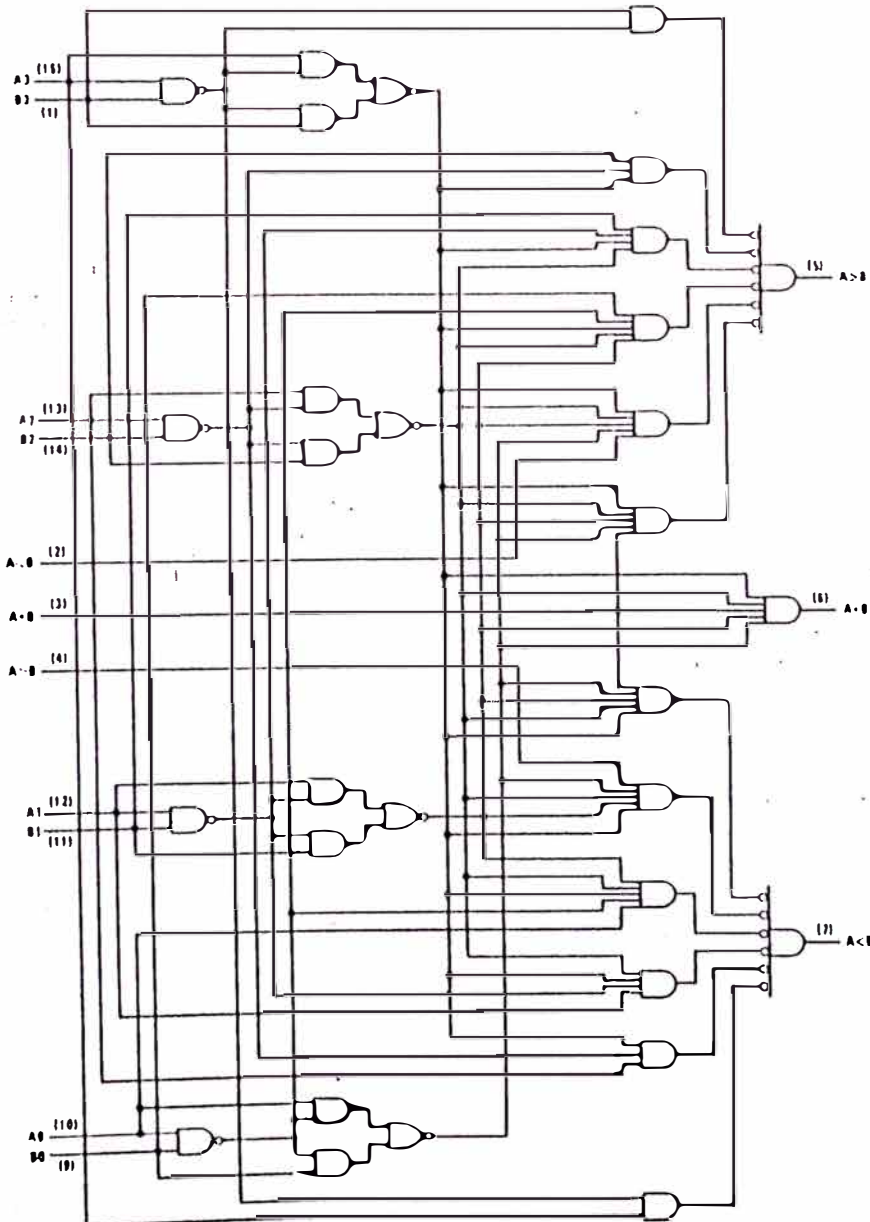
(2) Not more than one output should be shorted at a time, and for ECG74LS duration of short circuit should not exceed one second.

Characteristics $V_{CC} = 5V, T_A = 25^\circ C$

PARAMETER	FROM INPUT	TO OUTPUT	NUMBER OF GATE LEVELS	ECG74				EC074				UNITS
				B5				LS85				
				CONDITIONS	MIN	TYP	MAX	CONDITIONS	MIN	TYP	MAX	
Propagation Delay Time, High Level Output	Any A or B Data Input	A < B, A > B	1		7				14		ns	
			2		12				19			
			3		17	26			24	36		
			4		23	35			23	40		
Propagation Delay Time, Low Level Output	Any A or B Data Input	A < B, A > B	1		11				11		ns	
			2		15				15			
			3		20	30			20	30		
			4		20	30			20	30		
Propagation Delay Time, High Level Output	A < B or A = B	A > B	1	$C_L = 15 \mu F$ $R_L = 400 \Omega$	7	11	$C_L = 15 \mu F$ $R_L = 2 k \Omega$	14	22	ns		
			1		11	17		11	17	ns		
Propagation Delay Time, Low Level Output	A < B or A = B	A > B	1		11	17		11	17	ns		
			2		13	20		13	20	ns		
Propagation Delay Time, High Level Output	A = B	A = B	2		13	20		13	20	ns		
			2		11	17		11	17	ns		
Propagation Delay Time, High Level Output	A > B or A = B	A < B	1		7	11		14	22	ns		
			1		11	17		11	17	ns		
Propagation Delay Time, Low Level Output	A > B or A = B	A < B	1		11	17		11	17	ns		
			1		11	17		11	17	ns		

85, LS85

Logic Diagrams



4-BIT RIGHT SHIFT LEFT SHIFT REGISTERS REGISTRES A DECALAGE DROITE, DECALAGE GAUCHE 4 BITS

BASIC CHARACTERISTICS CARACTERISTIQUES PRINCIPALES

Type	Package Boitier	Operating free-air temperature range Gamme de température ambiante de fonctionnement	V _{CC} (V)		Maximum input voltage Tension d'entrée maximale	Fan-out Sortance		Input loading factor Facteur de charge d'entrée	
			Min	Max		Low state Etat bas	High state Etat haut		
SF.C 495 LSE	TO-116	0°C. - 70°C	4.75	5.25	7 V	20	20	All inputs except clocks Toutes entrées sauf horloges	1
SF.C 495 LSEM	TO-116	-55°C. - 125°C	4.5	5.5	7 V	10	20	Clock inputs Entrées horloge	2

- N bit serial to parallel converter
- N bit parallel to serial converter
- N bit storage register

- Conversion de N bits série parallèle
- Conversion de N bits parallèle série
- Registre de stockage de N bits

GENERAL DESCRIPTION

The SF.C 495 is a TTL monolithic circuit, employing 4 R-S master-slave flip-flops and other gates for either right-shift or left-shift operation. These operations are depending upon the logical level to the mode control input.

When a low level is applied to this input, the clock 1 is enabled and right-shift operation is performed.

When a high level is applied to the mode control input, the clock 2 is enabled and data parallel inputs (A_i, B_i, C_i, D_i) are loading the register. In this mode, left-shift can be accomplished by connecting the output of each flip-flop to the parallel input of the previous flip-flop, and serial data is entered at input D.

Data transfer occurs on the negative transition of the clock pulse.

Input clamping diodes

Average power dissipation 65 mW

Completely compatible with series 54/74 TTL and DTL.

DESCRIPTION GENERALE

Le circuit SF.C 495 est un circuit TTL monolithique. Il utilise 4 bascules maître-esclave RS et d'autres operateurs pour réaliser le décalage à droite ou le décalage à gauche. Ces 2 opérations dépendent du niveau logique de l'entrée contrôle 2 mode.

Si un niveau bas est appliqué sur cette entrée, l'horloge 1 est validée et on réalise alors le décalage à droite.

Si un niveau haut est appliqué sur l'entrée de contrôle, l'horloge 2 est validée, et on peut stocker dans le registre les données des bornes A_i, B_i, C_i, D_i. Dans ce mode, le décalage à gauche est réalisé en reliant la sortie de chaque bascule à l'entrée parallèle de la bascule précédente, l'entrée D_i est alors une entrée série.

Le transfert des données se fait lorsque les horloges passent du niveau haut au niveau bas

Diode d'écrêtage sur les entrées

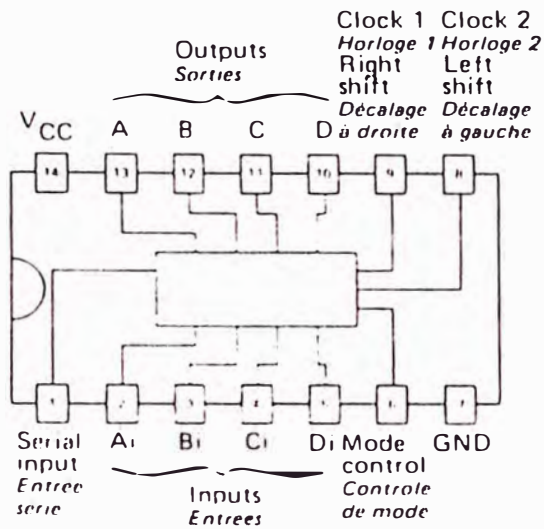
Dissipation moyenne 65 mW

Compatible avec les séries 54/74 TTL et DTL.

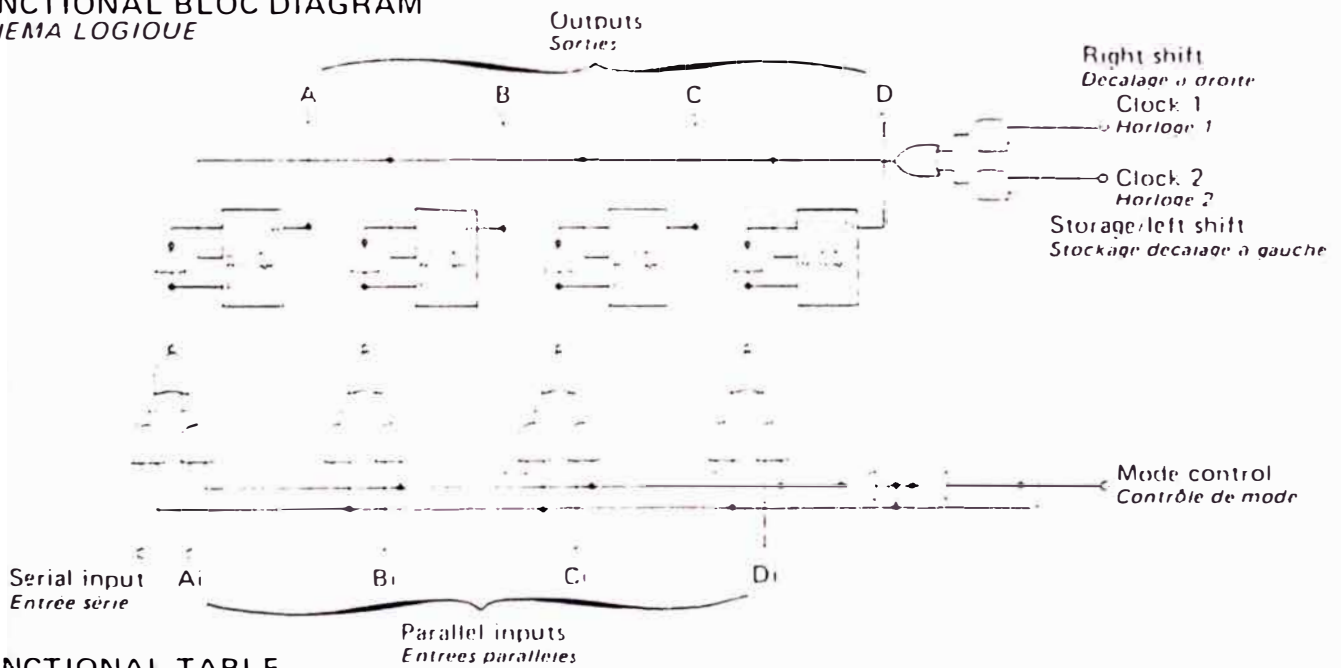
PIN CONFIGURATION
BOUCHAGE

Package TO-116 (CB-2)
 Boîtier

Top view
 Vue de dessus



FUNCTIONAL BLOC DIAGRAM
SCHEMA LOGIQUE



FUNCTIONAL TABLE
TABLE DE FONCTIONNEMENT

Desired operation <i>Operation désirée</i>	Conditions <i>Conditions</i>			
	Mode control <i>Contrôle de mode</i>	Special connections <i>Connexions spéciales</i>	Data input <i>Entrée des données</i>	Clock <i>Horloge</i>
Right-shift <i>Décalage à droite</i>	L		Serial input <i>Entrée série</i>	Clock 1 <i>Horloge 1</i>
Storage register <i>Registre de stockage</i>	H		Ai, Bi, Ci, Di	Clock 2 <i>Horloge 2</i>
Left-shift <i>Décalage à gauche</i>	H	D to Ci, C to Bi, B to Ai	Input Di <i>Entrée</i>	Clock 2 <i>Horloge 2</i>

ELECTRICAL CHARACTERISTICS
CARACTERISTIQUES ELECTRIQUES

 Over operating free-air temperature range (unless otherwise specified).
 Dans toute la gamme de température ambiante de fonctionnement (sauf indications
 contraires).

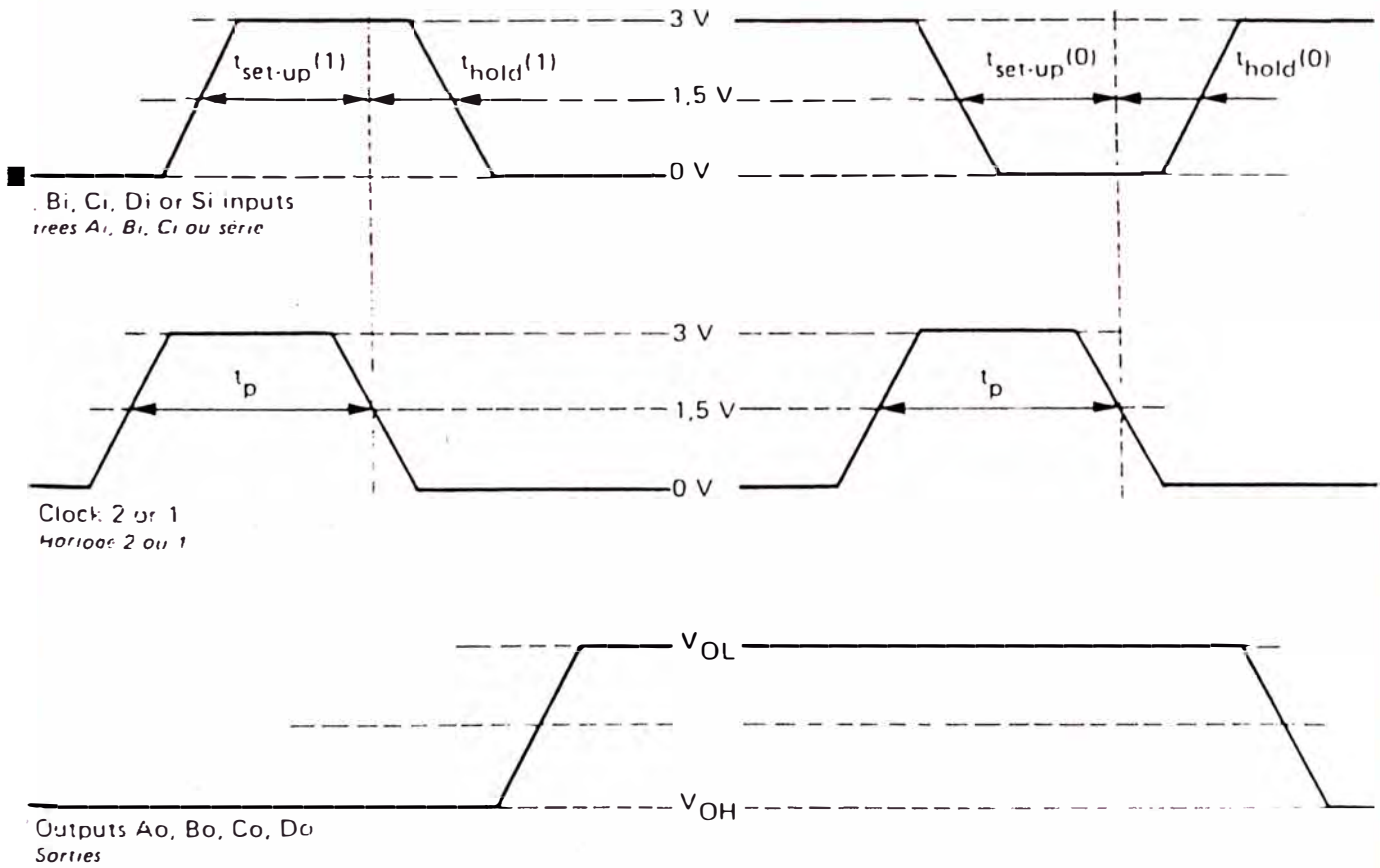
PARAMETERS PARAMETRES		TEST CONDITIONS CONDITIONS DE MESURE	LSEM			LSE			
			MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
V_{IHmin}	Input high level Niveau d'entrée haut		2			2			V
V_{ILmax}	Input low level Niveau d'entrée bas				0,8			0,8	V
V_{ILmin}	Input clamp voltage Tension d'écrobage d'entrée	V_{CCmin} $I_I = -18$ mA			-1,5			-1,5	V
V_{OHmin}	Output high voltage Tension de sortie haute	V_{CCmin} $V_I = 0,8$ V $I_O = -0,4$ mA	2,5			2,7			V
V_{OLmax}	Output low voltage Tension de sortie basse	V_{CCmin} $V_I = 2$ V $I_O = 4$ mA			0,4				V
		V_{CCmin} $V_I = 2$ V $I_O = 8$ mA						0,5	V
I_{IHmax}	Input high current Courant d'entrée haut	V_{CCmax} at clock $V_I = 7$ V			0,2			0,2	mA
		at other			0,1			0,1	mA
I_{IHmin}	Input high current Courant d'entrée haut	V_{CCmax} at clock			40			40	μ A
		$V_I = V_{OHmin}$ at other			20			20	μ A
I_{ILmax}	Input low current Courant d'entrée bas	V_{CCmax} at clock			-0,8			-0,8	mA
		$V_I = V_{OLmax}$ at other			-0,4			-0,4	mA
I_{OS}	Short-circuit output current Courant de sortie en court-circuit	V_{CCmax}	-30		-70	-30		-70	mA
I_{CCH}	Supply current, output high Courant d'alimentation, sortie haute	V_{CCmax}		13	21		13	21	mA

DYNAMIC CHARACTERISTICS
CARACTERISTIQUES DYNAMIQUES
 $t_{amb} = 25^\circ\text{C}$, $V_{CC} = 5$ V, $N = 10$

PARAMETERS PARAMETRES		TEST CONDITIONS CONDITIONS DE MESURE	MIN.	TYP.	MAX.	
Maximum shift frequency Fréquence de décalage maximale			25	36		MHz
t_{PLH}	Propagation delay time, low to high level output Temps de propagation à la croissance du signal de sortie	$R_L = 2\,000$ Ω $C_L = 15$ pF $V_{ref} = 1,4$ V		18	27	ns
			t_{PHL}	Propagation delay time, high to low level output Temps de propagation à la décroissance du signal de sortie		21

RECOMMENDED OPERATING CONDITIONS
 CONDITIONS D'EMPLOI RECOMMANDEES

SET-UP AND HOLD TIMES REQUIRED TO DATA INPUT
 AND SERIAL INPUT WITH RESPECT TO CLOCK INPUTS
 TEMPS DE PREPOSITIONNEMENT ET DE MAINTIEN SUR LES
 ENTREES PARALLELE / ET SERIE PAR RAPPORT AUX ENTREES
 DES HORLOGES



$$t_{\text{set-up}}^{(1)} = t_{\text{set-up}}^{(0)} \geq 10 \text{ ns}$$

$$t_{\text{hold}}^{(1)} = t_{\text{hold}}^{(0)} \geq 0 \text{ ns}$$

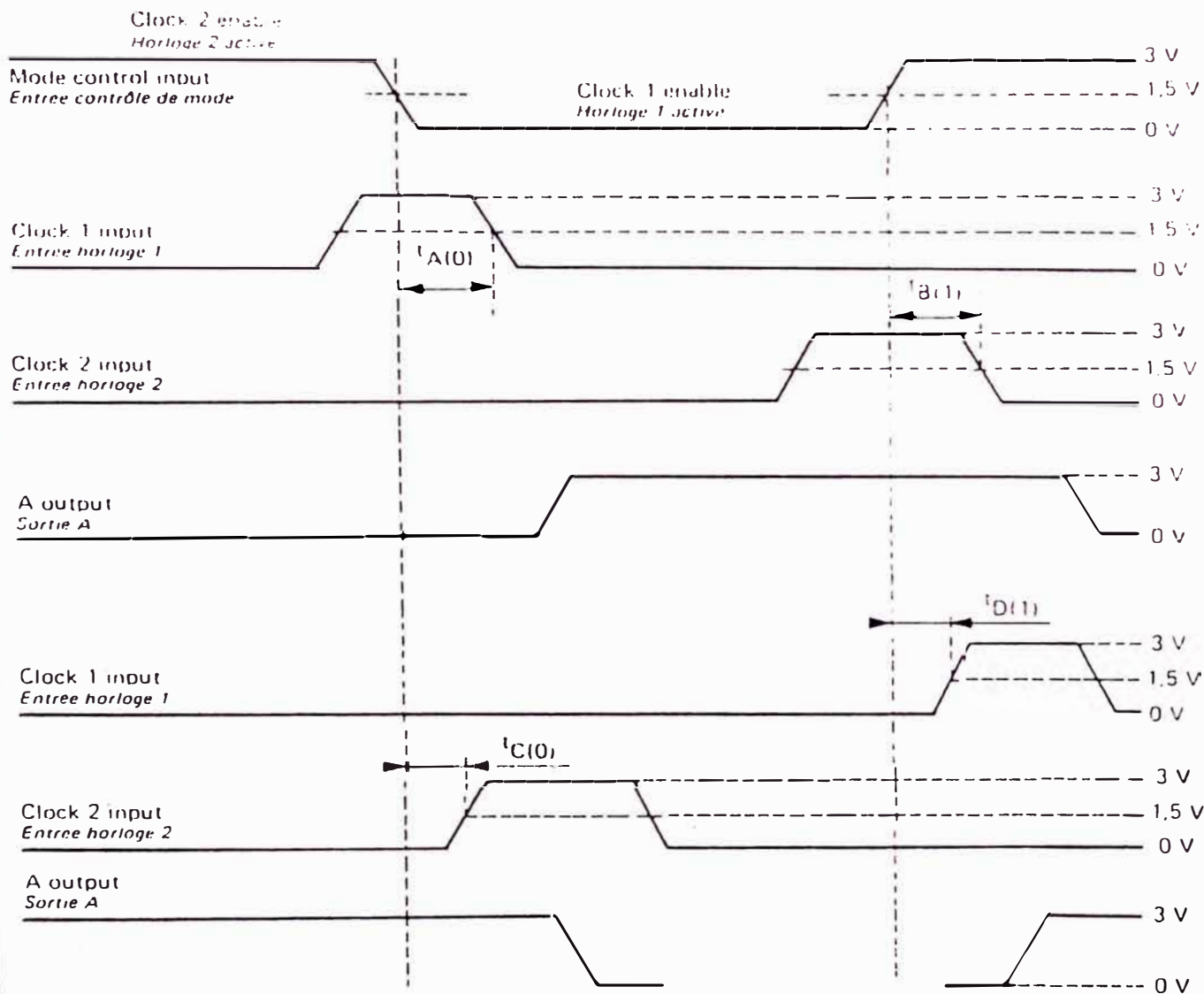
$$t_p \geq 15 \text{ ns} \quad ; \quad t_r = t_f = 10 \text{ ns} \quad (10\% \text{ to } 90\% \text{ on clocks and input data waveforms})$$

(10 % à 90 % pour les impulsions des horloges et des données)

* For serial input test
 Pour l'entrée série

RECOMMENDED OPERATING CONDITIONS
CONDITIONS D'EMPLOI RECOMMANDEES

SET-UP TIMES REQUIRED TO MODE CONTROL WITH RESPECT TO CLOCKS INPUTS
 TEMPS DE PREPOSITIONNEMENT DU CONTROLE DE MODE PAR RAPPORT AUX ENTrees DES HORLOGES



Notes : Serial input is at 4,5 V
 L'entree serie est a 4,5 V

Ai input is at 0 V
 L'entree Ai est a 0 V

from 10% to 90%
 de 10% a 90% $t_{TLH} = t_{THL} = 10 \text{ ns}$

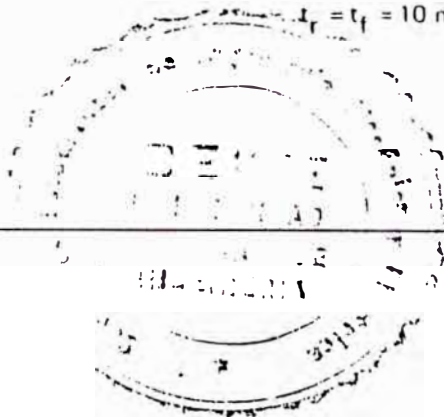
$t_{A(0)} = 15 \text{ ns}$

$t_{B(1)} = 15 \text{ ns}$

$t_{C(0)} = 5 \text{ ns}$

$t_{D(1)} = 5 \text{ ns}$

$t_r = t_f = 10 \text{ ns}$



FAST 74F112

Flip-Flop

Dual J-K Negative Edge-triggered Flip-Flop
Product Specification

Products

DESCRIPTION

The FAST 74F112, Dual Negative Edge-triggered Flip-Flop, features individual clock (\overline{CP}_n), Set (\overline{S}_D) and Reset (\overline{R}_D) inputs, true (Q_n) and complementary (\overline{Q}_n) outputs as shown in the Function Table as well as \overline{R}_D inputs, when Low, set or reset the flip-flop regardless of the level at the other

inputs. When the clock (\overline{CP}_n) input enables the flip-flop, the J and K inputs and data will be active logic levels at the J and K inputs. The flip-flop will perform according to the Function Table as long as the setup and hold times are observed. Output changes are initiated by the falling edge transition of the \overline{CP}_n .

TYPE	TYPICAL f_{MAX}	TYPICAL SUPPLY CURRENT (TOTAL)
N74F112	100MHz	15mA

ORDERING INFORMATION

PACKAGES	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$; $T_A = 0^\circ C$ to $+70^\circ C$
16-Pin Plastic DIP	N74F112N
16-Pin Plastic SO	N74F112D

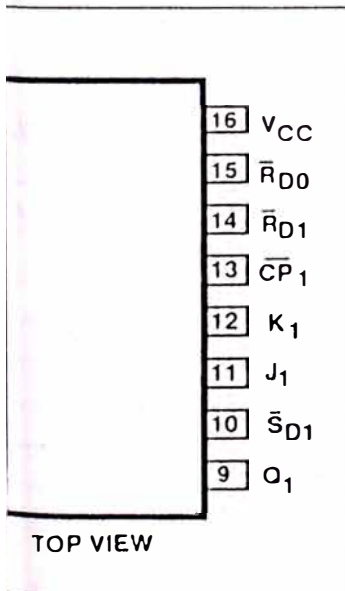
INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74F(U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
J_0, J_1	J inputs	10/10	20 μ A/0.6mA
K_0, K_1	K inputs	10/10	20 μ A/0.6mA
$\overline{S}_{D0}, \overline{S}_{D1}$	Set inputs (active Low)	10/50	20 μ A/3.0mA
$\overline{R}_{D0}, \overline{R}_{D1}$	Reset inputs (active Low)	10/50	20 μ A/3.0mA
$\overline{CP}_0, \overline{CP}_1$	Clock Pulse input (active falling edge)	10/40	20 μ A/2.4mA
$Q_0, \overline{Q}_0, Q_1, \overline{Q}_1$	Data outputs	50/33	10mA/20mA

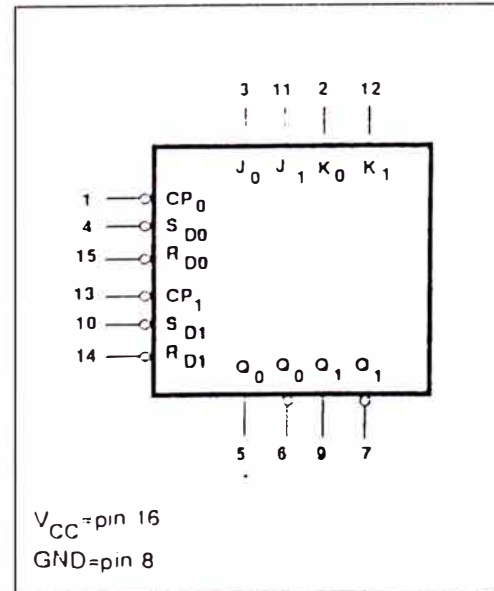
NOTE:

One FAST Unit Load is defined as 20 μ A in the High state and 0.6mA in the Low state.

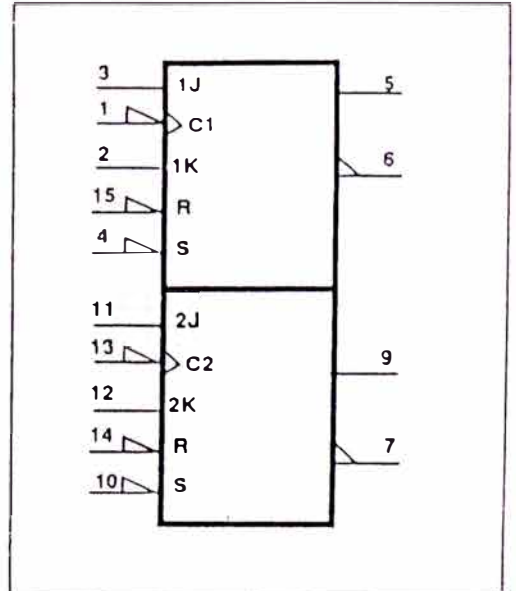
FIGURATION



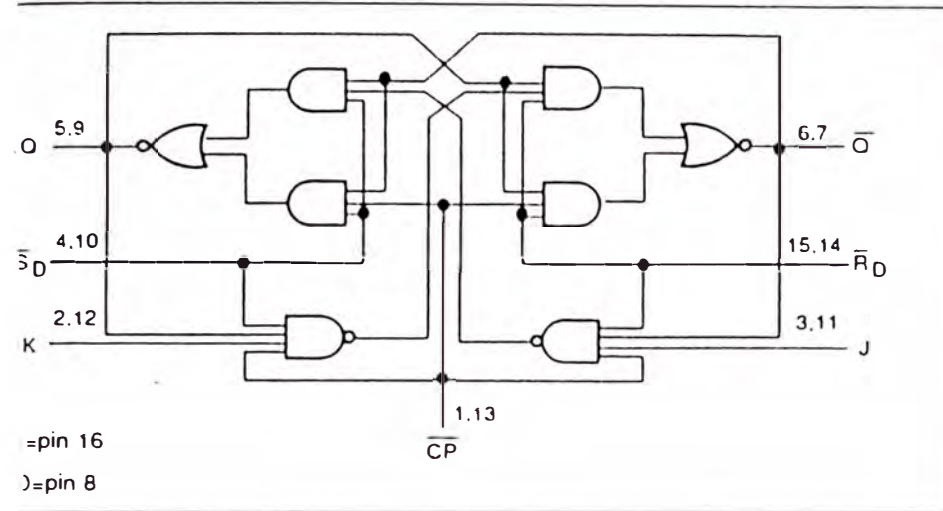
LOGIC SYMBOL



LOGIC SYMBOL (IEEE/IEC)



LOGIC DIAGRAM



Q = pin 16

S_D = pin 8

FUNCTION TABLE

INPUTS				OUTPUTS		OPERATING MODE
\overline{R}_D	\overline{CP}	J	\overline{K}	Q	\overline{Q}	
H	X	X	X	H	L	Asynchronous Set
L	X	X	X	L	H	Asynchronous Reset
L	X	X	X	H*	H*	Undetermined *
H	↓	h	h	q	q	Toggle
H	↓	l	h	L	H	Load "0" (Reset)
H	↓	h	l	H	L	Load "1" (Set)
H	↓	l	l	q	\overline{q}	Hold "no change"
H	H	X	X	Q	\overline{Q}	Hold "no change"

h voltage level

l voltage level one setup time prior to High-to-Low clock transition

l voltage level

l voltage level one setup time prior to High-to-Low clock transition

l case letters indicate the state of the referenced output prior to the High-to-Low clock transition

l care

l-to-Low clock transition

l outputs will be High while both S_D and R_D are Low, but the output states are unpredictable if S_D

l go High simultaneously.

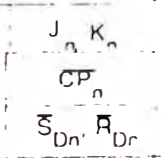
LIMITS MAXIMUM RATINGS (Operation beyond the limits set forth in this table may impair the useful life of the device. Unless otherwise noted these limits are over the operating free-air temperature range.)

CL	PARAMETER	RATING	UNIT
	Supply voltage	-0.5 to +7.0	V
	Input voltage	-0.5 to +7.0	V
	Input current	-30 to +5	mA
	Voltage applied to output in High output state	-0.5 to +V _{CC}	V
	Current applied to output in Low output state	40	mA
	Operating free-air temperature range	0 to +70	°C
	Storage temperature	-65 to +150	°C

RECOMMENDED OPERATING CONDITIONS

CL	PARAMETER	LIMITS			UNIT
		Min	Nom	Max	
	Supply voltage	4.5	5.0	5.5	V
	High-level input voltage	2.0			V
	Low-level input voltage			0.8	V
	Input clamp current			-18	mA
	High-level output current			-1	mA
	Low-level output current			20	mA
	Operating free-air temperature range	0		70	°C

ELECTRICAL CHARACTERISTICS (Over recommended operating free-air temperature range unless otherwise noted)

OL	PARAMETER	TEST CONDITIONS ¹	LIMITS			UNIT
			Min	Typ ²	Max	
	High-level output voltage	$V_{CC} = \text{MIN}, V_{IL} = \text{MAX}$	$\pm 10\%V_{CC}$	2.5		V
		$V_{IH} = \text{MIN}, I_{OH} = \text{MAX}$	$\pm 5\%V_{CC}$	2.7	3.4	V
	Low-level output voltage	$V_{CC} = \text{MIN}, V_{IL} = \text{MAX}$	$\pm 10\%V_{CC}$	0.35	0.50	V
		$V_{IH} = \text{MIN}, I_{OL} = \text{MAX}$	$\pm 5\%V_{CC}$	0.35	0.50	V
	Input clamp voltage	$V_{CC} = \text{MIN}, I_I = I_{IK}$		-0.73	-1.2	V
	Input current at maximum input voltage	$V_{CC} = \text{MAX}, V_I = 7.0\text{V}$			100	μA
	High-level input current	$V_{CC} = \text{MAX}, V_I = 2.7\text{V}$			20	μA
	Low-level input current	$V_{CC} = \text{MAX}, V_I = 0.5\text{V}$ 			-0.5	mA
					-2.4	mA
					-3.0	mA
	Short circuit output current ³	$V_{CC} = \text{MAX}$		-60	-150	mA
	Supply current (total) ⁴	$V_{CC} = \text{MAX}$		15	21	mA

Conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type. Typical values are at $V_{CC} = 5\text{V}, T_A = 25^\circ\text{C}$. More than one output should be shorted at a time. For testing I_{OS} , the use of high-speed test apparatus and/or sample-and-hold techniques are preferable in order to minimize internal heating and more accurately reflect operational values. Otherwise, prolonged shorting of a High output may raise the chip temperature above normal and thereby cause invalid readings in other parameter tests. In any sequence of parameter test, I_{OS} tests should be performed last. I_{CC} with the clock input grounded and all outputs open, with the Q and \bar{Q} outputs High in turn.

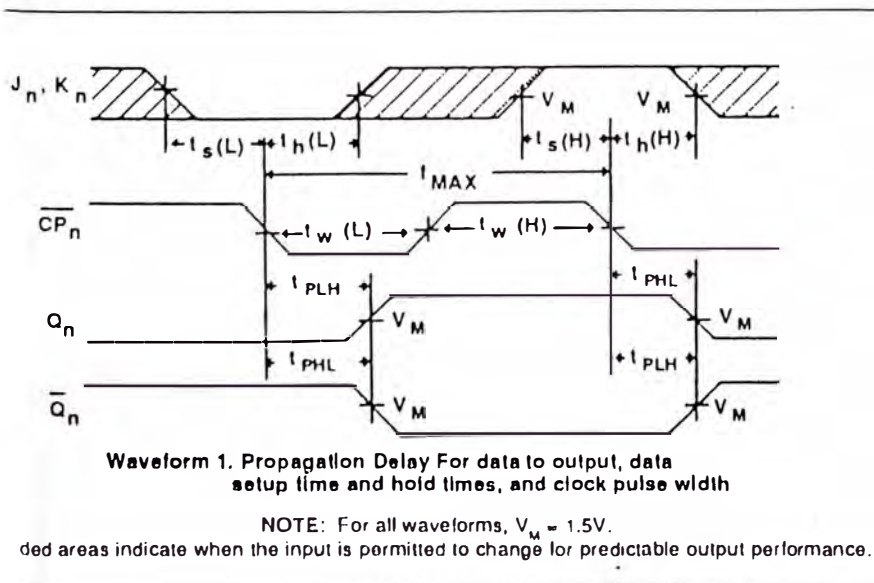
ELECTRICAL CHARACTERISTICS

SYMBOL	PARAMETER	TEST CONDITION	LIMITS					UNIT	
			T _A = +25°C V _{CC} = 5V C _L = 50pF R _L = 500Ω			T _A = 0°C to +70°C V _{CC} = 5V ±10% C _L = 50pF R _L = 500Ω			
			Min	Typ	Max	Min	Max		
t _{AX}	Maximum clock frequency	Waveform 1	85	100		80		MHz	
LH HL	Propagation delay CP _n to O _n or \bar{O}_n	Waveform 1	2.0	5.0	6.5	2.0	7.5		ns
LH HL	Propagation delay S _{Dn} , R _{Dn} to O _n or \bar{O}_n	Waveform 2.3	2.0	4.5	6.5	2.0	7.5		ns

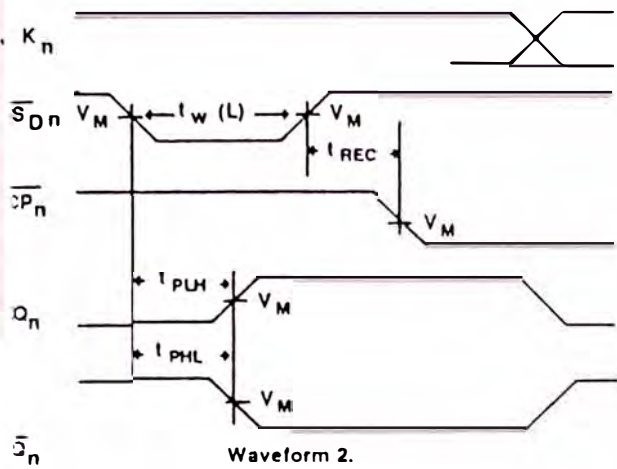
SETUP REQUIREMENTS

SYMBOL	PARAMETER	TEST CONDITION	LIMITS					UNIT	
			T _A = +25°C V _{CC} = 5V C _L = 50pF R _L = 500Ω			T _A = 0°C to +70°C V _{CC} = 5V ±10% C _L = 50pF R _L = 500Ω			
			Min	Typ	Max	Min	Max		
H) L)	Setup time High or Low J _n , K _n to CP _n	Waveform 1	4.0			5.0			ns
H) L)	Hold time High or Low J _n , K _n to CP _n	Waveform 1	0.0			0.0			ns
H) L)	CP _n Pulse width, High or Low	Waveform 1	4.5			5.0			ns
-)	\bar{S}_D or \bar{R}_D Pulse width, Low	Waveform 2.3	4.5			5.0			ns
C	Recovery time \bar{S}_D or \bar{R}_C to CP _n	Waveform 2.3	4.0			5.0			ns

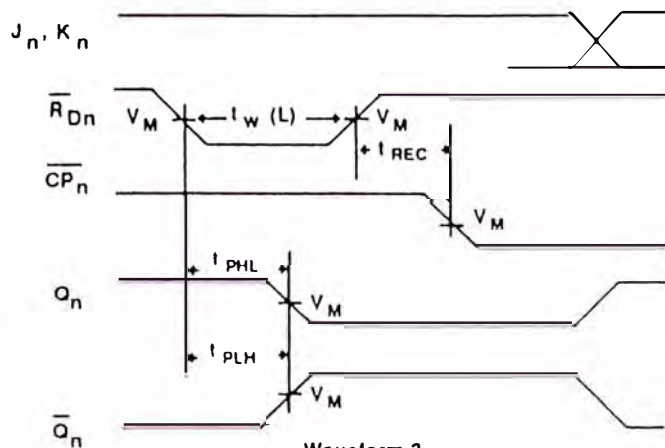
WAVEFORMS



WAVEFORMS



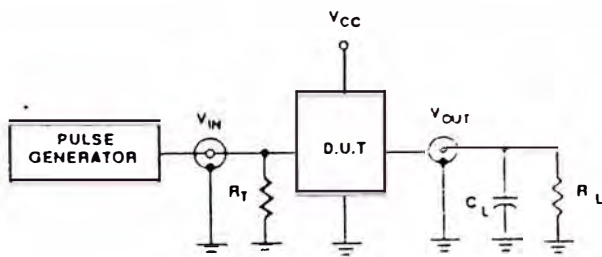
Waveform 2.
Propagation Delay for Set to Output, Set Pulse Width, and Recovery Time for Set to Clock



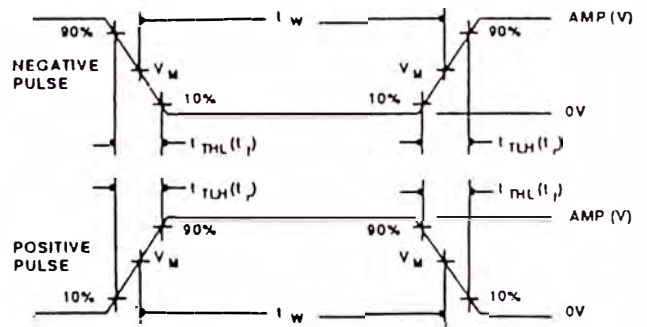
Waveform 3.
Propagation Delay for Reset to Output, Reset Pulse Width, and Recovery Time for Reset to Clock

NOTE: For all waveforms, $V_M = 1.5V$

CIRCUIT AND WAVEFORMS



Test Circuit For Totem-Pole Outputs



$V_M = 1.5V$

Input Pulse Definition

DEFINITIONS

- = Load resistor; see AC CHARACTERISTICS for value.
- = Load capacitance includes jig and probe capacitance; see AC CHARACTERISTICS for value.
- = Termination resistance should be equal to Z_{OUT} of pulse generators.

FAMILY	INPUT PULSE REQUIREMENTS				
	Amplitude	Rep. Rate	t_w	t_{TLH}	t_{THL}
74F	3.0V	1MHz	500ns	2.5ns	2.5ns

Ivania ECG micconductors

ECG74LS138, 74S138, 74LS139

Decoders/Demultiplexers

Key-clamped circuits are designed to be used in memory-decoding or data-routing applications requiring very short propagation delay times. In memory systems these decoders can be used to minimize the effects of system decoding. When used with high-speed memories, the delay times of these decoders are usually less than the typical access time of the memory. This means that the effective system delay in the decoder is negligible.

The 74LS138 and 74S138 decode one-of-eight lines, based on combinations at the three binary select inputs and two enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented with no external inverters, and a 32-line decoder can be implemented with one inverter. An enable input can be used as a data line in demultiplexing applications.

The 74LS139 comprises two separate two-line-to-four-line decoders in a single package. The active-low enable input can also be used as a data line in demultiplexing applications.

These decoders/demultiplexers feature fully buffered outputs, presenting only one normalized load to its driving circuit. The outputs are clamped with high-performance diodes to suppress line-ringing and simplify interfacing.

Designed specifically for high-speed:

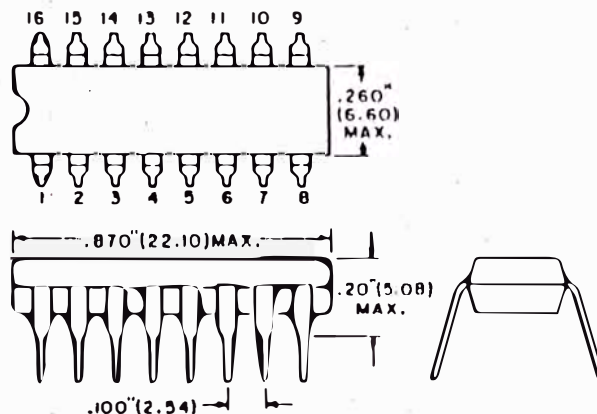
- Memory decoders
- Transmission systems

The 74LS138 3-to-8-line decoder incorporates 3 enable inputs to simplify cascading and/or data expansion.

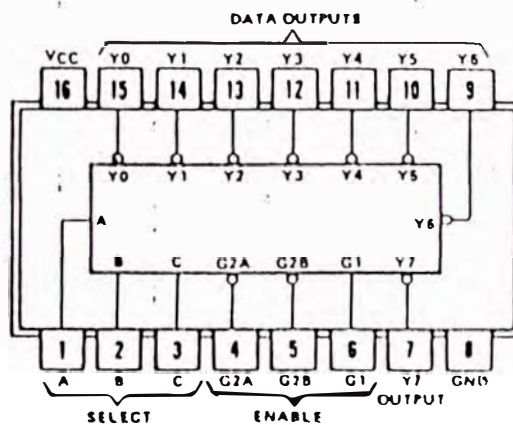
The 74LS139 contains two fully independent 2-to-4-line decoders/demultiplexers.

Key-clamped for high performance

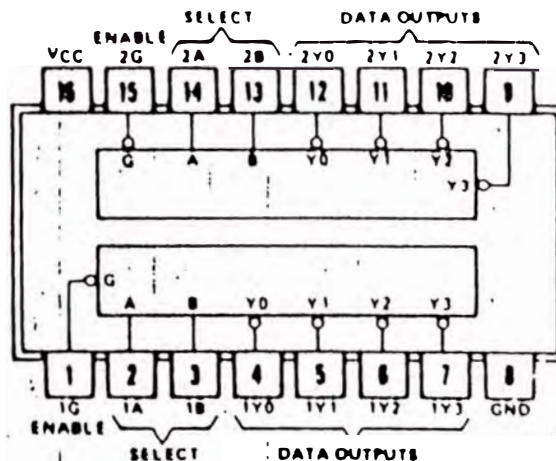
TYPICAL PROPAGATION DELAY (3 LEVELS OF LOGIC)	TYPICAL POWER DISSIPATION
21 ns	32 mW
8 ns	245 mW
21 ns	34 mW



ECG74LS138, ECG74S138—Dual 1-of-8



ECG74LS139—Dual 1-of-4



Electrical Characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		CONDITIONS		ECG74			ECG74			UNITS	
				LS138, LS139			S138				
				MIN	TYP(1)	MAX	MIN	TYP(1)	MAX		
V_{IH}	High Level Input Voltage			2			2			V	
V_{IL}	Low Level Input Voltage					0.8			0.8	V	
V_I	Input Clamp Voltage	$V_{CC} = \text{Min}, I_I = -18 \text{ mA}$				-1.5			-1.2	V	
I_{OH}	High Level Output Current					-400			-1000	μA	
V_{OH}	High Level Output Voltage	$V_{CC} = \text{Min}, V_{IH} = 2\text{V}$ $V_{IL} = \text{Max}, I_{OH} = \text{Max}$		2.7	3.4		2.7	3.4		V	
I_{OL}	Low Level Output Current					8			20	mA	
V_{OL}	Low Level Output Voltage	$V_{CC} = \text{Min}, V_{IH} = 2\text{V}$ $V_{IL} = \text{Max}$		$I_{OL} = \text{Max}$					0.5	V	
				$I_{OL} = 4 \text{ mA}$				0.35	0.5		0.4
I_I	Input Current at Maximum Input Voltage	$V_{CC} = \text{Max}$		$V_I = 5.5\text{V}$					1	mA	
				$V_I = 7\text{V}$				0.1			
I_{IH}	High Level Input Current	$V_{CC} = \text{Max}$		$V_I = 2.7\text{V}$				20	50	μA	
I_{IL}	Low Level Input Current	$V_{CC} = \text{Max}$		$V_I = 0.4\text{V}$				-0.36		mA	
				$V_I = 0.5\text{V}$					-2		
I_{OS}	Short Circuit Output Current	$V_{CC} = \text{Max}(2)$				-30		-130	-40	-100	mA
I_{CC}	Supply Current	$V_{CC} = \text{Max}$ Outputs Enabled and Open		LS138, S138		6.3	10		49	74	mA
				LS139		6.8	11		60	90	

Notes

- (1) All typical values are at $V_{CC} = 5\text{V}, T_A = 25^\circ\text{C}$.
- (2) Not more than one output should be shorted at a time, and duration of short circuit should not exceed one second.

ECG74LS138, 74S138, 74LS139

						MIN	TYP	MAX	MIN	TYP	MAX			MIN	TYP	MAX	
t_{PLH}	Propagation Delay Time, Low-to-High Level Output	Binary Select	Any	2	$C_L = 15 \text{ pF}$ $R_L = 2 \text{ k}\Omega$	13		20	13		20	4.5	7			ns	
t_{PHL}	Propagation Delay Time, High-to-Low Level Output					27	41	22	33	7	10.5		ns				
t_{PLH}	Propagation Delay Time, Low-to-High Level Output			3		18	27	18	29	7.5	12		ns				
t_{PHL}	Propagation Delay Time, High-to-Low Level Output					26	39	25	38	8	12		ns				
t_{PLH}	Propagation Delay Time, Low-to-High Level Output	Enable	Any	2		$C_L = 15 \text{ pF}$ $R_L = 280 \Omega$	12	18	16	24	5	8			ns		
t_{PHL}	Propagation Delay Time, High-to-Low Level Output						21	32	21	32	7	11		ns			
t_{PLH}	Propagation Delay Time, Low-to-High Level Output			3				17	26		N/A	7	11		ns		
t_{PHL}	Propagation Delay Time, High-to-Low Level Output							25	38		N/A	7	11		ns		

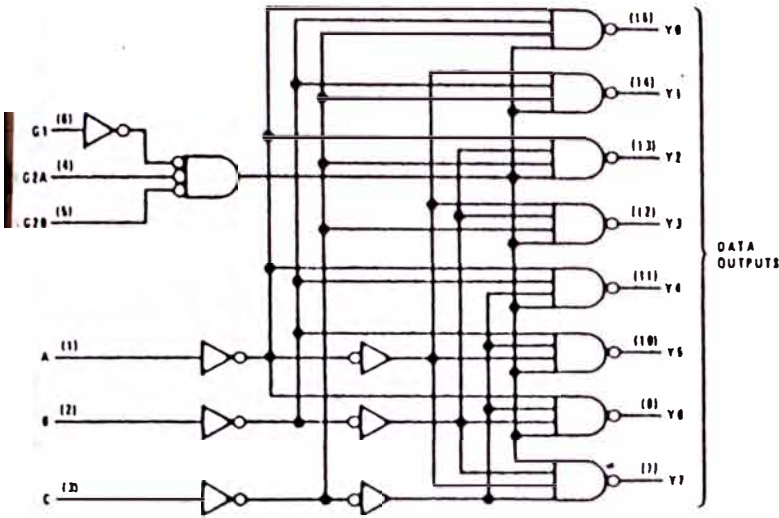
Max Ratings/Operating Conditions

RATINGS	74 SERIES	74H SERIES	74LS SERIES		74S SERIES	UNITS
			DIODE INPUTS	EMITTER INPUTS		
Maximum Allowable Supply Voltage	7	7	7	7	7	V
Guaranteed Operating Supply Voltage Range	4.75 to 5.25					V
Maximum Input Voltage	5.5	5.5	7	5.5	5.5	V
Maximum Voltage to Open-Collector Outputs*	7	7	7	7	7	V
Operating Free-Air Temperature Range	0 to +70					°C
Storage Temperature Range	-65 to +150					°C

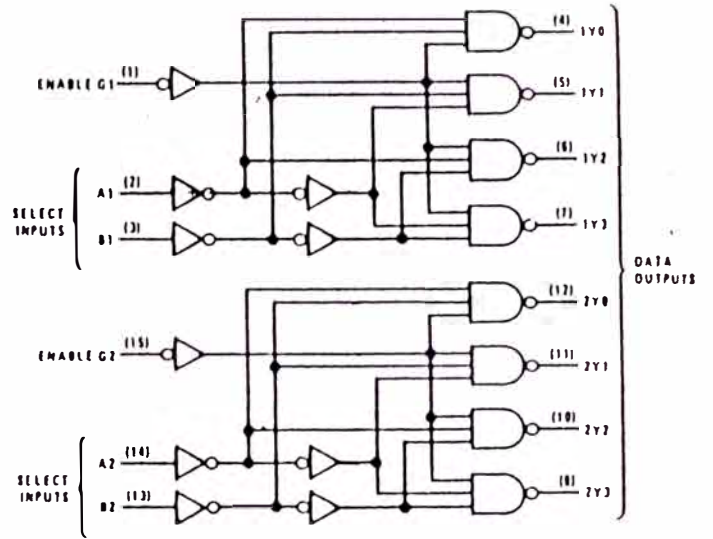
*Except for selected high voltage types, as specified in electrical tables.

Diagrams

LS138, S138



LS139



Truth Tables

LS138, S138

INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

*G2 = G2A + G2B

H = High level, L = low level, X = don't care

LS139

INPUTS			OUTPUTS			
ENABLE	SELECT					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = high level, L = low level, X = don't care

FAST 74F139

Decoder/Demultiplexer

ST Products

Dual 1-of-4 Decoder//Demultiplexer

FEATURES

- Multiplexing capability
- Two independent 1-of-4 decoders
- Multifunction capability

Product Specification

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F139	5.3ns	13mA

DESCRIPTION

The 74F139 is a high speed, dual 1-of-4 decoder/demultiplexer. This device has two independent decoders, each accepting two binary weighted inputs (A_{0n}, A_{1n}) and providing four mutually exclusive active-low outputs ($\bar{O}_{0n} - \bar{O}_{3n}$). Each decoder has an active-low Enable (\bar{E}). When \bar{E} is High, every output is forced High. The Enable can be used as the Data input for a 1-of-4 demultiplexer application.

ORDERING INFORMATION

PACKAGES	COMMERCIAL RANGE
16-Pin Plastic DIP	N74F139N
16-Pin Plastic SO	N74F139D

$V_{CC} = 5V \pm 10\%$; $T_A = 0^\circ C$ to $+70^\circ C$

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74F(U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
A_{0a}, A_{1a}	Address inputs	1/0/1/0	20 μ A/0.6mA
\bar{E}_a, \bar{E}_b	Enable inputs (active Low)	1/0/1/0	20 μ A/0.6mA
$\bar{O}_{0a}, \bar{O}_{3a}$	Data outputs (active Low)	50/33	1.4mA/20mA

NOTE:

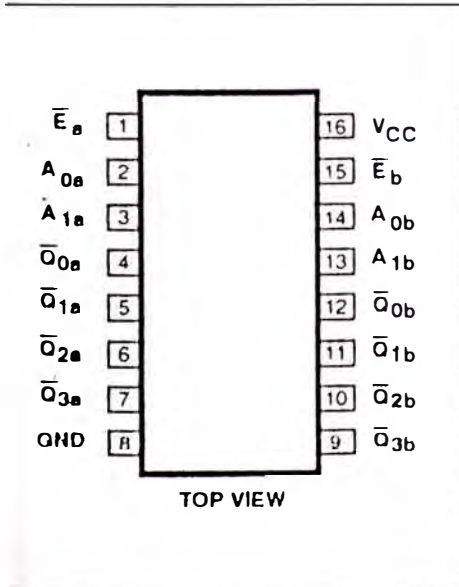
One (1) FAST Unit Load is defined as 20 μ A in the High state and 0.6mA in the Low state.

FUNCTION TABLE

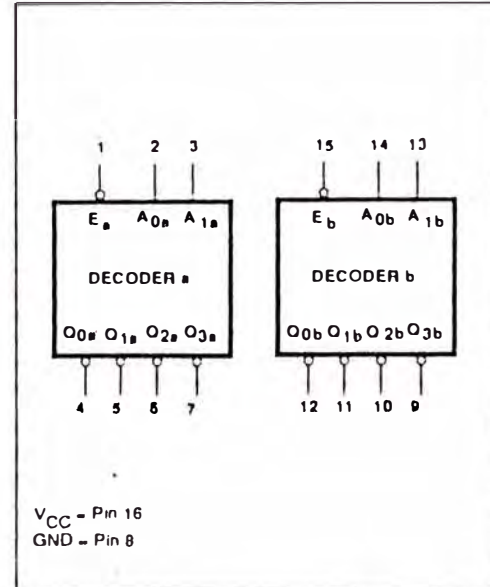
	INPUTS		OUTPUTS			
	A_0	A_1	\bar{O}_0	\bar{O}_1	\bar{O}_2	\bar{O}_3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	H	L	H	L	H	H
L	L	H	H	H	L	H
L	H	H	H	H	H	L

- High voltage level
- Low voltage level
- Don't care

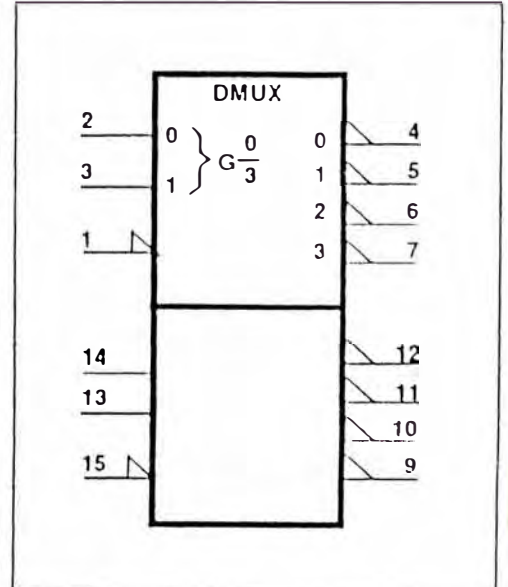
PIN CONFIGURATION



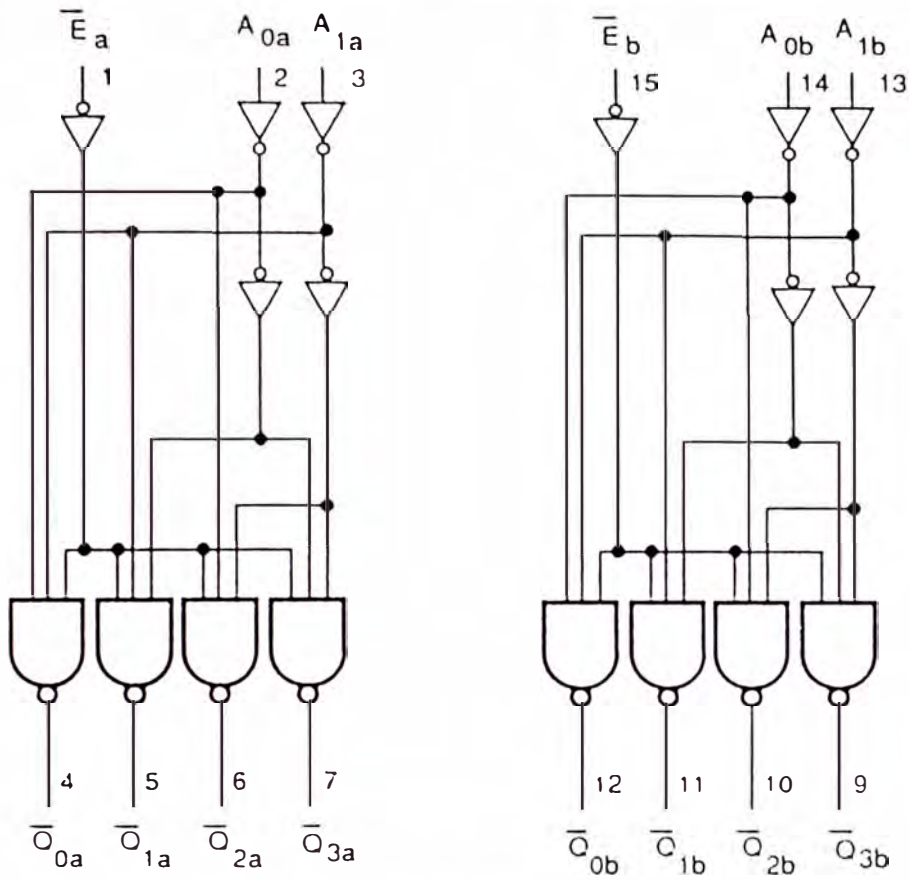
LOGIC SYMBOL



LOGIC SYMBOL (IEEE/IEC)



DIAGRAM



UTE MAXIMUM RATINGS (Operation beyond the limits set forth in this table may impair the useful life of the device. Unless otherwise noted these limits are over the operating free-air temperature range)

PARAMETER	RATING	UNIT
Supply voltage	-0.5 to +7.0	V
Input voltage	-0.5 to +7.0	V
Input current	-30 to +5	mA
Voltage applied to output in High output state	-0.5 to +V _{CC}	V
Current applied to output in Low output state	40	mA
Operating free-air temperature range	0 to +70	°C
Storage temperature	-65 to +150	°C

COMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	LIMITS			UNIT
		Min	Nom	Max	
V _{CC}	Supply voltage	4.5	5.0	5.5	V
V _{IH}	High-level input voltage	2.0			V
V _{IL}	Low-level input voltage			0.8	V
I _{ICL}	Input clamp current			-18	mA
I _{OIH}	High-level output current			-1	mA
I _{OL}	Low-level output current			20	mA
T _{OP}	Operating free-air temperature range	0		70	°C

ELECTRICAL CHARACTERISTICS (Over recommended operating free-air temperature range unless otherwise noted)

SYMBOL	PARAMETER	TEST CONDITIONS ¹	LIMITS			UNIT
			Min	Typ ²	Max	
V _{OIH}	High-level output voltage	V _{CC} = MIN, V _{IL} = MAX	±10%V _{CC}	2.5		V
		V _{IH} = MIN, I _{OIH} = MAX	±5%V _{CC}	2.7	3.4	V
V _{OL}	Low-level output voltage	V _{CC} = MIN, V _{IL} = MAX	±10%V _{CC}	0.30	0.50	V
		V _{IH} = MIN, I _{OL} = MAX	±5%V _{CC}	0.30	0.50	V
V _{ICL}	Input clamp voltage	V _{CC} = MIN, I _I = I _{IK}		-0.73	-1.2	V
I _{ICM}	Input current at maximum input voltage	V _{CC} = MAX, V _I = 7.0V			100	µA
I _{IH}	High-level input current	V _{CC} = MAX, V _I = 2.7V			20	µA
I _{IL}	Low-level input current	V _{CC} = MAX, V _I = 0.5V			-0.6	mA
I _{OS}	Short circuit output current ³	V _{CC} = MAX		-60	-150	mA
I _{CC}	Supply current (total)	V _{CC} = MAX		13	20	mA

¹ Conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.
² Typical values are at V_{CC} = 5V, T_A = 25°C.
³ More than one output should be shorted at a time. For testing I_{OS}, the use of high-speed test apparatus and/or sample-and-hold techniques are preferable in order to minimize internal heating and more accurately reflect operational values. Otherwise, prolonged shorting of a High output may raise the chip temperature above normal and thereby cause invalid readings in other parameter tests. In any sequence of parameter test, I_{OS} tests should be performed last.

RECOMMENDED OPERATING CONDITIONS

PARAMETER	LIMITS			UNIT
	Min	Norm	Max	
Supply voltage	4.5	5.0	5.5	V
High-level input voltage	2.0			V
Low-level input voltage			0.8	V
Input clamp current			-18	mA
High-level output current			-1	mA
Low-level output current			20	mA
Operating free-air temperature range			70	°C

ELECTRICAL CHARACTERISTICS (Over recommended operating free-air temperature range unless otherwise noted)

PARAMETER	TEST CONDITIONS ¹	LIMITS			UNIT
		Min	Typ ²	Max	
High-level output voltage	$V_{CC} = \text{MIN}, V_{IL} = \text{MAX}$		2.5		V
	$V_{IH} = \text{MIN}, I_{OH} = \text{MAX}$	$\pm 10\%V_{CC}$	2.7	3.4	V
Low-level output voltage	$V_{CC} = \text{MIN}, V_{IL} = \text{MAX}$	$\pm 10\%V_{CC}$	0.30	0.50	V
	$V_{IH} = \text{MIN}, I_{OL} = \text{MAX}$	$\pm 5\%V_{CC}$	0.30	0.50	V
Input clamp voltage	$V_{CC} = \text{MIN}, I_I = I_{IK}$		-0.73	-1.2	V
Input current at maximum input voltage	$V_{CC} = \text{MAX}, V_I = 7.0\text{V}$			100	μA
High-level input current	$V_{CC} = \text{MAX}, V_I = 2.7\text{V}$			20	μA
Low-level input current	$V_{CC} = \text{MAX}, V_I = 0.5\text{V}$			-0.6	mA
Short circuit output current ³	$V_{CC} = \text{MAX}$		-60	-150	mA
Supply current (total)	$V_{CC} = \text{MAX}$		13	20	mA

¹ Conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.
² Typical values are at $V_{CC} = 5\text{V}, T_A = 25^\circ\text{C}$.
³ More than one output should be shorted at a time. For testing I_{OS} , the use of high-speed test apparatus and/or sample-and-hold techniques are preferable in order to minimize internal heating and more accurately reflect operational values. Otherwise, prolonged shorting of a High output may raise the chip temperature above normal and thereby cause invalid readings in other parameter tests. In any sequence of parameter test, I_{OS} tests should be performed last.

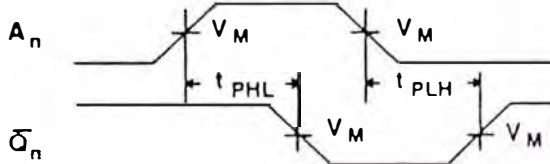
Encoder/Demultiplexer

FAST 74F139

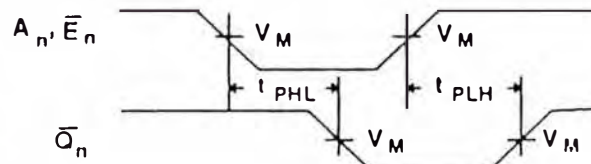
CRITICAL CHARACTERISTICS

PARAMETER	TEST CONDITION	LIMITS					UNIT
		$T_A = +25^\circ\text{C}$ $V_{CC} = 5\text{V}$ $C_L = 50\text{pF}$ $R_L = 500\Omega$			$T_A = 0^\circ\text{C to } +70^\circ\text{C}$ $V_{CC} = 5\text{V} \pm 10\%$ $C_L = 50\text{pF}$ $R_L = 500\Omega$		
		Min	Typ	Max	Min	Max	
Propagation delay A_0 or A_1 to \bar{O}_{na} , \bar{O}_{nb}	Waveform 1, 2	3.5 4.0	5.3 6.1	7.0 8.0	3.0 4.0	8.0 9.0	ns
Propagation delay E_n to \bar{O}_{na} , \bar{O}_{nb}	Waveform 2	3.5 3.0	5.4 4.7	7.0 6.5	3.5 3.0	8.0 7.5	ns

WAVEFORMS



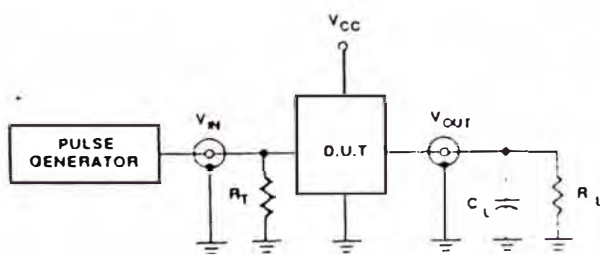
Waveform 1. For Inverting Outputs



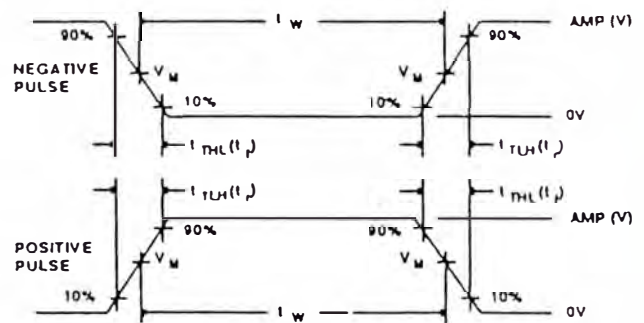
Waveform 2. For Non-Inverting Outputs

NOTE: For all waveforms, $V_M = 1.5\text{V}$

CIRCUIT AND WAVEFORMS



Test Circuit For Totem-Pole Outputs



$V_M = 1.5\text{V}$

Input Pulse Definition

FAMILY	INPUT PULSE REQUIREMENTS				
	Amplitude	Rep. Rate	t_W	t_{TLH}	t_{THL}
74F	3.0V	1MHz	500ns	2.5ns	2.5ns

DEFINITIONS

- = Load resistor; see AC CHARACTERISTICS for value
- = Load capacitance includes jig and probe capacitance; see AC CHARACTERISTICS for value
- = Termination resistance should be equal to Z_{OUT} of pulse generators.

Pennsylvania ECGSM Semiconductors

ECG74192, 74LS192, 74193, 74LS193

Synchronous Up/Down Counters with Clear

Circuits are synchronous up/down counters; the 74 and LS192 circuits are BCD counters and the 74 and LS193 are 4-bit binary counters. Synchronous operation is provided by having all flip-flops change simultaneously, so that the outputs change when so instructed by the steering logic. Mode of operation eliminates the output counting normally associated with asynchronous (ripple) counters.

Outputs of the four master-slave flip-flops are controlled by a low-to-high level transition of either clock input. The direction of counting is determined by which count input is pulsed, while the other input is held high.

Counters are fully programmable; that is, each counter may be preset to either level by entering the data at the inputs while the load input is low. Output will change independently of the count. This feature allows the counters to be used as N dividers by simply modifying the count length and preset inputs.

A load input has been provided which, when taken to a high level, forces all outputs to the low level; independent of count and load inputs. The clear, count, and load inputs are buffered to lower the drive requirements of drivers, etc., required for long words.

These counters were designed to be cascaded without the need for external circuitry. Both borrow and carry outputs are available to cascade both the up and down counting functions. The borrow output produces a pulse equal in width to the count down input when the counter underflows. Similarly, the carry output produces a pulse equal in width to the count down input when an overflow condition exists. The counters can then be easily cascaded by feeding the borrow and carry outputs to the count down and count up inputs respectively of the succeeding counter.

Features

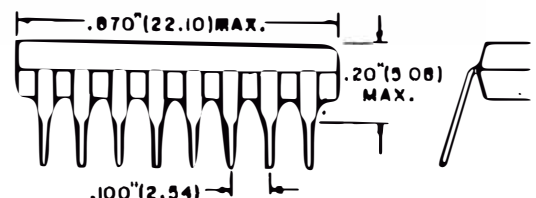
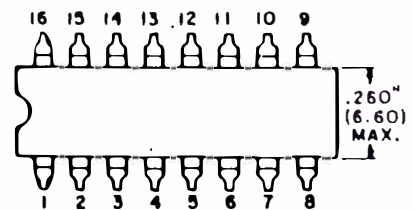
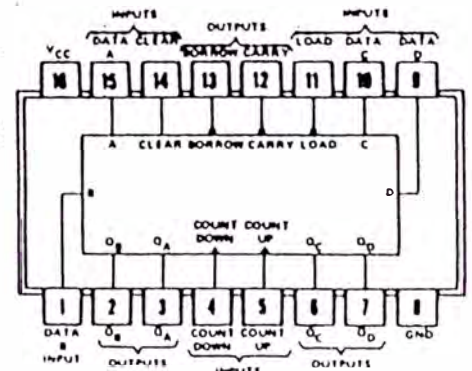
- Fully independent clear input
- Synchronous operation
- Cascading circuitry provided internally
- Individual preset each flip-flop

TYPE	TYPICAL COUNT FREQUENCY	TYPICAL POWER DISSIPATION
192, 193	26 MHz	325 mW
LS192, LS193	32 MHz	95 mW

Specifications/Operating Conditions

RATINGS	74 SERIES	74H SERIES	74LS SERIES		74S SERIES	UNITS
			DIODE INPUTS	EMITTER INPUTS		
Maximum Allowable Voltage	7	7	7	7	7	V
Recommended Operating Voltage Range	4.75 to 5.25					V
Maximum Input Voltage	5.5	5.5	7	5.5	5.5	V
Maximum Voltage to Open- Collector Outputs*	7	7	7	7	7	V
Operating Free-Air Temperature Range	0 to +70					°C
Storage Temperature Range	-65 to +150					°C

* For selected high voltage types, as specified in electrical tables.



Electrical Characteristics over recommended operating free-air temperature range

PARAMETER		CONDITIONS		192, 193			LS192, LS193			UNITS
				MIN	TYP(1)	MAX	MIN	TYP(1)	MAX	
				V_{IH}	High Level Input Voltage			2		
V_{IL}	Low Level Input Voltage			0.8			0.8			V
V_I	Input Clamp Voltage	$V_{CC} = \text{Min}$	$I_I = -12 \text{ mA}$ $I_I = -18 \text{ mA}$	-1.5			-1.5			V
I_{OH}	High Level Output Current			-400			-400			μA
V_{OH}	High Level Output Voltage	$V_{CC} = \text{Min}, V_{IH} = 2\text{V}$ $V_{IL} = \text{Max}, I_{OH} = \text{Max}$		2.4			2.7 3.4			V
I_{OL}	Low Level Output Current			16			8			mA
V_{OL}	Low Level Output Voltage	$V_{CC} = \text{Min}, V_{IH} = 2\text{V}$ $V_{IL} = \text{Max}$	$I_{OL} = \text{Max}$	0.4			0.35 0.5			V
			$I_{OL} = 4 \text{ mA}$				0.4			
I_I	Input Current at Maximum Input Voltage	$V_{CC} = \text{Max}$	$V_I = 5.5\text{V}$	1						mA
			$V_I = 7\text{V}$				0.1			
I_{IH}	High Level Input Current	$V_{CC} = \text{Max}$	$V_I = 2.4\text{V}$	40						μA
			$V_I = 2.7\text{V}$				20			
I_{IL}	Low Level Input Current	$V_{CC} = \text{Max}$	$V_I = 0.4\text{V}$	-1.6			-0.4			mA
I_{OS}	Short Circuit Output Current	$V_{CC} = \text{Max}(2)$		-18	-55	-30	-130			mA
I_{CC}	Supply Current	$V_{CC} = \text{Max}(3)$		65 102			19 34			mA

Notes

- (1) All typical values are at $V_{CC} = 5\text{V}, T_A = 25^\circ\text{C}$.
- (2) Not more than one output should be shorted at a time, and for ECG74LS duration of short circuit should not exceed one second.
- (3) I_{CC} is measured with all outputs open, clear and load inputs grounded, and all other inputs at 4.5V.

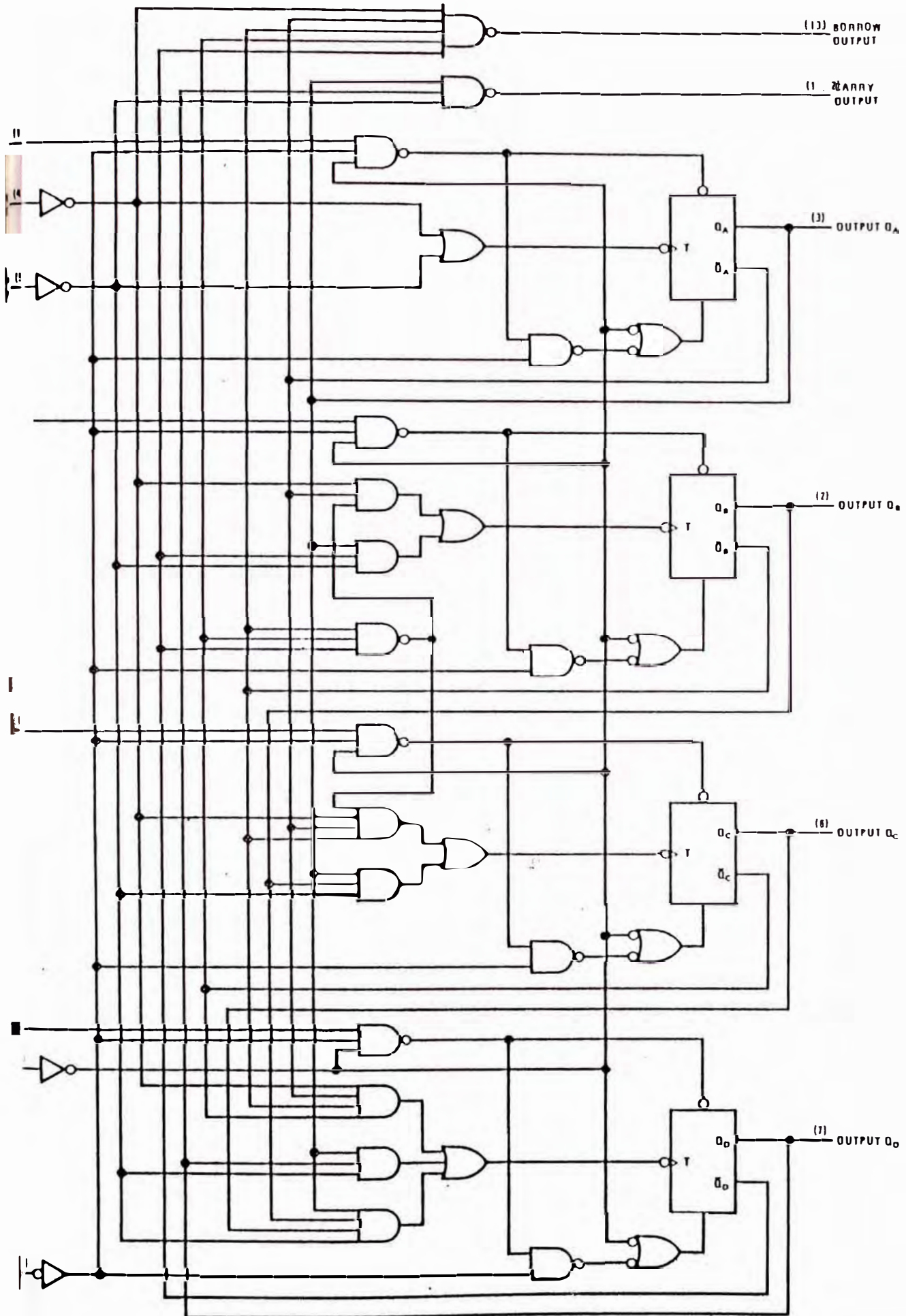
Switching Characteristics $V_{CC} = 5V$, $T_A = 25^\circ C$

PARAMETER		FROM INPUT	TO OUTPUT	ECG74								UNITS	
				192, 193				LS192, LS193					
				CONDITIONS	MIN	TYP	MAX	CONDITIONS	MIN	TYP	MAX		
f_{MAX}	Maximum Clock Frequency				20	25				25	32		MHz
t_{PLH}	Propagation Delay Time, Low-to-High Level Output	Count up	Carry	$C_L = 15 \text{ pF}$ $R_L = 400\Omega$		17	26	$C_L = 15 \text{ pF}$ $R_L = 2 \text{ k}\Omega$		17	26	ns	
t_{PHL}	Propagation Delay Time, High-to-Low Level Output					16	24			21	33	ns	
t_{PLH}	Propagation Delay Time, Low-to-High Level Output	Count down	Borrow			16	24			16	24	ns	
t_{PHL}	Propagation Delay Time, High-to-Low Level Output					16	24			21	33	ns	
t_{PLH}	Propagation Delay Time, Low-to-High Level Output	Either Count	Q			25	38			25	38	ns	
t_{PHL}	Propagation Delay Time, High-to-Low Level Output					31	47			31	47	ns	
t_{PLH}	Propagation Delay Time, Low-to-High Level Output	Load	Q			27	40			27	40	ns	
t_{PHL}	Propagation Delay Time, High-to-Low Level Output					29	40			29	40	ns	
t_{PHL}	Propagation Delay Time, High-to-Low Level Output	Clear	Q			22	35			22	35	ns	
t_W	Width of Any Input Pulse					25					20		ns
t_{SETUP}	Data Setup Time				20				20		ns		
t_{HOLD}	Data Hold Time				0				0		ns		

ECG74192, 74LS192, 74193, 74LS19

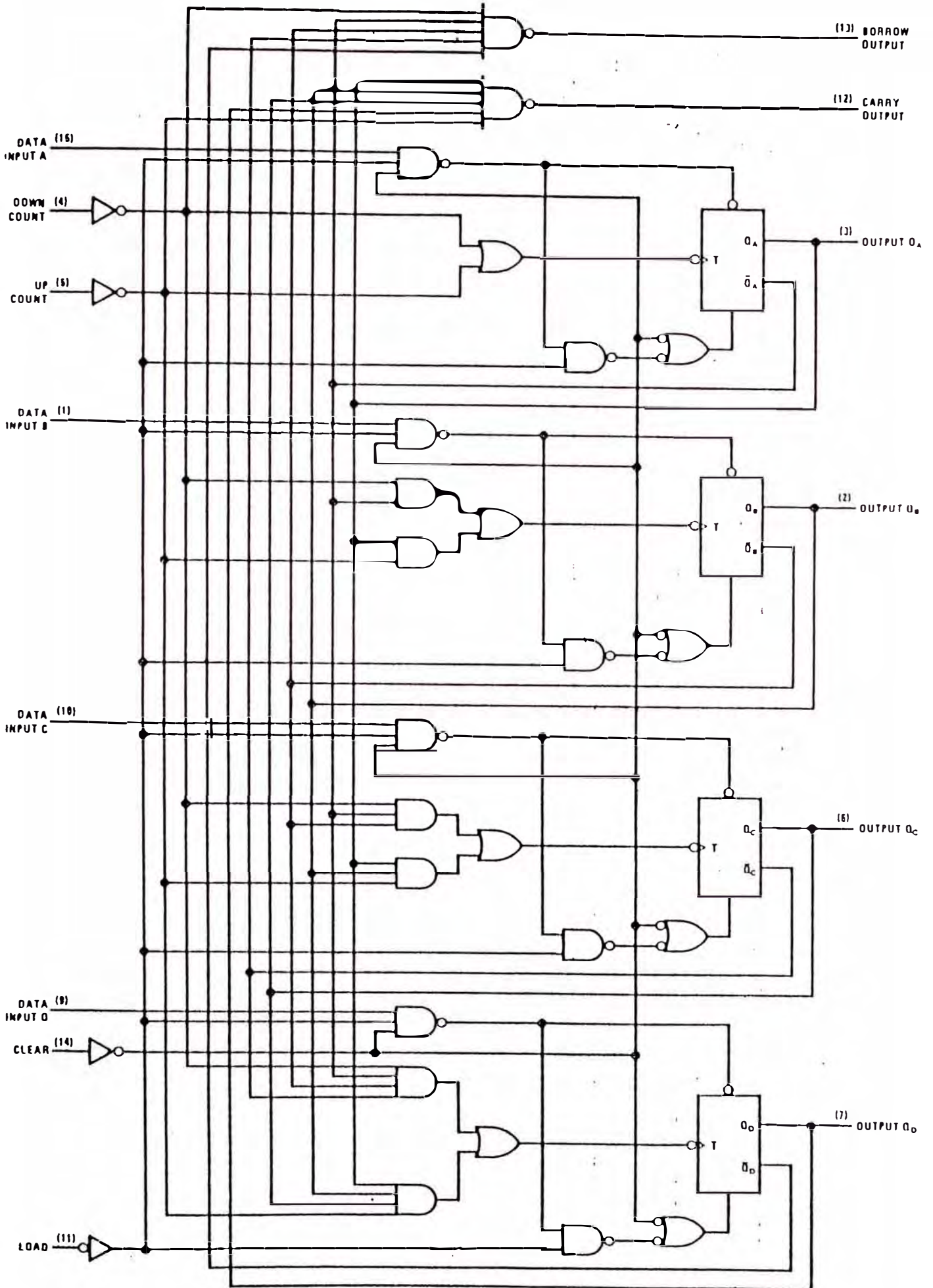
09 1715

192, LS192

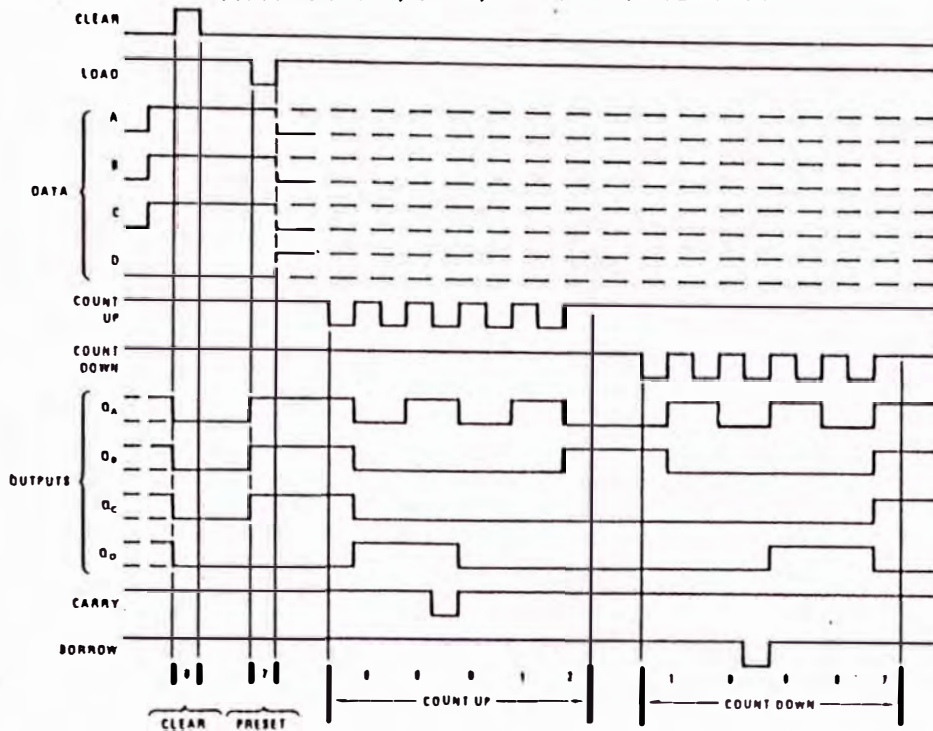


Logic Diagrams (Continued)

193, LS193

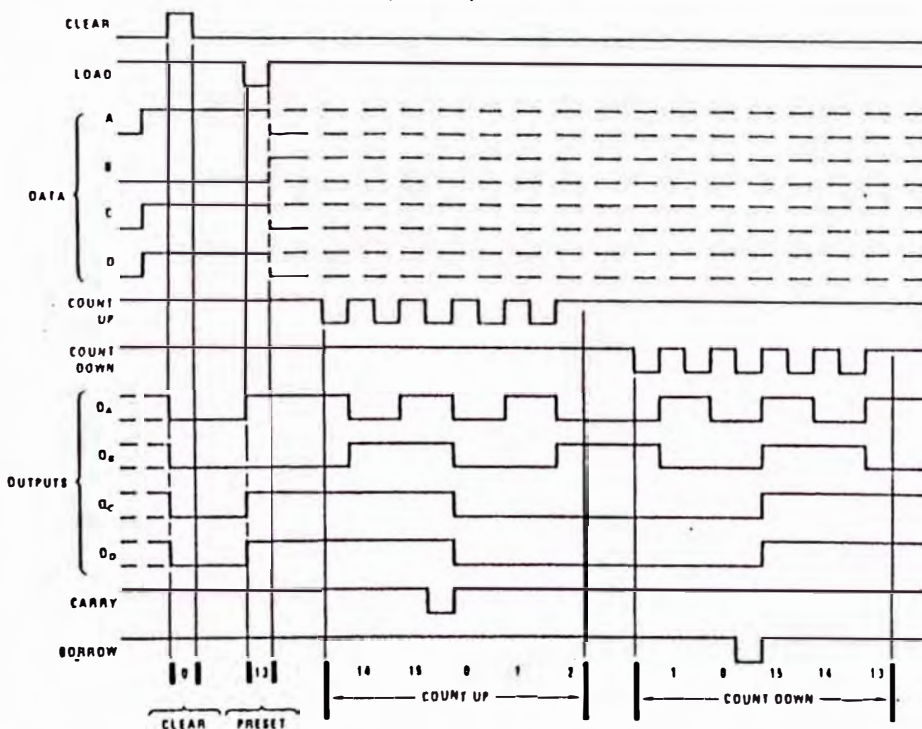


192, LS192 DECADE COUNTERS
TYPICAL CLEAR, LOAD, AND COUNT SEQUENCES



- Sequence:
- (1) Clear output to zero.
 - (2) Load (preset) to BCD seven.
 - (3) Count up to eight, nine, carry, zero, one, and two.
 - (4) Count down to one, zero, borrow, nine, eight, and seven.

193, LS193 BINARY COUNTERS
TYPICAL CLEAR, LOAD, AND COUNT SEQUENCES



- Sequence:
- (1) Clear outputs to zero.
 - (2) Load (preset) to binary thirteen.
 - (3) Count up to fourteen, fifteen, carry, zero, one, and two.
 - (4) Count down to one, zero, borrow, fifteen, fourteen, and thirteen.

- Notes:
- (A) Clear overrides load, data, and count inputs.
 - (B) When counting up, count-down input must be high; when counting down, count-up input must be high.

Buffers and Line Drivers with 3-State Outputs

These buffers and line drivers are designed to improve both the performance and reliability of memory address drivers, clock and bus-oriented receivers and transmitters. They offer a choice of selected combinations of noninverting outputs, symmetrical \bar{G} (output control) inputs, and complementary \bar{G} inputs. These devices feature high speed fan-in, and 400-mV noise-margin. They can be used to drive terminated lines.

Outputs Drive Bus Lines
Useful for Memory Address Registers

Inputs Reduce D-C Loading

Low Noise at Inputs Improves Noise Margins

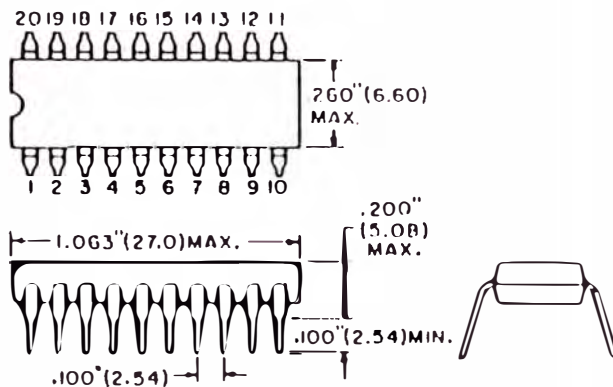
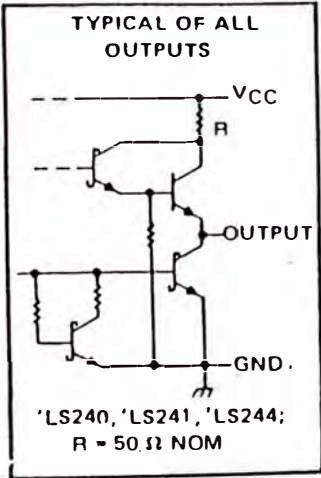
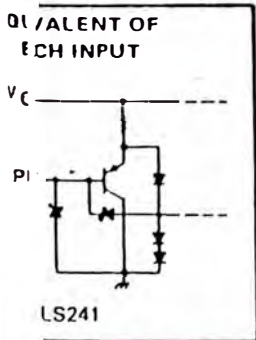
Typical Input Current (Source)	Typical Propagation Delay Times		Typical Enable/Disable Times	Typical Power Dissipation (Enabled)	
	Inverting	Noninverting		Inverting	Noninverting
-1 mA	10.5 ns	12 ns	18 ns	130 mW	135 mW

Standard operating conditions

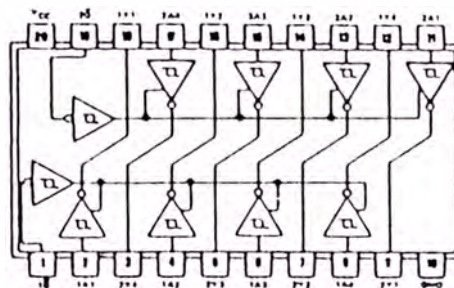
PARAMETER	MIN	NOM	MAX	UNIT
	Supply Voltage, V_{CC} (see Note 1)	4.75	5	
Output current, I_{OH}			-15	mA
Output current, I_{OL}			24	mA
Air temperature, T_A	0		70	$^{\circ}$ C

Output values are with respect to network ground terminal.

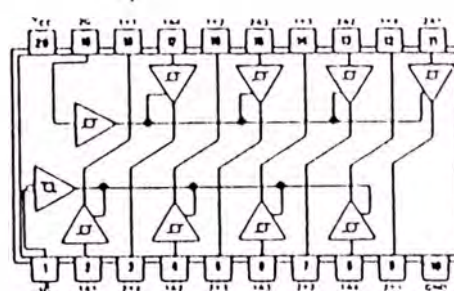
Typical inputs and outputs



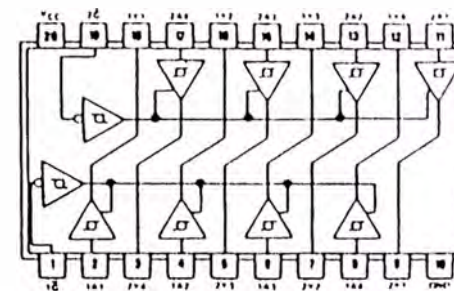
ECG74LS240-Inverting



ECG74LS241-Non-Inverting (2G)



ECG74LS244-Inverting (2 \bar{G})

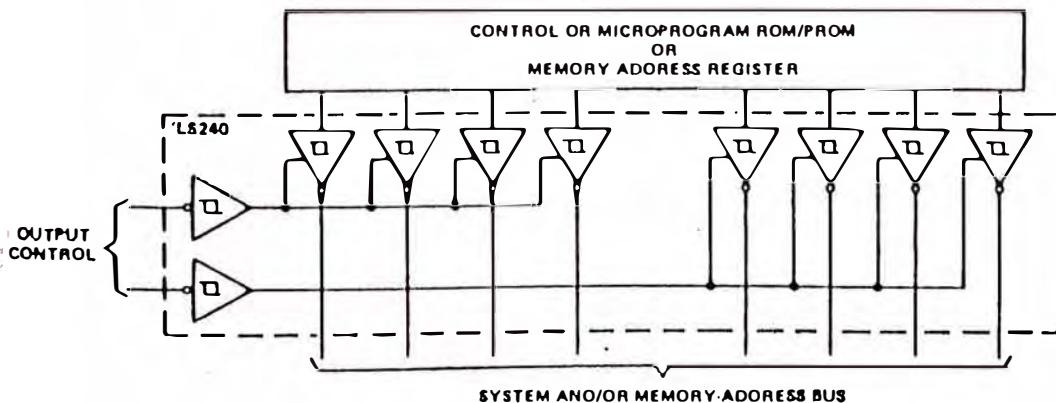
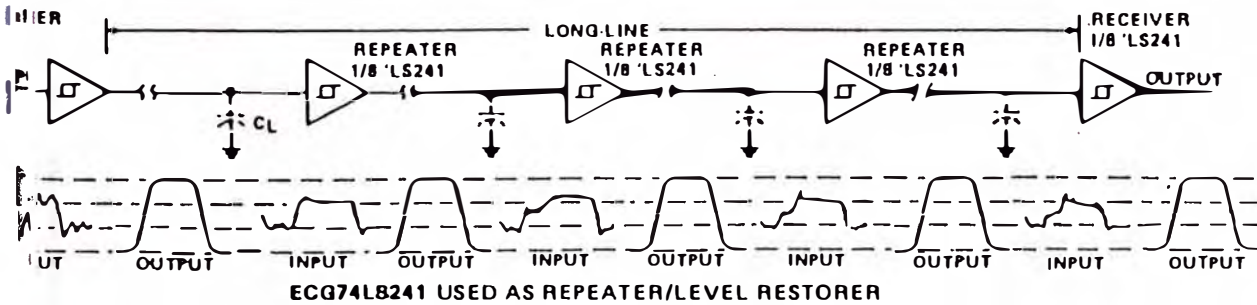


Static characteristics

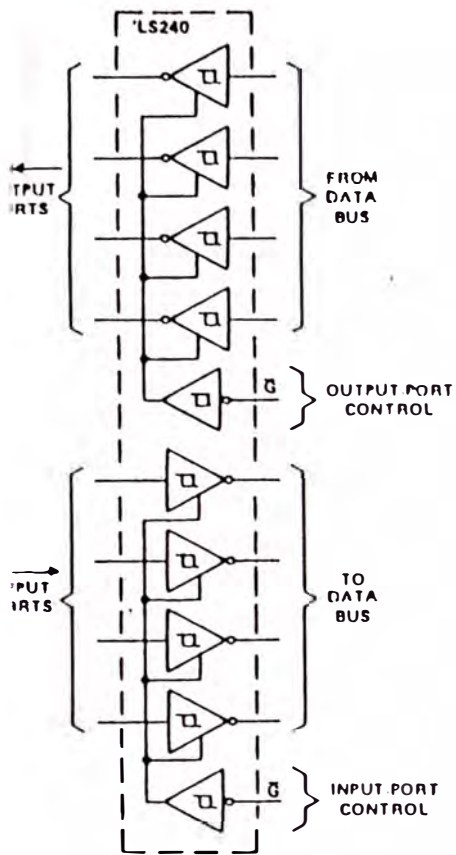
PARAMETER	TEST CONDITIONS	ECG74LS'			UNIT	
		MIN	TYP	MAX		
High-level input voltage		2			V	
Low-level input voltage				0.8	V	
Input clamp voltage	$V_{CC} = \text{MIN}, I_I = -18 \text{ mA}$			-1.5	V	
Hysteresis ($V_{T+} - V_{T-}$)	$V_{CC} = \text{MIN}$	0.2	0.4		V	
High-level output voltage	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = 0.8 \text{ V}, I_{OH} = \text{MAX}$	2.4	3.4		V	
	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = 0.5 \text{ V}, I_{OH} = \text{MAX}$	2				
Low-level output voltage	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, I_{OL} = 12 \text{ mA}$			0.4	V	
	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = V_{ILmax}, I_{OL} = 24 \text{ mA}$			0.5		
Off-state output current, high-level voltage applied	$V_{CC} = \text{MAX}, V_{IH} = 2 \text{ V}, V_O = 2.7 \text{ V}$			20	μA	
Off-state output current, low-level voltage applied	$V_{CC} = \text{MAX}, V_{IL} = V_{ILmax}, V_O = 0.4 \text{ V}$			-20		
Input current at maximum input voltage	$V_{CC} = \text{MAX}, V_I = 7 \text{ V}$			0.1	mA	
High-level input current, any input	$V_{CC} = \text{MAX}, V_I = 2.7 \text{ V}$			20	μA	
Low-level input current	$V_{CC} = \text{MAX}, V_{IL} = 0.4 \text{ V}$			-0.2	mA	
Short-circuit output current	$V_{CC} = \text{MAX}$	-50		-225	mA	
Supply current	Outputs high	$V_{CC} = \text{MAX}$	All	13	23	mA
	Outputs low		'LS240	26	44	
	All outputs disabled	Outputs open	'LS241, 'LS244	27	46	
			'LS240	29	50	
		'LS241, 'LS244	32	54		

Timing characteristics, $V_{CC} = 5 \text{ V}, T_A = 25^\circ \text{C}$

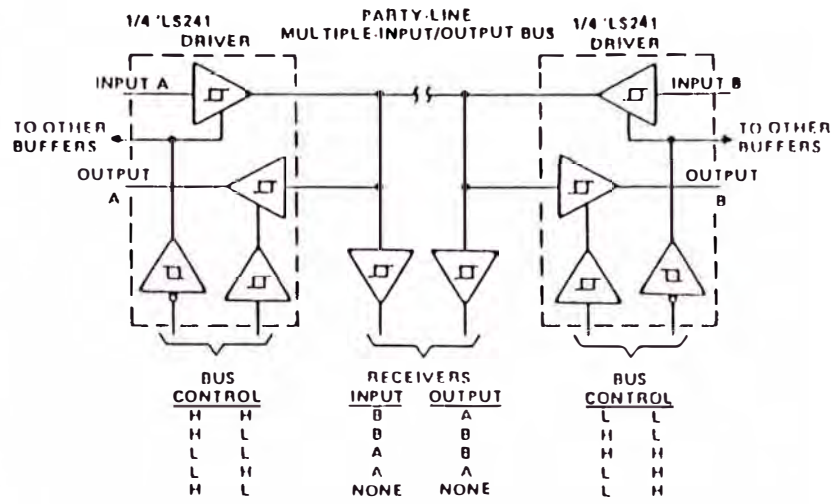
PARAMETER	TEST CONDITIONS	'LS240			'LS241, 'LS244			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
Propagation delay time, low-to-high-level output	$C_L = 45 \text{ pF}, R_L = 687 \Omega$		9	14	12	18	ns	
Propagation delay time, high-to-low-level output			12	18	12	18	ns	
Output enable time to low level			20	30	20	30	ns	
Output enable time to high level	$C_L = 5 \text{ pF}, R_L = 687 \Omega$		15	23	15	23	ns	
Output disable time from low level			15	25	15	25	ns	
Output disable time from high level			10	18	10	18	ns	



ECG74LS241 USED AS SYSTEM AND/OR MEMORY BUS DRIVER—4-BIT ORGANIZATION CAN BE APPLIED TO HANDLE BINARY OR BCD



**INDEPENDENT 4-BIT BUS DRIVERS/RECEIVERS
IN A SINGLE PACKAGE**



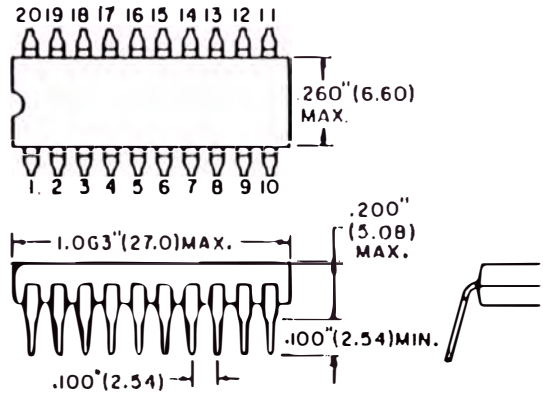
**PARTY-LINE BUS SYSTEM
WITH MULTIPLE INPUTS, OUTPUTS, AND RECEIVERS**

TTL D-TYPE FLIP-FLOP WITH CLEAR

This monolithic, positive-edge-triggered flip-flop utilizes TTL circuitry to implement D-type flip-flop logic with a direct clear input.

Information at the D inputs meeting the setup time requirements is transferred to the Q outputs on the positive-going edge of the clock pulse. Clock triggering occurs at a particular voltage level and is not directly related to the transition time of the positive-going pulse. When the clock input is at either the high or low level, the D input has no effect at the output.

This flip-flop is guaranteed to respond to clock frequencies ranging from 0 to 30 megahertz while maximum clock frequency is typically 10 megahertz. Typical power dissipation is 10 milliwatts per flip-flop.



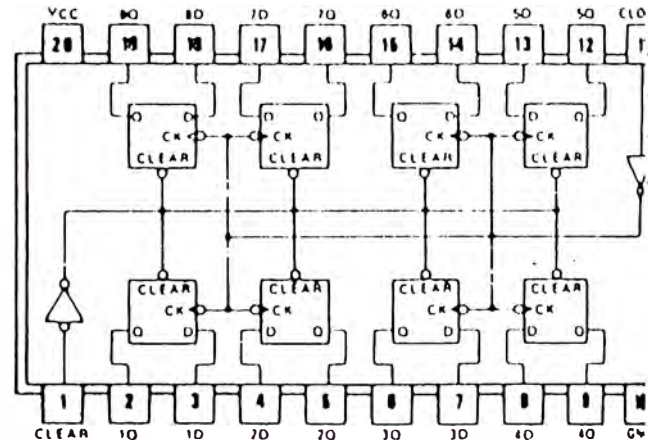
Contains Eight Flip-Flops with Single-Rail Outputs

Buffered Clock and Direct Clear Inputs

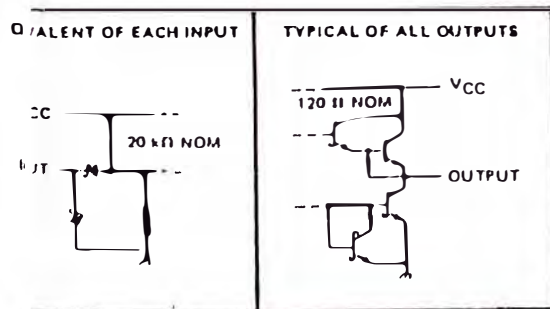
Individual Data Input to Each Flip-Flop

Applications Include:

- Buffer/Storage Registers
- Shift Registers
- Pattern Generators



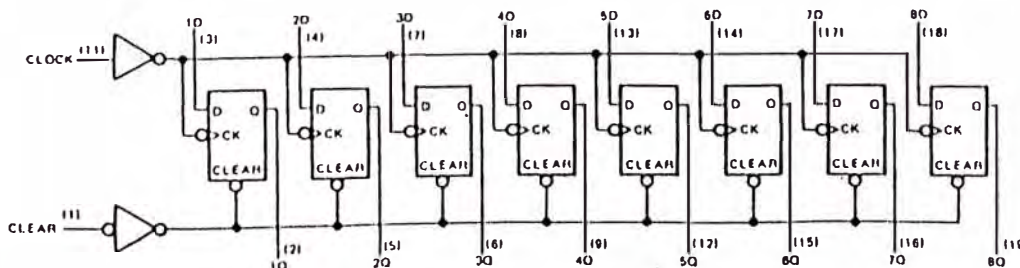
Timing of inputs and output



FUNCTION TABLE (EACH FLIP-FLOP)

INPUTS			OUTPUT
CLEAR	CLOCK	D	Q
L	X	X	L
H	↑	H	H
H	↑	L	L
H	L	X	Q ₀

Functional block diagram



Maximum ratings

Supply voltage, V _{CC} (see Note 1)	7 V
Input voltage	7 V
Operating free-air temperature range:	0°C to 70°C
Storage temperature range	65°C to 150°C

† Voltage values are with respect to network ground terminal

Permitted operating conditions

		74LS273			UNIT
		MIN	NOM	MAX	
Supply voltage, V _{CC}		4.75	5	5.25	V
Maximum output current, I _{OH}				-400	μA
Maximum output current, I _{OL}				8	mA
Maximum frequency, f _{clock}		0		30	MHz
Maximum clock or clear pulse, t _w		20			ns
Setup time, t _{su}	Data input	20			ns
	Clear Inactive state	25			ns
Hold time, t _h		5			ns
Maximum free-air temperature, T _A		0		70	°C

†† Indicates that the rising edge of the clock pulse is used for reference.

Static characteristics

PARAMETER	TEST CONDITIONS	74LS273			UNIT
		MIN	TYP	MAX	
High-level input voltage		2			V
Low-level input voltage				0.8	V
Input clamp voltage	V _{CC} = MIN, I _I = -18 mA			-1.5	V
High-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = V _{ILmax} , I _{OH} = -400 μA	2.7	3.4		V
Low-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = V _{ILmax}	I _{OL} = 4 mA	0.25	0.4	V
		I _{OL} = 8 mA	0.35	0.5	
Input current at maximum input voltage	V _{CC} = MAX, V _I = 7 V			0.1	mA
High-level input current	V _{CC} = MAX, V _I = 2.7 V			20	μA
Low-level input current	V _{CC} = MAX, V _I = -0.4 V			-0.4	mA
Short-circuit output current*	V _{CC} = MAX	-20		-100	mA
Supply current	V _{CC} = MAX, See Note 2.		17	27	mA

* When one output should be shorted at a time and duration of short circuit should not exceed one second.

† With all outputs open and 4.5 V applied to all data and clear inputs, I_{CC} is measured after a momentary ground, then 4.5 V is applied to clock.

Propagation characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Minimum clock frequency		30	40		MHz
Propagation delay time, high-to-low-level output from clear	C _L = 15 pF, R _L = 2 kΩ,		18	27	ns
Propagation delay time, low-to-high-level output from clock			17	27	ns
Propagation delay time, high-to-low-level output from clock			18	27	ns

FAST 74F373, 74F374

Latch/Flip-Flop

FAST Products

FEATURES

- 8-bit transparent latch-'F373
- 8-bit positive edge triggered register-'F374
- 3-State Outputs glitch free during power-up and power-down
- Common 3-state Output register
- Independent register and 3-state buffer operation

DESCRIPTION

The 74F373 is an octal transparent latch coupled to eight 3-State output devices. The two sections of the device are controlled independently by Enable (E) and Output Enable (\overline{OE}) control gates.

The data on the D inputs is transferred to the latch outputs when the Enable (E) input is High. The latch remains transparent to the data input while E is High, and stores the data that is present one set-up time before the High-to-Low enable transition.

The 3-State output buffers are designed to drive heavily loaded 3-State buses, MOS memories, or MOS microprocessors. The active Low Output Enable (\overline{OE}) controls all eight 3-State buffers independent of the latch operation. When \overline{OE} is Low, the latched or transparent data appears at the outputs.

74F373 Octal Transparent Latch (3-State)

74F374 Octal D Flip-Flop (3-State)

Product Specification

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F373	4.5ns	35mA
74F374	5.5ns	55mA

ORDERING INFORMATION

PACKAGES	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$; $T_A = 0^\circ C$ to $+70^\circ C$
20-Pin Plastic DIP	N74F373N, N74F374N
20-Pin Plastic SOL	N74F373D, N74F374D

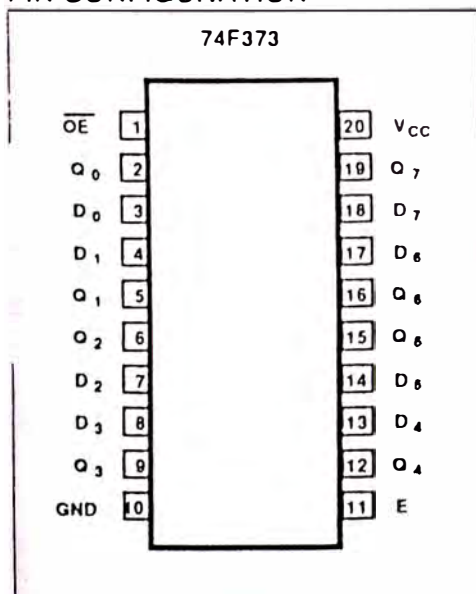
INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74F(U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
$D_0 - D_7$	Data inputs	1.0/1.0	20 μ A/0.6mA
E ('F373)	Enable input (active High)	1.0/1.0	20 μ A/0.6mA
\overline{OE}	Output Enable input (active Low)	1.0/1.0	20 μ A/0.6mA
CP ('F374)	Clock Pulse input (active rising edge)	1.0/1.0	20 μ A/0.6mA
$Q_0 - Q_7$	3-State outputs	150/40	3.0mA/24mA

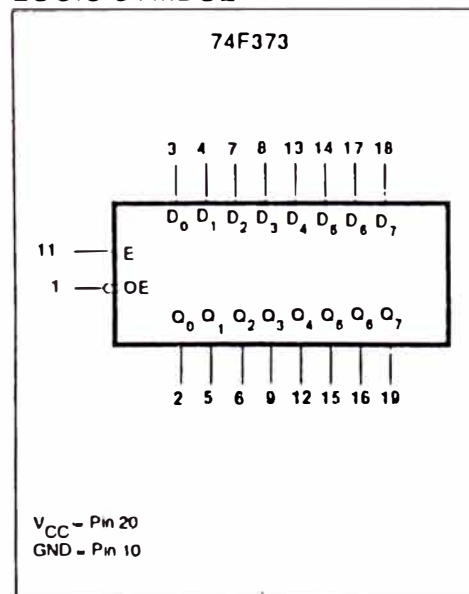
NOTE:

One (1.0) FAST Unit Load is defined as: 20 μ A in the High state and 0.6mA in the Low state

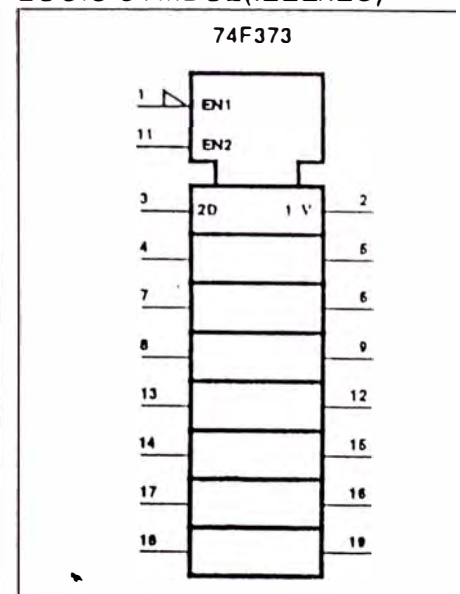
PIN CONFIGURATION



LOGIC SYMBOL

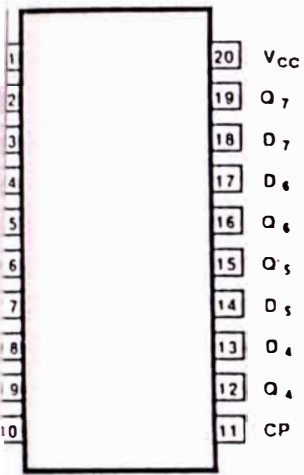


LOGIC SYMBOL (IEEE/IEC)



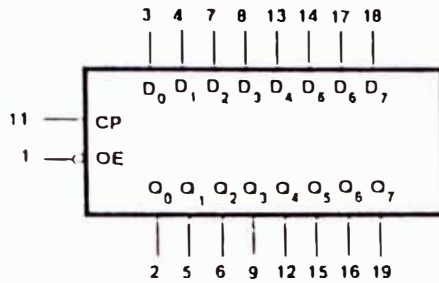
FIGURATION

74F374



LOGIC SYMBOL

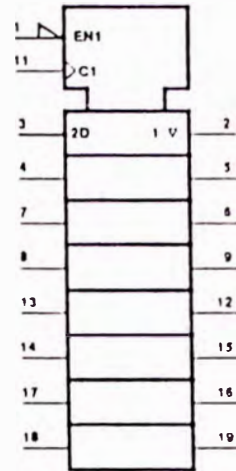
74F374



V_{CC} - Pin 20
GND - Pin 10

LOGIC SYMBOL (IEEE/IEC)

74F374



is High, the outputs are in high impedance "off" state, which means they will drive nor load the bus.

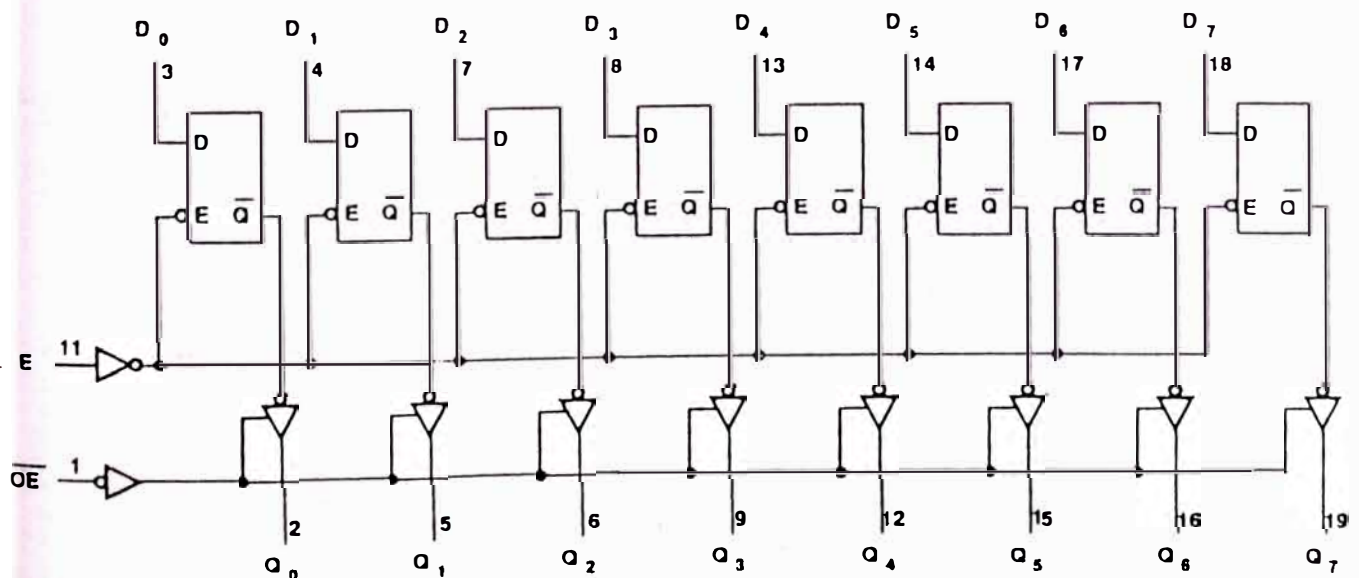
74F374 is an 8-bit, edge triggered register. The two sections of the device are controlled independently by the clock and Output Enable (\overline{OE}) control

The register is fully edge triggered. The state of each D input, one set-up time before the Low-to-High clock transition is transferred to the corresponding flip-flop's Q output.

The 3-State output buffers are designed to drive heavily loaded 3-State buses, MOS memories, or MOS microprocessors.

The active Low Output Enable (\overline{OE}) controls all eight 3-State buffers independent of the register operation. When \overline{OE} is Low, the data in the register appears at the outputs. When \overline{OE} is High, the outputs are in high impedance "off" state, which means they will neither drive nor load the bus.

DIAGRAM, 74F373

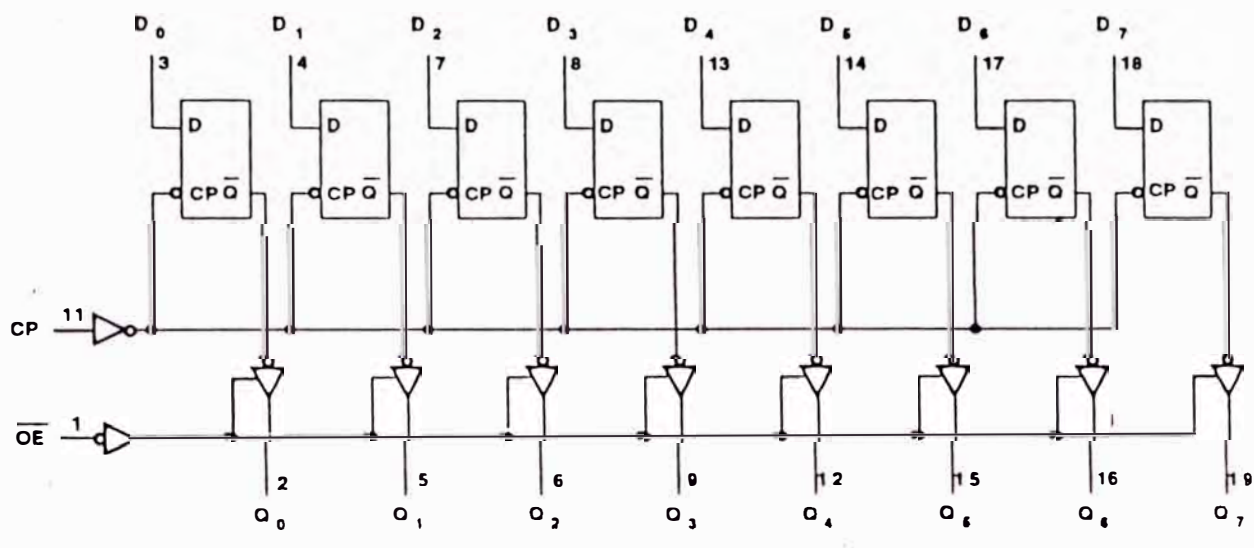


Pin 20
Pin 10

atch/Flip-Flop

FAST 74F373, 74F374

LOGIC DIAGRAM, 74F374



CC = Pin 20
 GND = Pin 10

FUNCTION TABLE, 74F373

INPUTS			INTERNAL REGISTER	OUTPUTS	OPERATING MODE
OE	E	D _n		Q ₀ - Q ₇	
L	H	L	L	L	Enable and read register
L	H	H	H	H	
L	↓	l	L	L	Latch and read register
L	↓	h	H	H	
L	L	X	NC	NC	Hold
H	L	X	NC	Z	Disable outputs
H	H	D _n	D _n	Z	

- H = High voltage level
- h = High voltage level one set-up time prior to the Low-to-High clock transition
- L = Low voltage level
- l = Low voltage level one set-up time prior to the Low-to-High clock transition
- NC = No change
- X = Don't care
- Z = High impedance "off" state
- ↓ = High-to-Low E transition

FUNCTION TABLE, 74F374

INPUTS			INTERNAL REGISTER	OUTPUTS	OPERATING MODE
OE	CP	D _n		Q ₀ - Q ₇	
L	↑	l	L	L	Load and read register
L	↑	h	H	H	
L	↑	X	NC	NC	Hold
H	↑	X	NC	Z	Disable outputs
H	↑	D _n	D _n	Z	

- H = High voltage level
- h = High voltage level one set-up time prior to the Low-to-High clock transition
- L = Low voltage level
- l = Low voltage level one set-up time prior to the Low-to-High clock transition
- NC = No change
- X = Don't care
- Z = High impedance "off" state
- ↑ = Low-to-High clock transition
- ↑ = Not a Low-to-High clock transition

Data/Flip-Flop

FAST 74F373, 74F374

SCHEMATIC MAXIMUM RATINGS (Operation beyond the limits set forth in this table may impair the useful life of the device. Unless otherwise noted these limits are over the operating free-air temperature range.)

PARAMETER	RATING	UNIT
Supply voltage	-0.5 to +7.0	V
Input voltage	-0.5 to +7.0	V
Input current	-30 to +5	mA
Voltage applied to output in High output state	-0.5 to +V _{CC}	V
Current applied to output in Low output state	48	mA
Operating free-air temperature range	0 to +70	°C
Storage temperature	65 to +150	°C

RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	LIMITS			UNIT
		Min	Nom	Max	
	Supply voltage	4.5	5.0	5.5	V
	High-level input voltage	2.0			V
	Low-level input voltage			0.8	V
	Input clamp current			-18	mA
I _C	High-level output current			-3	mA
I _C	Low-level output current			24	mA
T _A	Operating free-air temperature range	0		70	°C

ELECTRICAL CHARACTERISTICS (Over recommended operating free-air temperature range unless otherwise noted)

SYMBOL	PARAMETER	TEST CONDITIONS ¹	LIMITS			UNIT
			Min	Typ ²	Max	
V _O	High-level output voltage	V _{CC} = MIN, V _{IL} = MAX	±10%V _{CC}	2.4		V
		V _{IH} = MIN, I _{OH} = MAX	±5%V _{CC}	2.7	3.4	V
V _O	Low-level output voltage	V _{CC} = MIN, V _{IL} = MAX	±10%V _{CC}	0.35	0.50	V
		V _{IH} = MIN, I _{OL} = MAX	±5%V _{CC}	0.35	0.50	V
V _I	Input clamp voltage	V _{CC} = MIN, I _I = I _{IK}		-0.73	-1.2	V
I _I	Input current at maximum input voltage	V _{CC} = MAX, V _I = 7.0V			100	μA
I _I	High-level input current	V _{CC} = MAX, V _I = 2.7V			20	μA
I _I	Low-level input current	V _{CC} = MAX, V _I = 0.5V			-0.6	mA
I _O	Off-state output current, High-level voltage applied	V _{CC} = MAX, V _O = 2.7V			50	μA
I _O	Off-state output current, Low-level voltage applied	V _{CC} = MAX, V _O = 0.5V			-5.0	μA
I _{OS}	Short circuit output current ³	V _{CC} = MAX		-60	-150	mA
I _{CC}	Supply current (total)	'F373		35	60	mA
		'F374	V _{CC} = MAX		57	86

NOTES:

¹ Conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.

² Typical values are at V_{CC} = 5V, T_A = 25°C.

³ If more than one output should be shorted at a time. For testing I_{OS}, the use of high-speed test apparatus and/or sample-and-hold techniques are preferable in order to minimize internal heating and more accurately reflect operational values. Otherwise, prolonged shorting of a High output may raise the chip temperature well above normal and thereby cause invalid readings in other parameter tests. In any sequence of parameter tests, I_{OS} tests should be performed last.

ci/Flip-Flop

FAST 74F373, 74F374

ELECTRICAL CHARACTERISTICS

PARAMETER	TEST CONDITION	LIMITS					UNIT	
		T _A = +25°C V _{CC} = 5V C _L = 50pF R _L = 500Ω			T _A = 0°C to +70°C V _{CC} = 5V ±10% C _L = 50pF R _L = 500Ω			
		Min	Typ	Max	Min	Max		
Propagation delay D _n to O _n	74F373	Wavelform 3	3.0 2.0	5.3 3.7	7.0 5.0	3.0 2.0	8.0 6.0	ns
Propagation delay E to O _n		Wavelform 2	5.0 3.0	9.0 4.0	11.5 7.0	5.0 3.0	12.0 8.0	ns
Output Enable time to High or Low level		Wavelform 6 Wavelform 7	2.0 2.0	5.0 5.6	11.0 7.5	2.0 2.0	11.5 8.5	ns ns
Output Disable time to High or Low level		Wavelform 6 Wavelform 7	2.0 2.0	4.5 3.8	6.5 5.0	2.0 2.0	7.0 6.0	ns ns
Maximum Clock frequency	74F374	Wavelform 1	150	165		140		ns
Propagation delay CP to O _n		Wavelform 1	3.5 3.5	5.0 5.0	7.5 7.5	3.0 3.0	8.5 8.5	ns
Output Enable time to High or Low level		Wavelform 6 Wavelform 7	2.0 2.0	9.0 5.3	11.0 7.5	2.0 2.0	12.0 8.5	ns ns
Output Disable time to High or Low level		Wavelform 6 Wavelform 7	2.0 2.0	5.3 4.3	6.0 5.5	2.0 2.0	7.0 6.5	ns ns

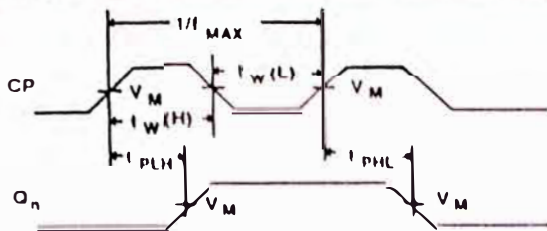
SETUP REQUIREMENTS

PARAMETER	TEST CONDITION	LIMITS					UNIT
		T _A = +25°C V _{CC} = 5V C _L = 50pF R _L = 500Ω			T _A = 0°C to +70°C V _{CC} = 5V ±10% C _L = 50pF R _L = 500Ω		
		Min	Typ	Max	Min	Max	
Set-up time D _n to E	74F373	Wavelform 4	0 1.0			0 1.0	ns
Hold time D _n to E		Wavelform 4	3.0 3.0			3.0 3.0	ns
E Pulse width, High	74F374	Wavelform 1	3.5			4.0	ns
Set-up time D _n to CP		Wavelform 5	2.0 2.0			2.0 2.0	ns
Hold time D _n to CP		Wavelform 5	0 0			0 0	ns
CP Pulse width, High or Low		Wavelform 1	3.5 4.0			3.5 4.0	ns

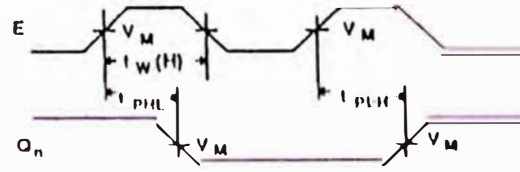
74F373/74F374

FAST 74F373, 74F374

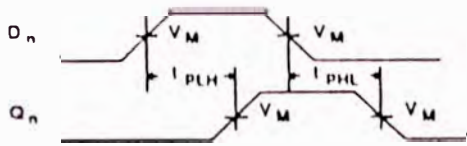
WAVEFORMS



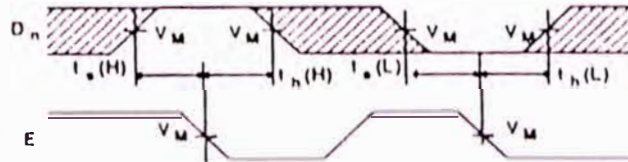
Waveform 1. Propagation Delay Clock To Output, Clock Pulse Widths, and Maximum Clock Frequency



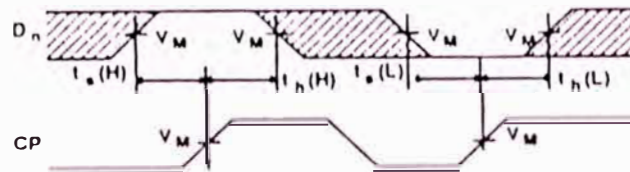
Waveform 2. Propagation Delay, Enable to Outputs And Enable Pulse Width



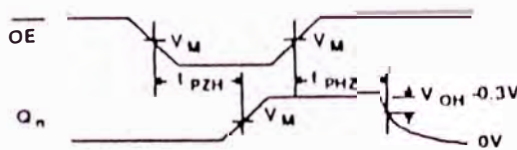
Waveform 3. Propagation Delay, Data To Outputs



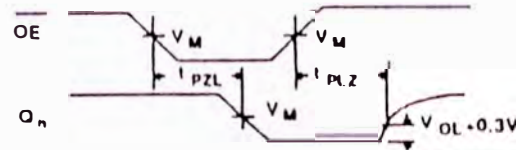
Waveform 4. Data Setup And Hold Times



Waveform 5. Data Setup And Hold Times



Waveform 6. 3-State Output Enable Time To High Level And Output Disable Time From High Level

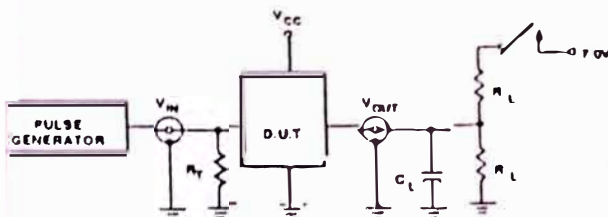


Waveform 7. 3-State Output Enable Time To Low Level And Output Disable Time From Low Level

NOTE: For all waveforms, $V_M = 1.5V$.

The shaded areas indicate when the input is permitted to change for predictable output performance.

TEST CIRCUIT AND WAVEFORMS



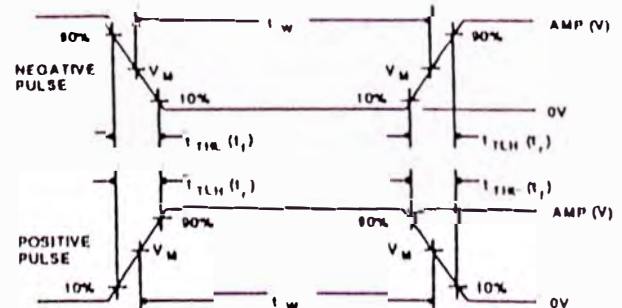
Test Circuit For 3-State Outputs

SWITCH POSITION

TEST	SWITCH
t_{PLZ}	closed
t_{PZL}	closed
All other	open

DEFINITIONS

- R_L = Load resistor; see AC CHARACTERISTICS for value.
- C_L = Load capacitance includes jig and probe capacitance; see AC CHARACTERISTICS for value.
- R_T = Termination resistance should be equal to Z_{OUT} of pulse generators.



$V_M = 1.5V$
Input Pulse Definition

FAMILY	INPUT PULSE REQUIREMENTS				
	Amplitude	Rep. Rate	t_w	$t_{TLH}(t_r)$	t_{THL}
74F	3.0V	1MHz	500ns	2.5ns	2.5ns

FAST 74F521

8-Bit Identity Comparator

Products

Features

Compares two 8-bit words in 7.0 ns typical
 Expandable to any word length
 High speed version of ALS688

Description

The 74F521 is an expandable 8-bit comparator that compares two words of up to 8 bits each and provides a Low output when two words match bit for bit. The expansion input \bar{I}_{A-B} also serves as an active-low enable input.

Comparator

Product Specification

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74F521	7.0ns	24mA

ORDERING INFORMATION

PACKAGES	COMMERCIAL RANGE
	$V_{CC} = 5V \pm 10\%$; $T_A = 0^\circ C$ to $+70^\circ C$
20-Pin Plastic DIP	N74F521N
20-Pin Plastic SO	N74F521D

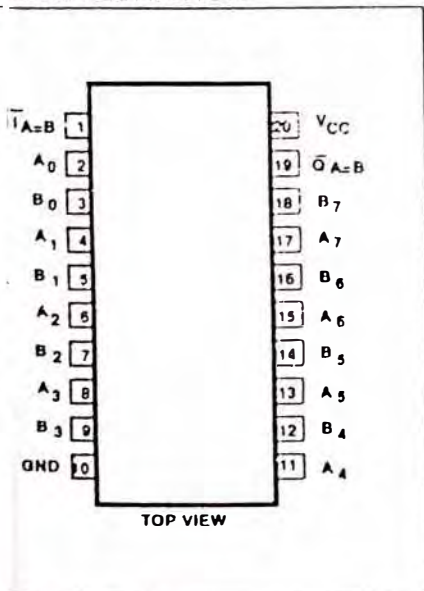
INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74F(U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
$A_0 - A_7$	Word A inputs	1.0/1.0	20 μ A/0.6mA
$B_0 - B_7$	Word B inputs	1.0/1.0	20 μ A/0.6mA
\bar{I}_{A-B}	Expansion or Enable input (active Low)	1.0/1.0	20 μ A/0.6mA
\bar{O}_{A-B}	Identity output (active Low)	50/33	1.0mA/20mA

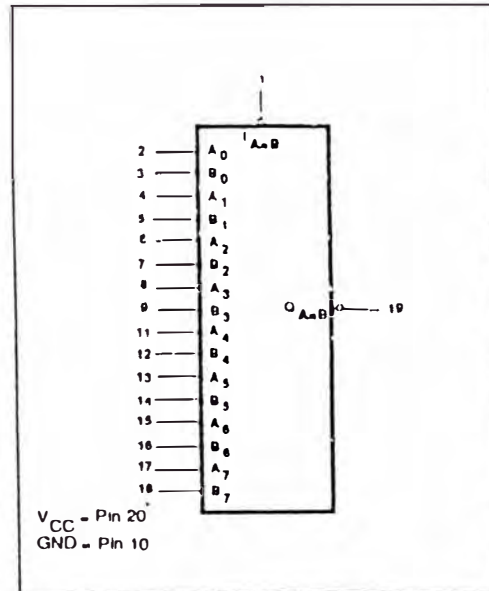
NOTE:

One (1.0) FAST Unit Load is defined as: 20 μ A in the High state and 0.6mA in the Low state

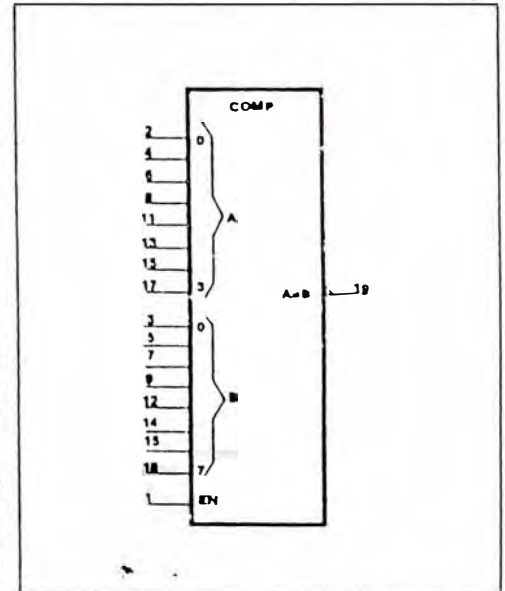
Pin Configuration



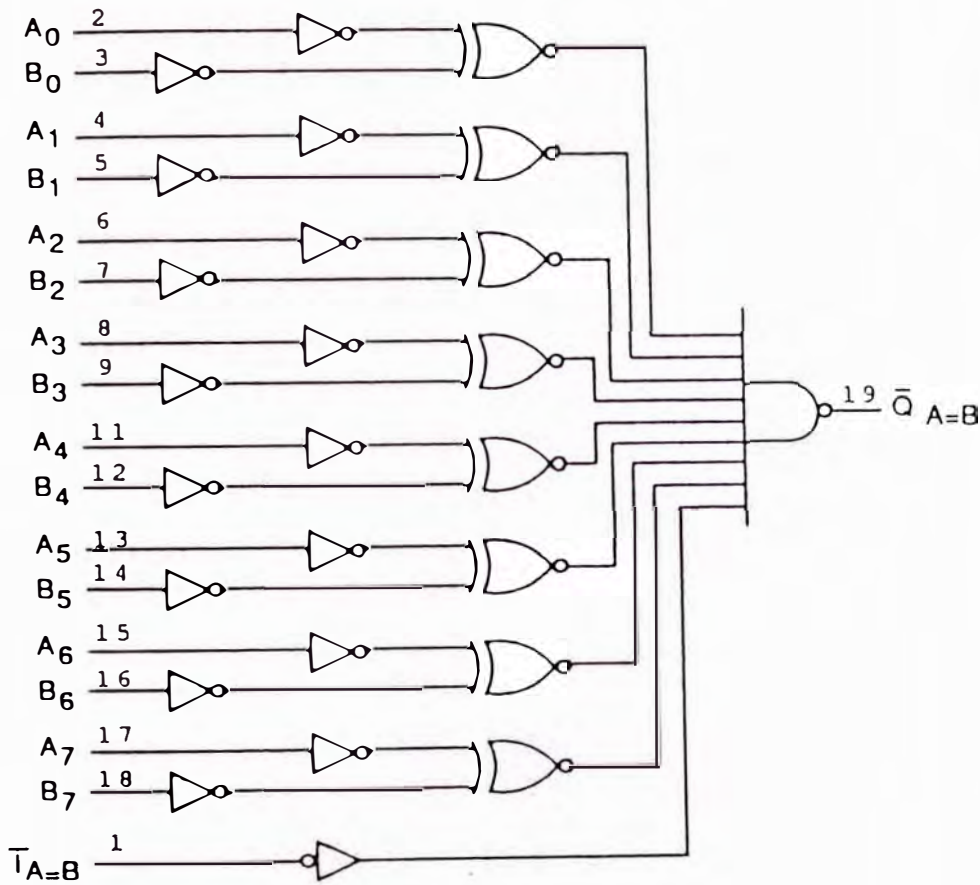
Logic Symbol



Logic Symbol (IEEE/IEC)



DIAGRAM



C - Pin 20
D - Pin 10

FUNCTION TABLE

INPUTS		OUTPUT
$\bar{T}_{A=B}$	A, B	$\bar{Q}_{A=B}$
L	$A=B^*$	L
L	$A \neq B$	H
H	$A=B^*$	H
H	$A \neq B$	H

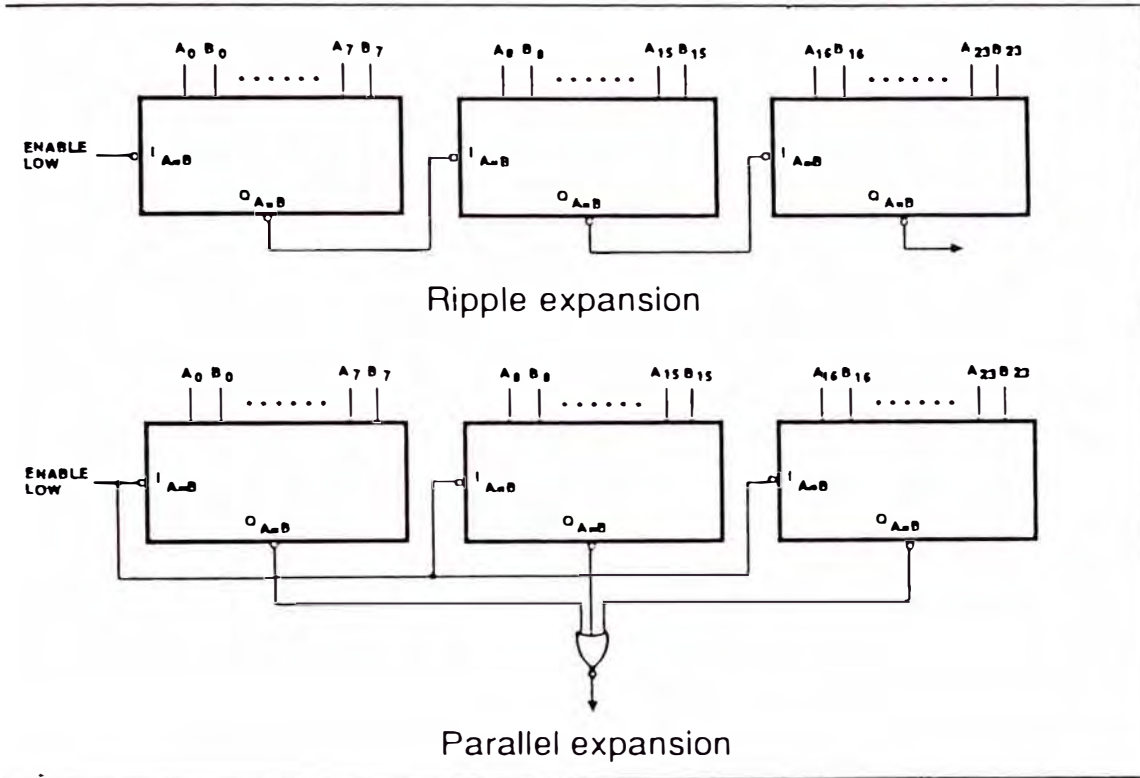
- High voltage level
- Low voltage level
- Don't care

$0=B_0, A_1=B_1, A_2=B_2, \text{ etc.}$

Comparator

FAST 74F521

APPLICATIONS



ABSOLUTE MAXIMUM RATINGS (Operation beyond the limits set forth in this table may impair the useful life of the device. Unless otherwise noted these limits are over the operating free-air temperature range.)

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Supply voltage	-0.5 to +7.0	V
V_{IN}	Input voltage	-0.5 to +7.0	V
I_{IN}	Input current	-30 to +5	mA
V_{OUT}	Voltage applied to output in High output state	-0.5 to $+V_{CC}$	V
I_{OUT}	Current applied to output in Low output state	40	mA
	Operating free-air temperature range	0 to +70	$^{\circ}C$
T_{STG}	Storage temperature	-65 to +150	$^{\circ}C$

RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	LIMITS			UNIT
		Min	Nom	Max	
V_{CC}	Supply voltage	4.5	5.0	5.5	V
V_{IH}	High-level input voltage	2.0			V
V_{IL}	Low-level input voltage			0.8	V
I_{IK}	Input clamp current			-18	mA
I_{OH}	High-level output current			-1	mA
I_{OL}	Low-level output current			20	mA
T_A	Operating free-air temperature range	0		70	$^{\circ}C$

ELECTRICAL CHARACTERISTICS (Over recommended operating free-air temperature range unless otherwise noted.)

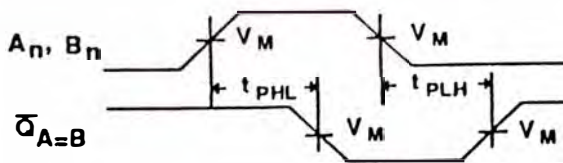
PARAMETER	TEST CONDITIONS ¹	LIMITS			UNIT	
		Min	Typ ²	Max		
High-level output voltage	$V_{CC} = \text{MIN}, V_{IL} = \text{MAX}$	$\pm 10\% V_{CC}$	2.5		V	
	$V_{IH} = \text{MIN}, I_{OH} = \text{MAX}$	$\pm 5\% V_{CC}$	2.7	3.4	V	
Low-level output voltage	$V_{CC} = \text{MIN}, V_{IL} = \text{MAX}$	$\pm 10\% V_{CC}$	0.30	0.50	V	
	$V_{IH} = \text{MIN}, I_{OL} = \text{MAX}$	$\pm 5\% V_{CC}$	0.30	0.50	V	
Input clamp voltage	$V_{CC} = \text{MIN}, I_I = I_{IK}$		-0.73	-1.2	V	
Input current at maximum input voltage	$V_{CC} = \text{MAX}, V_I = 7.0\text{V}$			100	μA	
High-level input current	$V_{CC} = \text{MAX}, V_I = 2.7\text{V}$			20	μA	
Low-level input current	$V_{CC} = \text{MAX}, V_I = 0.5\text{V}$			-0.6	mA	
Short circuit output current ³	$V_{CC} = \text{MAX}$		-60	-150	mA	
Supply current (total)	I_{CCH}	$V_{CC} = \text{MAX}$		24	36	mA
	I_{CCL}			24	36	mA

¹ Conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.
² Typical values are at $V_{CC} = 5\text{V}, T_A = 25^\circ\text{C}$.
³ More than one output should be shorted at a time. For testing I_{OS} , the use of high-speed test apparatus and/or sample-and-hold techniques are preferable in order to minimize internal heating and more accurately reflect operational values. Otherwise, prolonged shorting of a High output may raise the chip temperature above normal and thereby cause invalid readings in other parameter tests. In any sequence of parameter tests, I_{OS} tests should be performed last.

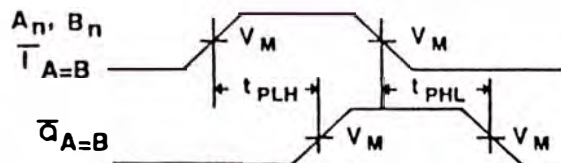
ELECTRICAL CHARACTERISTICS

PARAMETER	TEST CONDITION	LIMITS					UNIT
		$T_A = +25^\circ\text{C}$			$T_A = 0^\circ\text{C to } +70^\circ\text{C}$		
		Min	Typ	Max	Min	Max	
Propagation delay A_n or B_n to $\bar{Q}_{A=B}$	Waveform 1,2	$V_{CC} = 5\text{V}$ $C_L = 50\text{pF}$ $R_L = 500\Omega$					ns
		3.5	8.0	9.5	3.5	11.0	
Propagation delay $\bar{A}=B$ to $\bar{Q}_{A=B}$	Waveform 2	$V_{CC} = 5\text{V} \pm 10\%$ $C_L = 50\text{pF}$ $R_L = 500\Omega$					ns
		3.0	5.0	6.5	3.0	7.5	

WAVEFORMS



Waveform 1. For Inverting Outputs



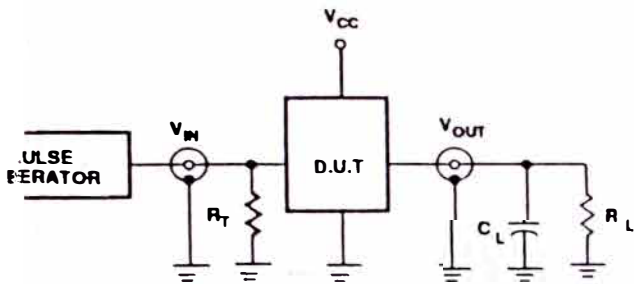
Waveform 2. For Non-Inverting Outputs

NOTE: For all waveforms, $V_M = 1.5\text{V}$

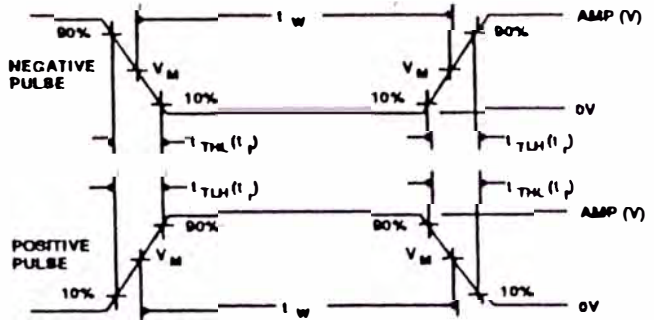
Comparator

FAST 74F521

TEST CIRCUIT AND WAVEFORMS



Test Circuit For Totem-Pole Outputs



$V_M = 1.5V$

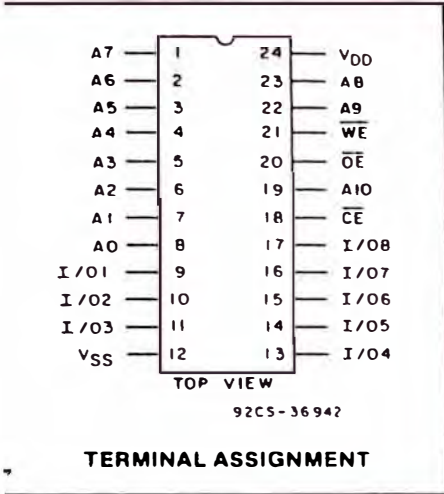
Input Pulse Definition

FAMILY	INPUT PULSE REQUIREMENTS				
	Amplitude	Rep. Rate	t_w	$t_{TLH}(10\%)$	$t_{TLH}(90\%)$
74F	3.0V	1MHz	500ns	2.5ns	2.5ns

DEFINITIONS

- R_L = Load resistor; see AC CHARACTERISTICS for value
- C_L = Load capacitance includes jig and probe capacitance; see AC CHARACTERISTICS for value.
- R_T = Termination resistance should be equal to Z_{OUT} of pulse generators.

CDM6116A



CMOS 2048-Word by 8-Bit Static RAM

Features:

- Fully static operation
- Single power supply: 4.5 V to 5.5 V
- All inputs and outputs directly TTL compatible
- 3-state outputs
- Industry standard 24-pin configuration
- Chip-enable gates address buffers for minimum standby current
- Data retention voltage: 2 V min.

	CDM6116A-2	CDM6116A-3	CDM6116A-9
Access Time (max.)	200 ns	150 ns	250 ns
Output Enable Time (max.)	120 ns	60 ns	150 ns
Operating Temperature	0° to +70° C		-40° to +85° C
Operating Current (max.)	35 mA	35 mA	40 mA
Standby Current I _{DDSI} (max.)	30 μA	50 μA	100 μA

The RCA-CDM6116A is a CMOS 2048-word by 8-bit static random-access memory. It is designed for use in memory systems where high-speed, low power and simplicity in use are desirable. This device has common data inputs and data outputs and utilizes a single power supply of 4.5 V to 5.5 V. A chip-enable input and an output-enable input are provided for memory expansion and output buffer control.

The chip enable (\overline{CE}) gates the address and output buffers and powers down the chip to the low power standby mode.

The output enable (\overline{OE}) controls the output buffers to eliminate bus contention.

The CDM6116A-2 and CDM6116A-3 have an operating temperature range of 0° to +70° C. The CDM6116A-9 has an operating temperature range of -40° to +85° C.

The CDM6116A-2 and CDM6116A-3 are supplied in a 24-lead dual-in-line plastic package (E suffix). The CDM6116A-9 is supplied in a 24-lead dual-in-line plastic package (E suffix) and a 24-lead dual-in-line side-braced ceramic package (D suffix).

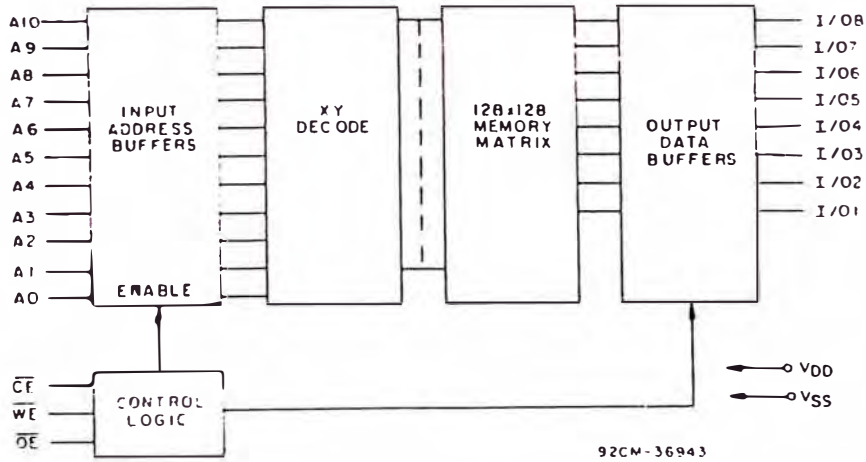


Fig. 1 - Functional block diagram.

TRUTH TABLE

\overline{CE}	\overline{OE}	\overline{WE}	A0 TO A10	MODE	I/O1 TO I/O8	DEVICE CURRENT
H	X	X	X	NOT SELECTED	HIGH Z	STANDBY
L	L	H	STABLE	READ	DATA OUT	ACTIVE
L	H	L	STABLE	WRITE	DATA IN	ACTIVE
L	L	L	STABLE	WRITE	DATA IN	ACTIVE

L = LOW H = HIGH X = H or L

CDM6116A

MAXIMUM RATINGS, Absolute-Maximum Ratings

SUPPLY-VOLTAGE RANGE, (V_{DD}):
 (with reference to V_{SS} terminal) -0.3 to +7 V

VOLTAGE RANGE, ALL INPUTS -0.3 to +7 V

INPUT CURRENT, ANY ONE INPUT ±10 mA

POWER DISSIPATION PER PACKAGE (P_D):
 T_A = -40° to +60° C (PACKAGE TYPE E) 500 mW
 T_A = +60° to +85° C (PACKAGE TYPE E) Derate Linearly at 8 mW/°C to 300 mW
 T_A = -40° to +85° C (PACKAGE TYPE D) 500 mW

POWER DISSIPATION PER OUTPUT TRANSISTOR 500 mW

TEMPERATURE RANGE (T_A):
 FULL PACKAGE-TEMPERATURE RANGE (All Package Types) 100 mW

OPERATING-TEMPERATURE RANGE (T_{OP}):
 CDM6116A-2, CDM6116A-3 (PACKAGE TYPE E) 0 to +70° C
 CDM6116A-9 (PACKAGE TYPES D, E) -40 to +85° C

STORAGE TEMPERATURE RANGE (T_{STG}) -55 to +125° C

TEMPERATURE (DURING SOLDERING):
 Distance 1/16 ± 1/32 in (1.59 ± 0.79 mm) from case for 10 s max -265° C

OPERATING CONDITIONS at T_A = 0 to +70° C. (CDM6116A-2, CDM6116A-3); T_A = -40° to +85° C (CDM6116A-9)

For maximum reliability, operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTIC	LIMITS ALL TYPES		UNITS
	MIN.	MAX.	
Operating Voltage Range	4.5	5.5	V
Voltage Range	V _{IH}	V _{DD} - 0.3	
	V _{IL}	-0.3	
Signal Rise or Fall Time Δ	t _r , t _f	5	μs

Signal rise and fall times longer than the maximum value can cause loss of stored data in the selected mode.

ELECTRICAL CHARACTERISTICS at T_A = 0 to +70° C (CDM6116A-2, CDM6116A-3);
 -40° to +85° C (CDM6116A-9), V_{DD} = 5 V ± 10%. Except as noted

CHARACTERISTIC	CONDITIONS	LIMITS									UNITS	
		CDM6116A-2			CDM6116A-3			CDM6116A-9				
		MIN.	TYP.*	MAX.	MIN.	TYP.*	MAX.	MIN.	TYP.*	MAX.		
Supply Device Current	I _{DD5}	CE = V _{IH}	—	0.6	2	—	0.6	2	—	0.3	2	mA
Standby Device Current	I _{DDS1}	CE = V _{DD} - 0.2 V	—	1	30	—	1	50	—	1	100	μA
Input Voltage Level	V _{OL} Max.	I _{OL} = 2.1 mA	—	—	0.4	—	—	0.4	—	—	0.4	V
		I _{OL} = 1 μA	—	0.1	—	—	0.1	—	—	0.1	—	
Output Voltage Level	V _{OH} Min.	I _{OH} = -1 mA	2.4	—	—	2.4	—	—	2.4	—	—	V
		I _{OH} = -1 μA	—	V _{DD} - 0.1	—	—	V _{DD} - 0.1	—	—	V _{DD} - 0.1	—	
Input Leakage Current	I _{IN} Max.	V _{DD} = 5.5 V	—	±0.1	±2	—	±0.1	±2	—	±0.1	±2	μA
		V _{IN} = 0 V to V _{DD}	—	±0.5	±2	—	±0.5	±2	—	±0.5	±2	
Output Leakage Current	I _{OUT}	CE or OE = V _{IH} V _{IO} = 0 V to V _{DD}	—	±0.5	±2	—	±0.5	±2	—	±0.5	±2	μA
Operating Current	I _{OPER} #	V _{IN} = V _{IL} , V _{IH}	—	20	35	—	20	35	—	28	40	mA
Input Capacitance	C _{IN}	V _{IN} = 0 V, f = 1 MHz, T _A = 25° C	—	4	6	—	4	6	—	4	6	pF
		V _{IO} = 0 V, f = 1 MHz, T _A = 25° C	—	6	8	—	6	8	—	6	8	

*All values are for T_A = 25° C and nominal V_{DD}.

#Outputs open circuited; cycle time = Min. t_{cycle}, duty = 100%.

CDM6116A

SIGNAL DESCRIPTIONS

- A0-A10 (Address Inputs):** These inputs must be stable prior to a write operation, but may change asynchronously during read operations
- I/O1-I/O8:** 8-bit tristate data bus.
- \overline{CE} (Chip Enable):** Powers down chip, disables Read and Write functions, and gates off address inputs.
- \overline{OE} (Output Enable):** Enables tristate outputs if \overline{CE} is low and \overline{WE} is high.
- \overline{WE} (Write Enable):** Enables Write function, if \overline{CE} is low. \overline{WE} will dominate if both \overline{WE} and \overline{OE} are low (i.e., the bus will be tristated and a Write will occur).
- V_{DD}, V_{SS} :** Power supply connections.

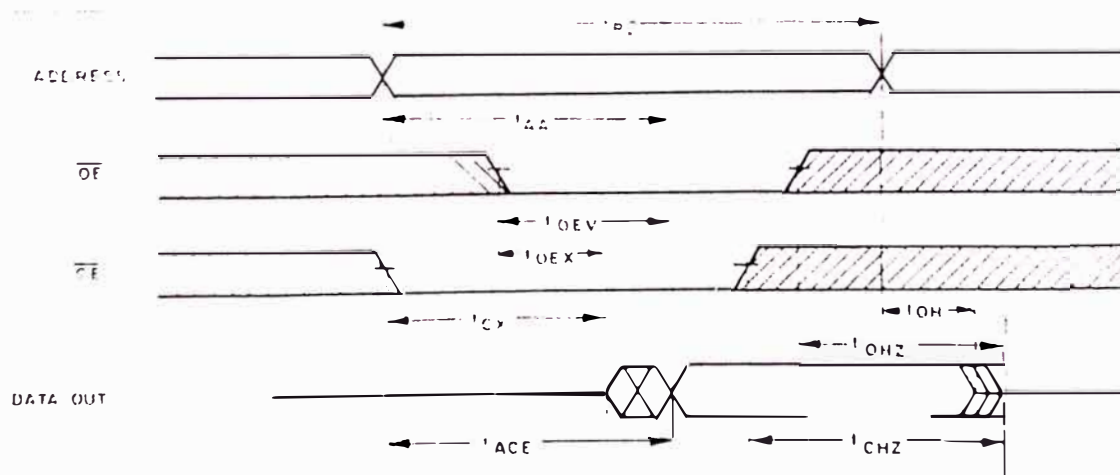
DYNAMIC ELECTRICAL CHARACTERISTICS at $T_A = 0$ to -70°C (CDM6116A-2, CDM6116A-3);

$T_A = -40^\circ$ to $+85^\circ\text{C}$ (CDM6116A-9), $V_{DD} = 5\text{V} \pm 10\%$.

Input $t_r, t_f = 10\text{ ns}$; $C_L = 100\text{ pF}$ and 1 TTL Load. Input Pulse Levels: 0.8 V to 2.4 V

CHARACTERISTIC		LIMITS						UNITS
		CDM6116A-2		CDM6116A-3		CDM6116A-9		
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Read Cycle Times See Fig. 2								
Read Cycle Time	t_{RC}	200	—	150	—	250	—	ns
Address Access Time	t_{AA}	—	200	—	150	—	250	
Chip Enable Access Time	t_{CE}	—	200	—	150	—	250	
Chip Enable to Output Active	t_{CA}	15	—	15	—	15	—	
Output Enable to Output Valid	t_{OE}	—	120	—	60	—	150	
Output Enable to Output Active	t_{OEA}	15	—	15	—	15	—	
Chip Disable to Output "High Z"	t_{CD}	0	60	0	50	0	80	
Output Disable to Output "High Z"	t_{OD}	0	60	0	50	0	80	
Output Hold from Address Change	t_{OH}	15	—	15	—	15	—	

*Time required by a limit device to allow for the indicated function



92CM-36944

Fig. 2 - Read-cycle timing waveforms.

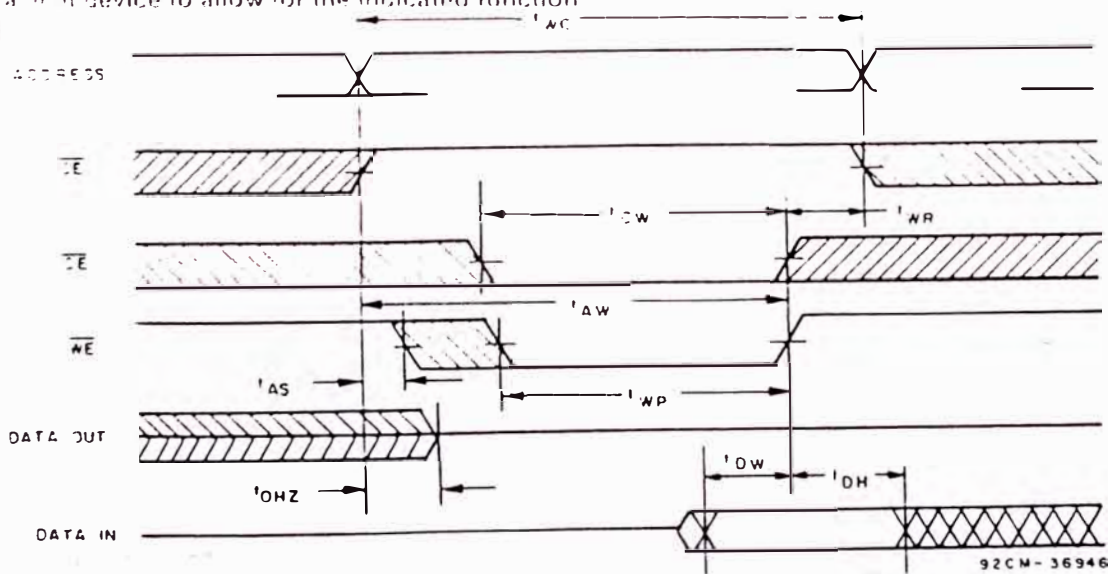
CDM6116A

W: ELECTRICAL CHARACTERISTICS at $T_A = 0$ to $+70^\circ\text{C}$ (CDM6116A-2, CDM6116A-3);
 to $+85^\circ\text{C}$ (CDM6116A-9), $V_{DD} = 5\text{ V} \pm 10\%$,
 $t_{setup} = 10\text{ ns}$; $C_L = 100\text{ pF}$ and 1 TTL Load, Input Pulse Levels: 0.8 V to 2.4 V

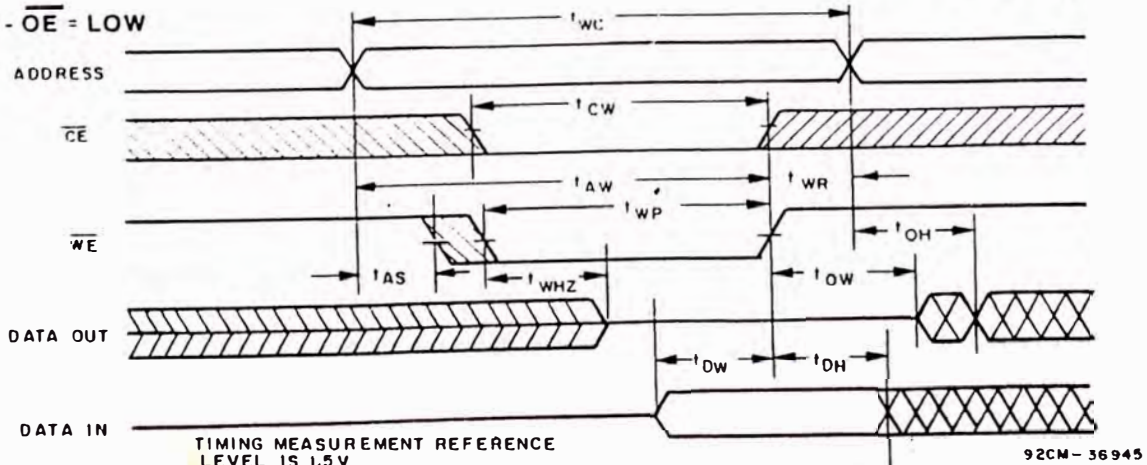
CHARACTERISTIC		LIMITS						UNITS
		CDM6116A-2		CDM6116A-3		CDM6116A-9		
		MIN. [†]	MAX.	MIN. [†]	MAX.	MIN. [†]	MAX.	
Write Cycle Time	t_{WC}	200	—	150	—	250	—	ns
Chip Enable to End of WRITE	t_{CEW}	160	—	90	—	200	—	
Write Enable to End of WRITE	t_{AWE}	160	—	90	—	200	—	
Write Setup Time	t_{AS}	0	—	0	—	0	—	
Chip Enable Width	t_{CEW}	160	—	90	—	200	—	
Chip Enable Recovery Time	t_{CEW}	10	—	0	—	10	—	
Chip Enable Disable to Output "High Z"	t_{CEW}	0	60	0	50	0	80	
Chip Enable to Output "High Z"	t_{CEW}	0	60	0	40	0	80	
Data Setup Time	t_{DWS}	80	—	50	—	100	—	
Data Hold Time	t_{DWH}	10	—	5	—	10	—	
Write Enable Active from End of Write	t_{AWE}	10	—	10	—	10	—	

† Measured by a unit device to allow for the indicated function

CYCLE 1



CYCLE 2 - $\overline{OE} = \text{LOW}$



TIMING MEASUREMENT REFERENCE LEVEL IS 1.5V

Fig. 3 - Write-cycle timing waveforms.

DM6116A

DATA RETENTION CHARACTERISTICS at $T_A = 0$ to 70°C (CDM6116A-2, CDM6116A-3);
 $T_A = -40$ to $+85^\circ\text{C}$ (CDM6116A-9), Unless otherwise noted, See Fig. 4.

CHARACTERISTIC	TEST CONDITIONS	LIMITS		UNITS
		ALL TYPES		
		MIN.	MAX.	
Minimum Data Retention Voltage V_{DR} CDM6116A-2, CDM6116A-3, CDM6116A-9	$T_A = 0$ to 70°C $\overline{CE} \geq V_{DD} - 0.2\text{ V}$	2	—	V
	CDM6116A-9 $T_A = -40$ to 0°C $\overline{CE} \geq V_{DD} - 0.2\text{ V}$	4.5	—	
Data Retention Quiescent Current I_{DDDR}^*	CDM6116A-2 $V_{DD} = 3\text{ V}, \overline{CE} \geq 2.8\text{ V}$	—	15	μA
	CDM6116A-3 $V_{DD} = 3\text{ V}, \overline{CE} \geq 2.8\text{ V}$	—	25	
	CDM6116A-9 $V_{DD} = 3\text{ V}, \overline{CE} \geq 2.8\text{ V}$	—	50	
Chip Disable to Data Retention Time t_{CDR}	See Fig 4	0	—	ns
Recovery to Normal Operation Time t_{in}	See Fig 4	t_{inc}	—	

$I_{DDDR} = 7.5\ \mu\text{A max}$ at $T_A = 0^\circ$ to -40°C for CDM6116A-2 and CDM6116A-3
 t_{inc} = Read Cycle Time.

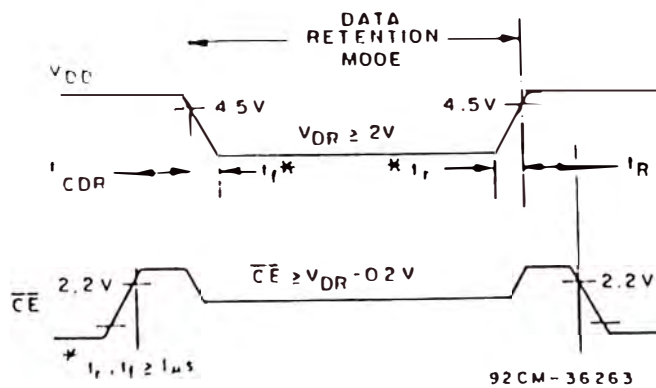


Fig 4 - Low V_{DD} data retention timing waveforms.