

ALUMNO: D. LUIS GUERRERO AROCA.

TUTOR: D. EDUARDO HERNANDEZ PEREZ.

INDICE.

INDICE

- INTRODUCCION.	-----	1
- TELEVISION MONOCROMA.	-----	5
A) Formación de $V(t)$.	-----	12
B) Señales Auxiliares $V_a(t)$.	-----	12
C) Señales de fin y supresión de línea	-----	13
C1) Caso ideal: Impulsos rectangulares	-----	13
C2) Caso real: Impulsos trapezoidales	-----	14
C3) Señales de fin de campo.	-----	15
- CIRCUITOS DE MUESTREO Y RETENCION.	-----	17
- CONVERTIDORES ANALOGICO DIGITALES.	-----	23
- Muestreo.	-----	25
- Cuantificación.	-----	29
- Codificación.	-----	34
- Clasificación general de los subsistemas		
convertidores A/D	-----	35
A) A/D de bucle abierto.	-----	36
A1) ADC con comparadores.	-----	36
A2) ADC analógico a ancho		
de impulso.	-----	38
A3) ADC de doble rampa.	-----	39
B) A/D realimentados o de bucle		
cerrado	-----	41

B1) ADC con contadores. -----	41
- Con rampa en escalera. -----	41
- De cuenta continua. -----	43
B2) ADC de aproximaciones	
sucesivas. -----	44
Especificaciones de los convertidores A/d -----	49
- CIRCUITO ELECTRICO. -----	53
A) Bloque de extracción de sincronismos. -----	58
B) Bloque de control. -----	62
C) Bloque de Sample and Hold y digitalización. -	65
D) Alimentación. -----	71
- PROGRAMACION EN ENSAMBLADOR. -----	72
A) Segmentación. -----	73
B) Registros del 8088. -----	74
B1) Registros de trabajo. -----	76
B2) Registros de Segmento. -----	77
B3) Registros relativos. -----	78
B4) Registros de estado. -----	79
C) Modos de direccionamiento. -----	80
C1) Direccionamiento inmediato. -----	80
C2) Direccionamiento de registro. -----	80
C3) Direccionamiento directo. -----	81
C4) Direccionamiento indirecto. -----	81
C5) Registro indirecto con base. -----	81
C6) Registro indirecto con indice. -----	81

C7) Registro indirecto con base e índice. -----	81
D) Interrupciones. -----	82
E) Instrucciones clasificadas por categorías. --	83
E1) Propósito general. -----	83
E2) E/S. -----	83
E3) Carga de direcciones. -----	83
E4) Almacenamiento y recuperación de indicadores. -----	84
E5) Aritméticas. -----	84
E6) Operaciones de cadena. -----	85
E7) Operaciones lógicas. -----	86
E8) Operaciones de rotación y desplazamiento. -----	86
E9) Operaciones de los indicadores. -----	86
E10) Control del procesador. -----	87
E11) Transderencia de control. -----	87
F) Introducción al macroensamblador para el microprocesador 8086 (MASM). -----	90
F1) Directivos DB,DW y DD. -----	92
F2) Directivo Segment/Ends. -----	93
F3) Directivo Assume. -----	94
F4) Etiquetas. -----	95
F5) Directivo END. -----	95
F6) Estructura de un programa. -----	95

- INTERFACE PARALELO. -----	98
A) Conector del interface. -----	100
B) Programación del interface. -----	103
- PROGRAMA. -----	105
A) Adquisición. -----	106
B) Adaptadores gráficos. Modos de video. -----	114
B1) Mapeado de los pixels. -----	118
C) Presentación en pantalla. -----	120
- PRESUPUESTO. -----	127

APENDICES

- A1) REALIZACION DEL CIRCUITO. -----	132
A) Placa de circuito impreso. -----	133
B) Componentes. -----	138
B1) Resistencias. -----	138
B2) Condensadores. -----	138
B3) Semiconductores. -----	139
B4) Circuitos Integrados. -----	139
B5) Varios. -----	140
- A2) VISTA DEL EQUIPO. -----	141
- A3) PROGRAMA COMPLETO. -----	143
- A4) CARACTERISTICAS DE CIRCUITOS EMPLEADOS. -----	158

INTRODUCCION .

INTRODUCCION

Una imagen es la representación de una escena y puede hacerse en papel o tela (dibujo o pintura), celuloide (fotografía) o elementos optoelectrónicos (televisión). Este concepto puede ampliarse para cubrir definiciones de las imágenes que no ^{se}corresponden con el concepto clásico que se tiene de ella.

Así se puede decir que el equivalente digital de una imagen analógica almacenada en la memoria de un ordenador es también, una imagen aunque no se puede ver.

En este trabajo se aborda el diseño y desarrollo del hardware y el software de un sistema para la adquisición, digitalización y almacenamiento de imágenes fijas en blanco y negro, procedentes de una cámara de TV o de un VTR, en un ordenador PC compatible.

Puesto que consideramos únicamente imágenes fijas, no se requiere su adquisición en tiempo real, es decir a la velocidad de barrido de la cámara de video (una imagen de 625 líneas cada 40 milisegundos). De este modo no se precisa la digitalización ultrarrápida de la imagen y su almacenamiento en un banco de memoria RAM externa rápida, previamente a su transferencia al ordenador, dado que en el tiempo real es difícil encontrar un interface que pueda realizar la transferencia de muestras digitalizadas al ordenador tan rápidamente. Por consiguiente, las muestras de la imagen se irán transfiriendo a la memoria central del ordenador a través del interface en el instante siguiente a

su adquisición y digitalización. Una vez en memoria, las imágenes se gestionarán en ficheros secuenciales ya que de esta forma se manipulan con gran sencillez.

Las posibles aplicaciones de este sistema están sujetas a la propia imaginación del usuario, ya que una vez digitalizada la imagen y en la memoria del ordenador, se le puede aplicar procedimientos y técnicas insospechadas. Entre las posibles aplicaciones, podemos indicar algunas:

-A modo de "album de imágenes", almacenandolas en discos duros o en disketes, teniendo la ventaja que no se degrada con el tiempo la información obtenida de una imagen. Estas imágenes podrían ser transmitidas a otros ordenadores distantes empleando cualquier sistema de transmisión de datos y sin necesidad de un gran ancho de banda del canal utilizado.

-Otra posible utilización con aplicación a la industria sería el reconocimiento de formas, aplicando a la imagen digitalizada un procedimiento adecuado. De esta forma se podrían analizar por ejemplo fallos en cadenas de producción. Se podrían calcular perímetros y superficies conociendo la distancia desde el objetivo de la cámara al objeto en cuestión.

Dividiremos en dos partes el desarrollo del trabajo.

En una primera parte trataremos algunos aspectos generales sobre los conocimientos y técnicas que se emplearán. Se hablará de las señales de video en blanco y negro, de circuitos "Sample and Hold", de convertidores

analógicos digitales, se dará una idea de programación en ensamblador, y por último, se verá algo del puerto paralelo e interface centrónica.

En la segunda parte se abordará el circuito eléctrico, su conexión al ordenador y el desarrollo del programa necesario para su control.

También se hablará de las limitaciones gráficas de un ordenador compatible PC, y como se sorteian dichas limitaciones.

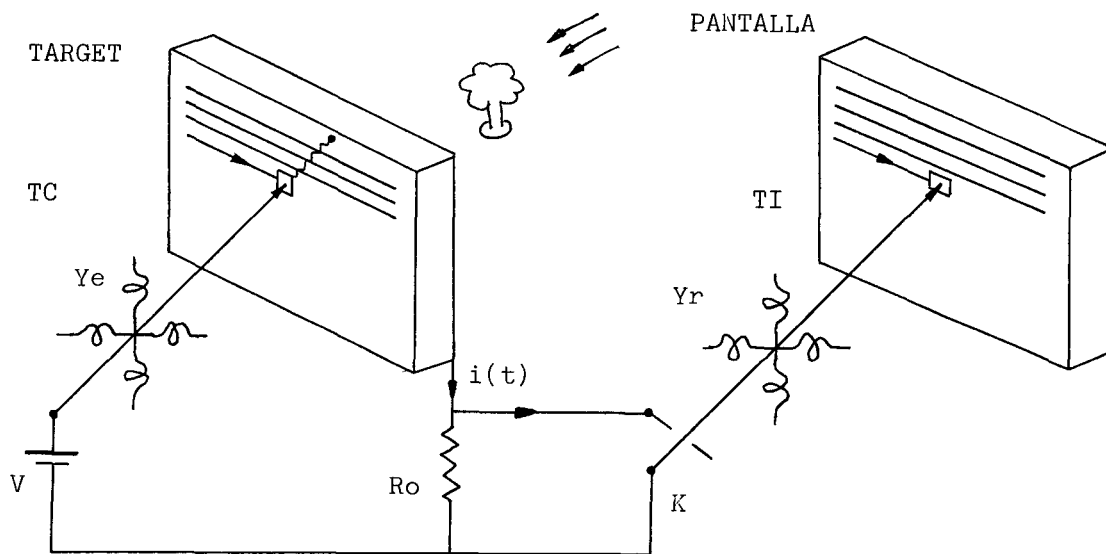
TELEVISION MONOCROMA.

TELEVISION MONOCROMA

El ojo humano es un instrumento óptico completo, reducido simplemente a tres elementos básicos: retina, nervio óptico y corteza visual. La visión es la interpretación consciente de un objeto iluminado por los fotones del espectro visible.

El ojo opera como un "minisistema de TV", y los esfuerzos realizados para lograr la televisión se han orientado en lo posible a tratar de imitarle; Así el mosaico, el canal de comunicación y la pantalla se corresponden respectivamente con la retina, nervio óptico y corteza visual.

El conjunto de todos los puntos de la escena se transmite en un tiempo inferior al de memoria del ojo, 40 milisegundos y constituye un cuadro del movimiento. Transmitiendo como en el aire un número suficiente de cuadros por segundo, se produce en la corteza visual, la visión animada de la escena. Basta transmitir una serie de 25 cuadros en el intervalo de 1 segundo para que resulte satisfactoriamente reproducido el movimiento de la escena. Para atenuar el efecto de centelleo, se divide el cuadro en 2 semicuadros (campos) de igual duración (20 milisegundos). Una primera aproximación al principio de funcionamiento del circuito y elementos básicos del sistema electrónico de TV en blanco y negro se muestra en la siguiente figura:



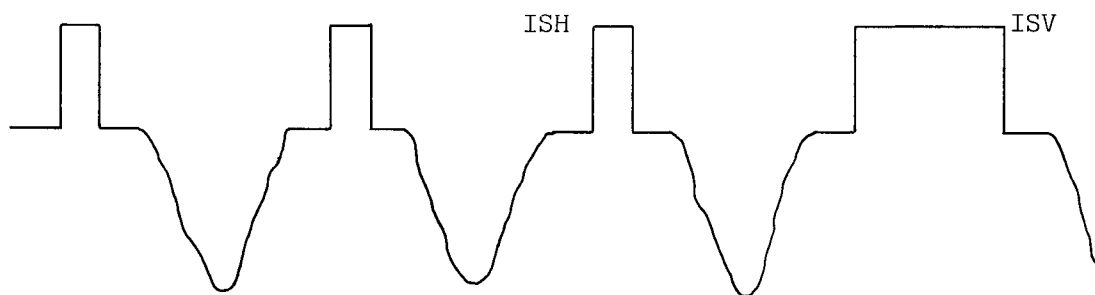
La luz incide sobre la escena que se desea transmitir. En general será policromática, pero sólo intervienen los valores de luminancia. En el lado emisor se encuentra el tubo de cámara (TC), reducido al mosaico o Target. Este será explorado por líneas producidas por un haz catódico sometido a la acción de un yugo emisor Y_e .

El mosaico o Target es una capa de material fotosensible sobre el que proyectada la escena, mediante el objetivo de la cámara, produce una "imagen resistiva". El valor r en cada punto es función de la luminancia incidente. Debido a la presencia de una fuente de tensión constante V y al desplazamiento del haz sobre el mosaico, se produce una corriente secuencial $i(t)$ que circulando por la resistencia de utilización R_o , establece en bornas la tensión $V(t)$ de información de luminancia.

Un generador de sincronismos permite que tenga lugar la aparición de cortos impulsos de sincronismo horizontal ISH, que determinan el fin de cada línea, y de largos impulsos de sincronismo vertical ISV, que marcan el fin de cada cuadro. Estos impulsos constituyen una señal auxiliar $V_a(t)$, necesaria para obtener el riguroso sincronismo entre los dos haces electrónicos.

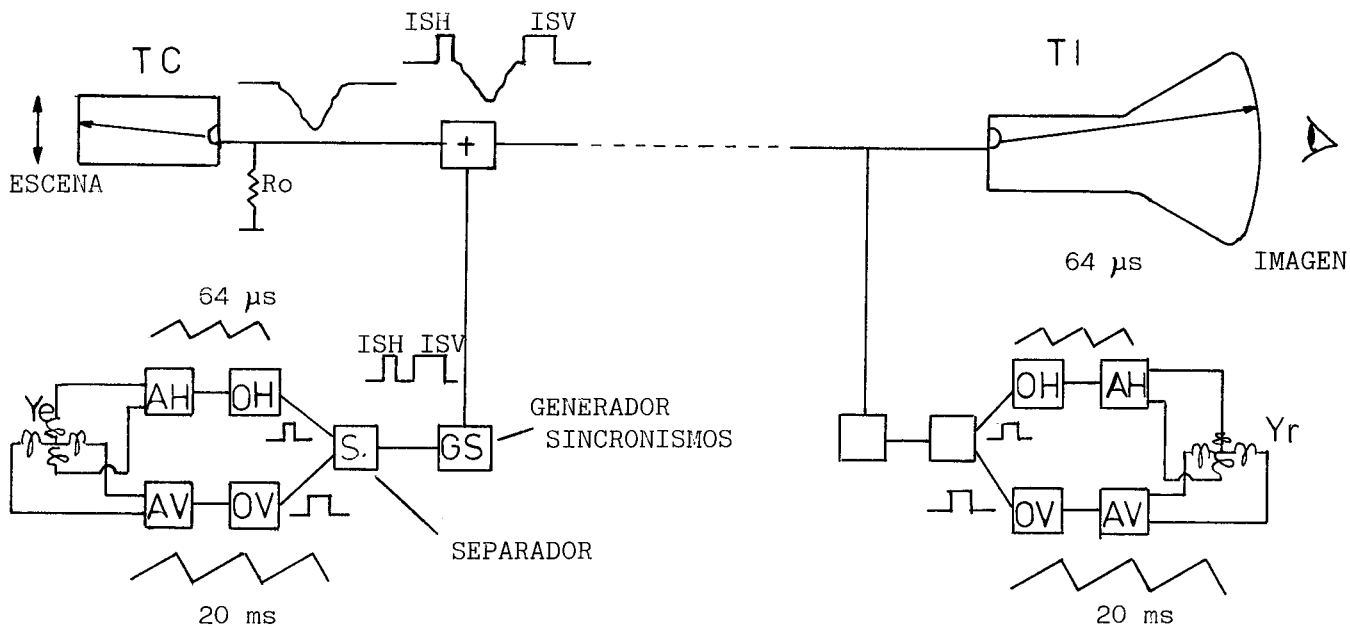
Combinadas ambas señales en un sumador obtenemos la señal de video:

$$V = V(t) + V_a(t).$$



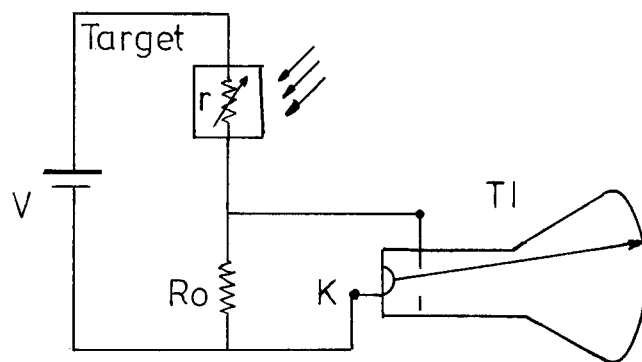
Esta señal es enviada al tubo de imagen situado en el extremo receptor aplicandolo sobre el haz y modulando su intensidad.

El yugo receptor Y_r obliga a trazar un cuadro de líneas en la pantalla al haz, que sincronizado con el extremo emisor reproduce las luminancias de la escena, cuya imagen es contemplada por el observador en el tubo de imagen.



Los periodos de los dientes de sierra del oscilador de relajación son $H=64\mu\text{s}$ y $V=20\text{mms}$ correspondientes a las frecuencias $F_h=15.625$ y $F_v=50$ Hz para la norma de 625 líneas.

El circuito equivalente queda representado en la siguiente figura:



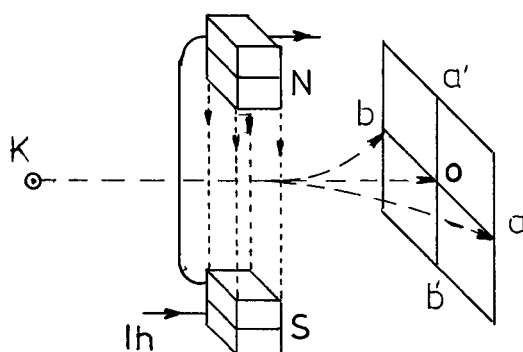
El valor de la tensión transmitida será:

$$V(t) = V \times (R_o / (V + R_o))$$

Para un punto blanco de la escena, r tiende a cero en el target y resulta una tensión máxima de luminancia. El tubo de imagen debe traducir esta tensión como un punto blanco en la pantalla. Puntos de tonalidades grisáceas en la escena corresponderán a valores de r , tales que, en la pantalla reproducirán análogamente dichas tonalidades.

Veamos más determinadamente los yugos. Estos forman un bloque único de dos electroimanes ortogonales que permiten barrer todos los puntos del mosaico o de la pantalla. El electro imán vertical produce la desviación del haz al aplicar las respectivas corrientes I_h e I_v . Como ya hemos visto, la frecuencia horizontal es $F_h=15.625$ Hz y la vertical $F_v=50$ Hz, y por tanto los periodos $H=64$ microsegundos y $V=20$ milisegundos.

Procedente del cátodo K el electrón se dirige a la pantalla P impulsado por un campo eléctrico E y un campo magnético H . Si $B=0$ no existe inducción por lo que el electrón incide en el centro de P .



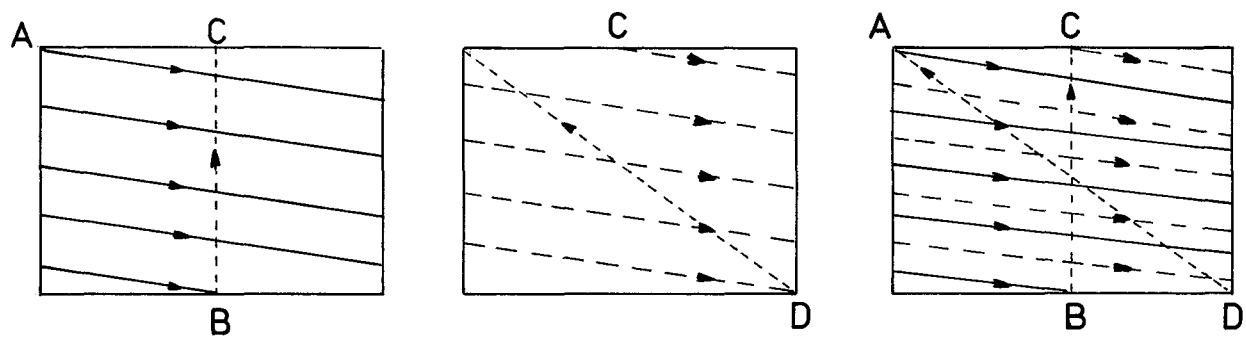
Si $B \neq 0$ el electrón se desvía en el sentido ba , si se trata de I_h el que crea el campo, o en sentido $a'b'$ si es I_v el que lo crea.

La acción simultánea de los dientes horizontal y vertical, obliga al haz a describir líneas ligeramente inclinadas una debajo de otras.

La imagen de la escena tiene la forma de un rectángulo horizontal, es más ancha que alta debido al predominio del movimiento horizontal sobre el vertical en las escenas animadas. Esto exige mayor número de elementos horizontales.

Existe un número infinito de combinaciones de altura y anchura, pero se toma la relación Ancho / Alto = 4/3 llamada relación de aspecto.

Como ya hemos comentado, la secuencia alterna o exploración entrelaza, analiza, el cuadro de la imagen en dos campos de líneas para evitar el efecto de centelleo o parpadeo.



El campo primero C1 comprende el trayecto ABC, con retorno en el punto B y el campo segundo C2 abarca el trayecto CDA con retorno en D. La superposición de ambos campos nos da el cuadro completo de líneas:

$$C = C1 + C2$$

cuyo trayecto es ABCDA.

A) Formación de V(t)

Consideremos la imagen y las variaciones locales de iluminación que la caracterizan. Su análisis se efectúa en el TC por el desplazamiento de un haz de pequeña superficie $dS = a$, que traduce en tensión eléctrica la cantidad de luz integrada de cada cuadro elemental dS de la imagen.

Cuando el haz se desplaza con velocidad de barrido V_b "constante" sobre una línea recta de la imagen se forma la señal eléctrica $V(t)$ o tensión instantánea, que es función de la iluminación E del punto analizado.

B) Señales auxiliares $V_a(t)$.

Veamos las características de este tipo de señales formadas por los impulsos siguientes: IS_h , "Fin línea"; IB_h , borrado o "Supresión línea"; IS_v , "Fin campo" e IB_v , "Supresión campo", creados en Generador de Sincronismos.

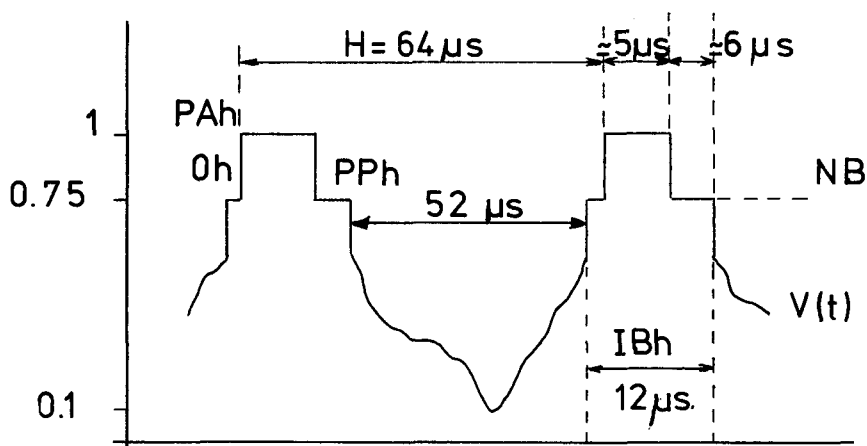
C) Señales de fin y supresión de línea.

C1) Caso ideal: Impulsos rectangulares.

Estas señales son también conocidas como ISh (impulso de sincronismo horizontal) e IBh (impulso de borrado horizontal).

El ISh es una señal rectangular cuyo flanco anterior constituye el "Trigger" o instante Oh de sincronismo del barrido producido por Ih. El IBh es una señal más ancha similar, cuyos flancos son función del nivel de la señal simple V(t) y cuya misión es borrar el retorno de línea. Ambos impulsos están superpuestos en el tiempo y son de distinta amplitud, para poder separar ISh de V(t) en el receptor.

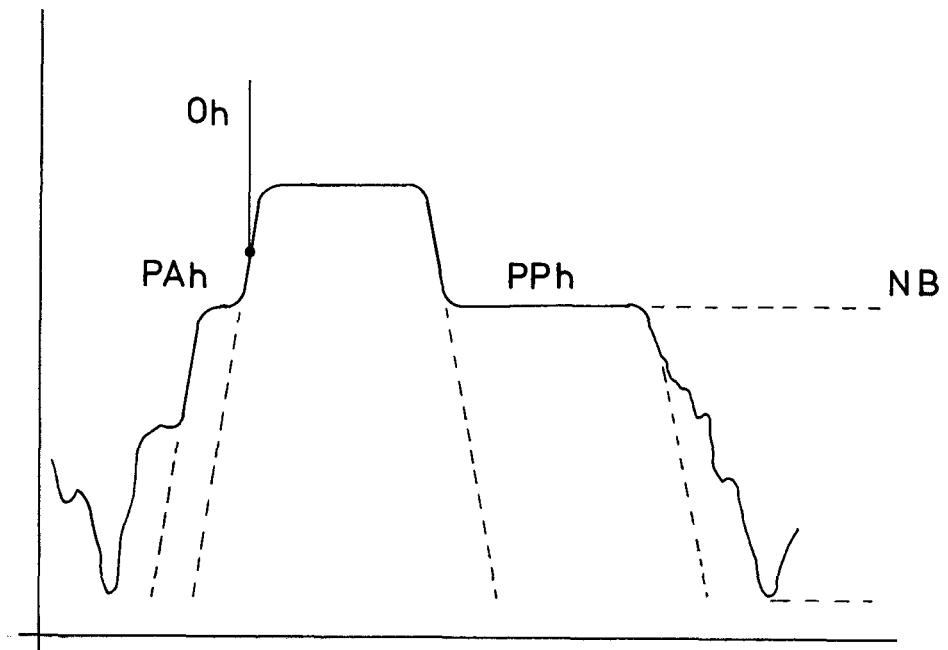
El ISh va precedido por el pértico anterior PAh y seguido por el pértico posterior PPh. En cada cuadro hay 15.625 ISh. Parten del nivel de borrado NB, su amplitud es $S = 0.25$ y su misión de sincronizar el oscilador horizontal del TC y del TI.



C2) Caso real: Impulsos trapezoidales.

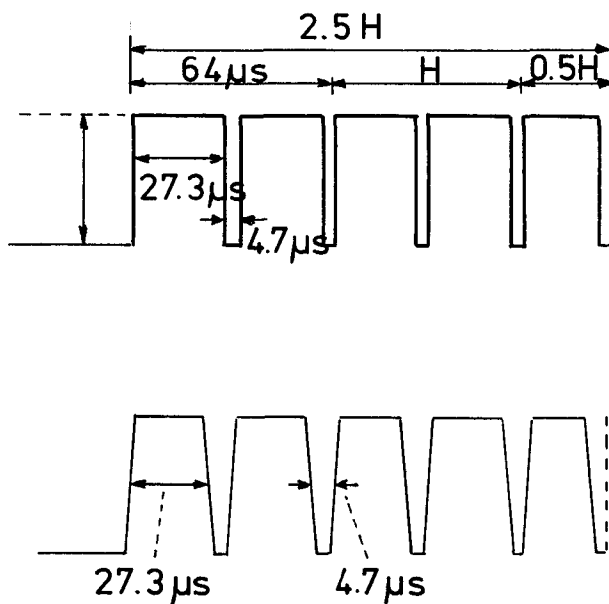
En el impulso ideal los flancos son verticales y el tiempo de "establecimiento" o de subida y bajada es nulo, $T_e = 0$.

El espectro de frecuencias de Fourier, para un impulso rectangular, comprende la frecuencia fundamental e infinitos armónicos, lo que exige circuitos de ancho de banda infinito. Como en TV el ancho de banda de vídeo es de 5 MHz, el impulso que pasa por los circuitos correspondientes es de forma trapezoidal, ya que el ojo acepta la fundamental y armónicos hasta del orden de 10.



C3) Señales de fin de campo.

Reciben el nombre de impulsos de sincronismo vertical ISv, y están formados por la duración de $2.5 H$ (siendo H el tiempo de barrido de una línea, $64 \mu\text{seg.}$) en la que caben cinco impulsos verticales de $27.3 \mu\text{seg}$ separados entre sí por serrados o hendiduras de $4.7 \mu\text{seg}$, sumando en total $0.5 H = 32 \mu\text{seg}$ el periodo de cada ISv. El serrado de estos impulsos tiene la misma duración, $4.7 \mu\text{seg}$ que los ISh. La amplitud de los cinco ISv es $S = 0.25$ a partir del NB y su misión es sincronizar el oscilador vertical OV del TC y del TI.



El grupo podría ser uno sólo de duración $2.5 H$ sin serrados, pero se le divide en cinco para que el haz siga explorando líneas durante el ascenso

vertical del haz. Los ISv son de mayor duración que los ISh con el fin de efectuar su separación respectiva, mediante un circuito RC integrador y diferenciador y ser aplicado cada uno al oscilador OV y OH.

Es decir, los ISv al pasar por un circuito integrador forman la señal efectiva de fin de campo integrada, que es un diente de forma exponencial, cuya amplitud controla al diente de sierra.

El ISv comienza en la parte final de cada campo de 312.5 H, y durante él asciende el haz a la parte superior del cuadro, para iniciar el barrido del campo siguiente.

Para que el periodo de campo sea siempre igual a 20 milisegundos, y debido a que en el campo C1 la duración de la última línea es de 0.5 H, se introducen impulsos de "igualación". De ésta forma siempre tendremos periodos de campo iguales a 20 milisegundos.

*CIRCUITOS DE
MUESTREO Y RETENCION.*

CIRCUITOS DE MUESTREO Y RETENCION.

Los circuitos de muestreo y retención (Sample and Hold) se usan ampliamente en el procesado de señales analógicas y en sistemas de conversión de datos para almacenar de forma precisa, una tensión analógica durante un tiempo que puede estar comprendido entre algo menos de 1 microsegundo y varios minutos.

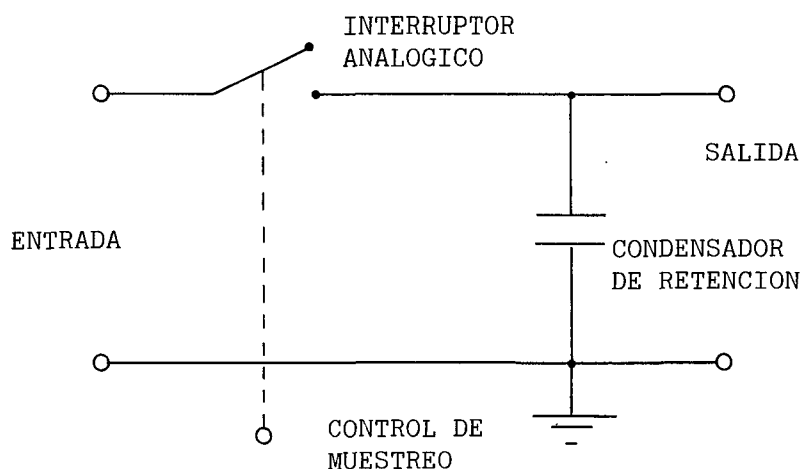
Esta característica les da importancia en aplicaciones que incluyen sistemas de distribución de datos, sistemas de adquisición de datos, sistemas simultáneos de muestreo y retención, convertidores A/D, osciloscopios de muestreo, multímetros digitales, filtros reconstructores de señal, etc..

Un circuito S&H es un elemento de memoria analógica que almacena una tensión dada en un condensador de alta calidad. El circuito puede tomar una muestra de tensión y congelarla durante un período de tiempo especificado, mientras otros sistemas o circuitos se sirven de ella.

La figura muestra un circuito S&H básico en concepto, y en el cual, un interruptor electrónico está conectado a un condensador de retención (Hold) de forma que cuando el interruptor se cierra, el condensador se carga a la tensión de entrada. Cuando el interruptor se abre, el condensador almacena esta carga, por lo que retiene la tensión durante un período de tiempo especificado por un

circuito exterior. Dado que la amplitud de la señal analógica aplicada contiene información, el circuito de muestreo debe poseer una buena linealidad durante los períodos de toma de muestras, sin ningún pedestal o tensión de offset.

En cada terminal de un S&H hay una situación distinta: en la entrada nos encontramos con una señal analógica, en la salida una discreta y en el terminal de control de muestreo, una digital. El terminal de entrada del S&H debe ser en alta impedancia para que la fuente de tensión a muestrear no sea cargada. La salida debe tener baja impedancia para que pueda gobernar una carga relativamente alta. En todos los diseños de S&H se pone un amplificador con una muy alta impedancia de entrada (pase de entrada con transistor de efecto de campo FET) y una muy baja corriente de polarización, para que la carga del condensador no se pierda a través de él.



Por tanto, los circuitos de muestreo y retención son básicamente circuitos de almacenamiento de energía. Hay ciertos tipos de condensadores próximos al condensador ideal, con unas pérdidas muy bajas, y por lo tanto una muy alta resistencia paralelo. Esta resistencia especificada en megaohmios-microfaradio (resistencia de aislamiento) lo es para un condensador de un microfaradio, y es numericamente igual a la constante de tiempo de autodescarga del condensador en segundos. Se puede obtener la resistencia paralelo para otros valores de capacidad. Para ello, dividimos la resistencia de aislamiento por la capacidad en microfaradios. Debido a que para pequeños valores de capacidad, este valor podría llegar a ser muy alto, algunos fabricantes especifican un valor mínimo garantizado, generalmente doble de la resistencia de aislamiento. Los condensadores de alta calidad utilizados en S&H tienen resistencias de aislamiento altísimas, con una consiguiente autodescarga muy baja, del orden de días.

Los condensadores poseen, además, otros parámetros. Unos, como el coeficiente de temperatura de la capacidad, son poco importantes, mientras que otros lo son más. Entre estos últimos cabe destacar el coeficiente de absorción dieléctrica ya que afecta a la precisión de la tensión almacenada. Si un condensador se carga a una determinada tensión, se descarga cortocircuitándolo y después se deja a circuito abierto, su tensión tiende a aumentar desde cero a

la tensión original. El condensador tiene una característica de memoria de tensión conocida como absorción dieléctrica, y es debida a que el material dieléctrico no se polariza instantaneamente. Por lo tanto, no se puede recuperar toda la energía almacenada en un condensador de forma instantánea en la descarga. Esto ocasiona un error, ya que si el condensador a permanecido a una tensión V_0 y llega una tensión de muestreo distinta, el condensador empezará a deslizarse a la tensión V_0

Hay factores que reducen considerablemente este error. Por un lado la remanencia aumenta con el tiempo, mientras que en un S&H una tensión se almacena por un tiempo relativamente corto. Por otro lado, los efectos de la absorción dieléctrica aumentan para un tiempo largo en circuito abierto comparado con el tiempo de cortocircuito, y en un S&H el tiempo de retención se pretende que no sea mucho más largo que el tiempo de muestreo entre otras cosas, para que se cumpla el teorema del muestreo y pueda reconstruirse la señal de entrada.

En el montaje práctico, como circuito 'sample and hold' utilizamos uno de los cuatro interruptores analógicos de los que dispone el circuito integrado 4066; el amplificador operacional con entrada JFET LF-351, en configuración seguidor de tensión de ganancia unidad. Este operacional, tiene un gran ancho de banda (4 MHz) y muy elevada impedancia de entrada (10 M Ω). Debido a que la

frecuencia de muestreo no es muy grande 15625 Hz (1/64 μ s), y que el tiempo de toma de muestra es relativamente elevado $(R2.C9) \approx 1\mu$ s, el valor del condensador de retención C8 no es crítico, empleando para este fin un condensador cerámico de 100 pF con resistencia de aislamiento mayor de 1000 M Ω .

*CONVERTIDORES
ANALOGICO DIGITALES.*

CONVERTIDORES ANALOGICO DIGITALES.

El vertiginoso desarrollo experimentado en los últimos años por los sistemas digitales debido fundamentalmente a la facilidad, sencillez y exactitud de la técnica digital sumado al espectacular avance tecnológico y al continuo abaratamiento de componentes, subsistemas y sistemas digitales, hace que la mayoría de los sistemas electrónico tengan un tratamiento digital, por lo que resulta evidente la necesidad de disponer de dispositivos conversores A/D y D/A, ya que las señales del mundo físico son generalmente de naturaleza analógica. Por consiguiente, el esquema de bloques de un sistema electrónico con tratamiento digital será el siguiente:



El objeto de los subsistemas de conversión A/D ó D/A es producir una relación biunívoca entre una magnitud eléctrica de caracter analógico (tensión o corriente) y una función digital codificada o viceversa. En otras palabras, un convertidor A/D es un dispositivo electrónico, que transforma una magnitud analógica en una magnitud digital o palabra binaria, que podría decirse, es proporcional a dicha magnitud analógica.

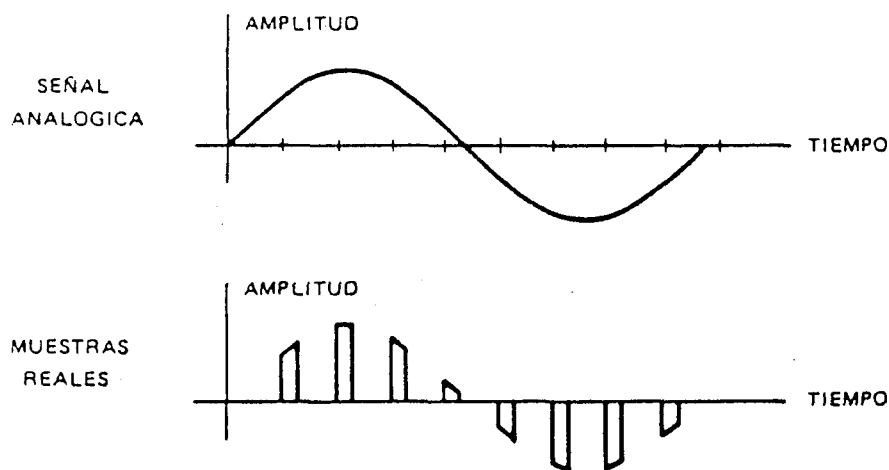
El proceso de conversión A/D (que es del único que hablaremos) comprende tres fases; muestreo, cuantificación y codificación.

Muestreo.

El proceso de muestreo consiste en tomar valores instantáneos de una señal analógica, a intervalos de tiempos iguales y con una periodicidad que viene determinada por el teorema del muestreo. A los valores instantáneos obtenidos se les denominan MUESTRAS.

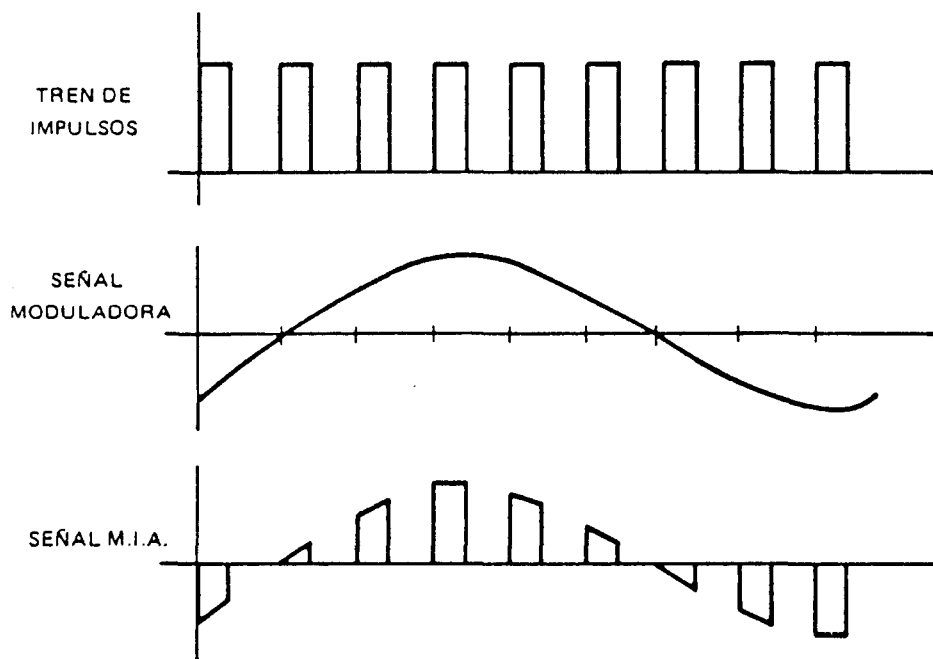
Podemos considerar dos tipos de muestreo; el ideal y el real.

En el muestreo ideal las muestras obtenidas son instantáneas, de anchura nula.



El muestreo ideal no es físicamente realizable. En la práctica, una muestra es una medida del valor instantáneo de una señal, pero tomada durante un tiempo que es muy corto comparado con el tiempo entre dos muestras consecutivas. A este tipo de muestreo se le denomina muestreo real, y la señal obtenida es un tren de impulsos, cada uno de los cuales tiene una amplitud igual al valor que tenía la señal en el instante del muestreo.

El muestreo se puede considerar como un proceso de modulación en amplitud X de un tren de impulsos. Por esta razón a la señal muestreada se la denomina señal MIA (Modulación de Impulsos en Amplitud) o PAM, según la notación anglosajona (Pulse Amplitude Modulation).



El muestreo se efectua siempre a un ritmo uniforme, que viene dado por la frecuencia de muestreo F_m . La condición que debe cumplir F_m viene dada por el teorema del muestreo, que para el caso de una señal analógica que contenga distintas frecuencias se enuncia de la siguiente forma:

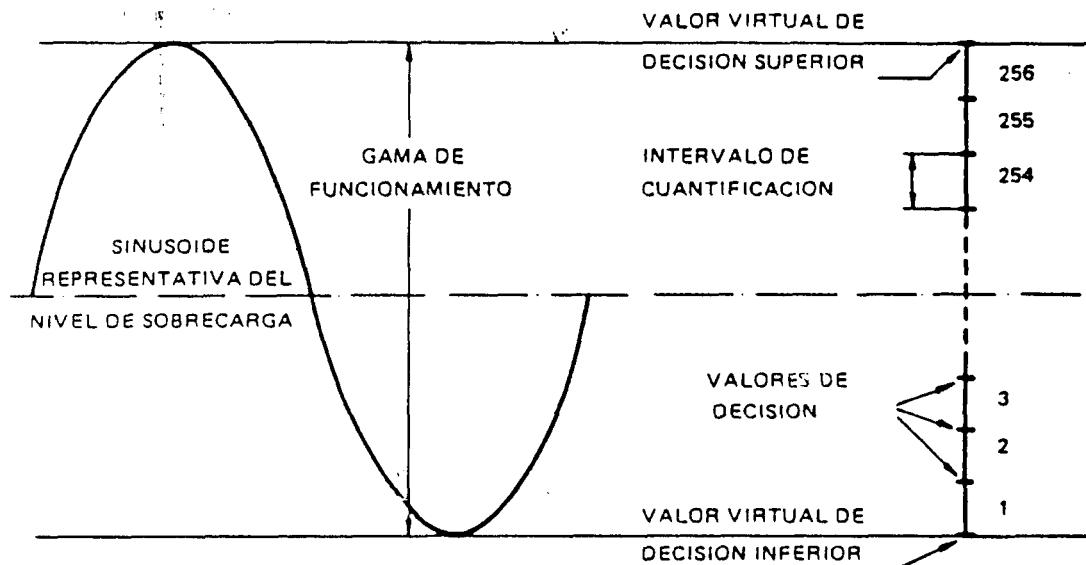
"Una señal cuyo espectro de frecuencias queda limitado a la frecuencia F , o de banda limitada a F , queda completamente determinada por muestras tomadas a una velocidad igual o superior a $2.F$ "

Cuantificación.

Según se vió, las muestras obtenidas en un muestreo real tienen una duración o anchura finita, pero su amplitud puede tomar infinitos valores comprendidos entre el valor 0 y el valor más alto de la señal a muestrear.

Para representar en forma aproximada la amplitud de las muestras se va a utilizar un número finito de valores discretos. Para ello, toda la gama de amplitudes que pueden tomar las muestras, o GAMA DE FUNCIONAMIENTO, se divide en intervalos iguales y a todas las muestras cuya amplitud cae dentro de un mismo intervalo, se les asigna el mismo valor.

A este proceso se le denomina cuantificación, y a cada intervalo en que se ha dividido la gama de funcionamiento se le llama INTERVALO DE CUANTIFICACION. Dentro de una determinada gama de funcionamiento, cada intervalo de cuantificación está limitado por dos valores de decisión. Estos valores situados en los extremos de la gama de funcionamiento se denominan VALORES VIRTUALES DE DECISION y limitan la máxima amplitud de la señal que se puede transmitir sin recorte de crestas. Al nivel que posee esta señal se le llama NIVEL DE SOBRECARGA.



El proceso de cuantificación introduce necesariamente un error, ya que se sustituye la amplitud real de la muestra por un valor aproximado. A este error se le denomina ERROR DE CUANTIFICACION. El error de cuantificación se puede reducir aumentando el número de intervalos de cuantificación, pero existen limitaciones de tipo práctico que obligan a que el número de intervalos no sobrepase un determinado valor.

Una cuantificación de este tipo, en la que todos los intervalos tienen la misma amplitud, se llama cuantificación uniforme.

El efecto de la distorsión o error de cuantificación es similar al de un ruido aleatorio superpuesto a la señal analógica original. Por lo tanto, la relación entre el nivel de señal N_s y el nivel de distorsión de cuantificación N_c ha de ser tal, que dicha distorsión no interfiera de forma apreciable en la información que contiene la señal analógica.

En la cuantificación uniforme la distorsión de cuantificación por muestra es \leq a la mitad del ancho del intervalo de cuantificación, independientemente de la amplitud que posea la muestra, dado que si la amplitud de una muestra cae dentro de un intervalo determinado de cuantificación se le asigna el valor mitad de dicho intervalo. Por esta razón, la relación señal/ruido va empeorando al disminuir el nivel de la señal de entrada. La situación se hace ya inadmisibile para señales con amplitud análoga a la de un intervalo de cuantificación.

Si con este tipo de cuantificación uniforme queremos mantener una relación señal /ruido aceptable para las señales de nivel bajo, es necesario dividir la gama de funcionamiento en un número mayor de intervalos de cuantificación, lo cual daría lugar a una relación señal/ruido innecesariamente buena para las señales de nivel alto. Además de esto, un mayor número de intervalos daría lugar a un mayor número de bits,

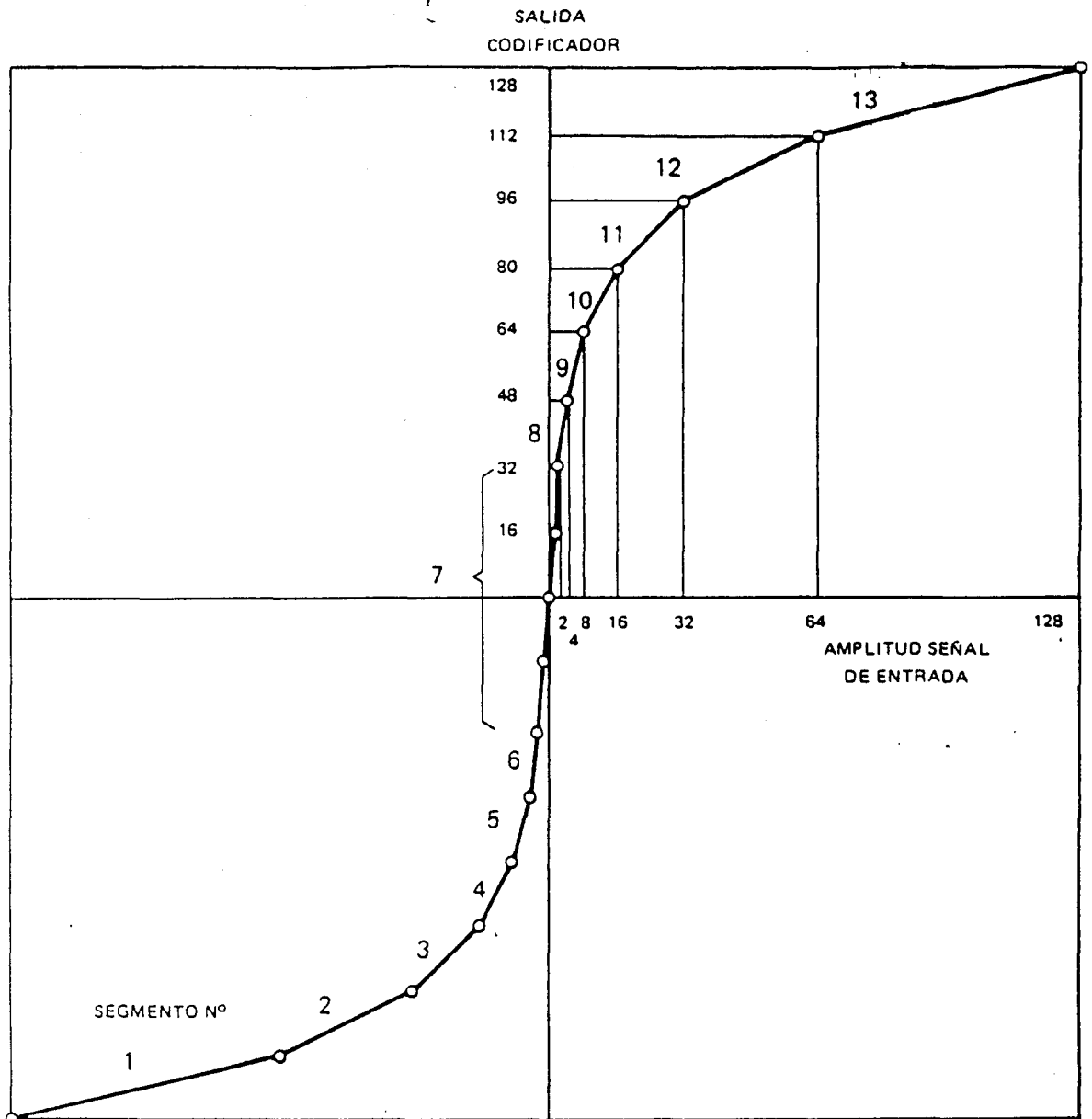
necesarios para codificar las muestras.

Por lo tanto hemos de buscar un procedimiento en el cual la relación señal/ruido sea aceptable y se mantenga prácticamente constante, para toda la gama de niveles de señales que se van a utilizar con el dispositivo electrónico.

Este problema se resuelve utilizando una cuantificación NO UNIFORME, en la cual se toma un número determinado de intervalos y se distribuyen de forma no uniforme, aproximandolos en los niveles bajos de señal separandolos en los niveles altos. De esta forma, para las señales de nivel débil es como si se empleara un número muy elevado de niveles de cuantificación, obteniendo una disminución en la relación señal/ruido. Sin embargo para las señales fuertes se tendrá una situación menos favorable que la correspondiente a una cuantificación uniforme, pero todavía suficientemente buena.

Esta técnica de cuantificación se utiliza frecuentemente en telefonía, para señales vocales, recomendando el C.C.I.T.T. (Comité Consultivo Internacional Telegráfico y Telefónico) la utilización de 256 intervalos de cuantificación.

La función de transferencia de la cuantificación no uniforme se muestra en la siguiente figura:



Codificación

La codificación es el proceso mediante el cual se representa una muestra cuantificada mediante una sucesión de "1" y "0", es decir, mediante un número binario.

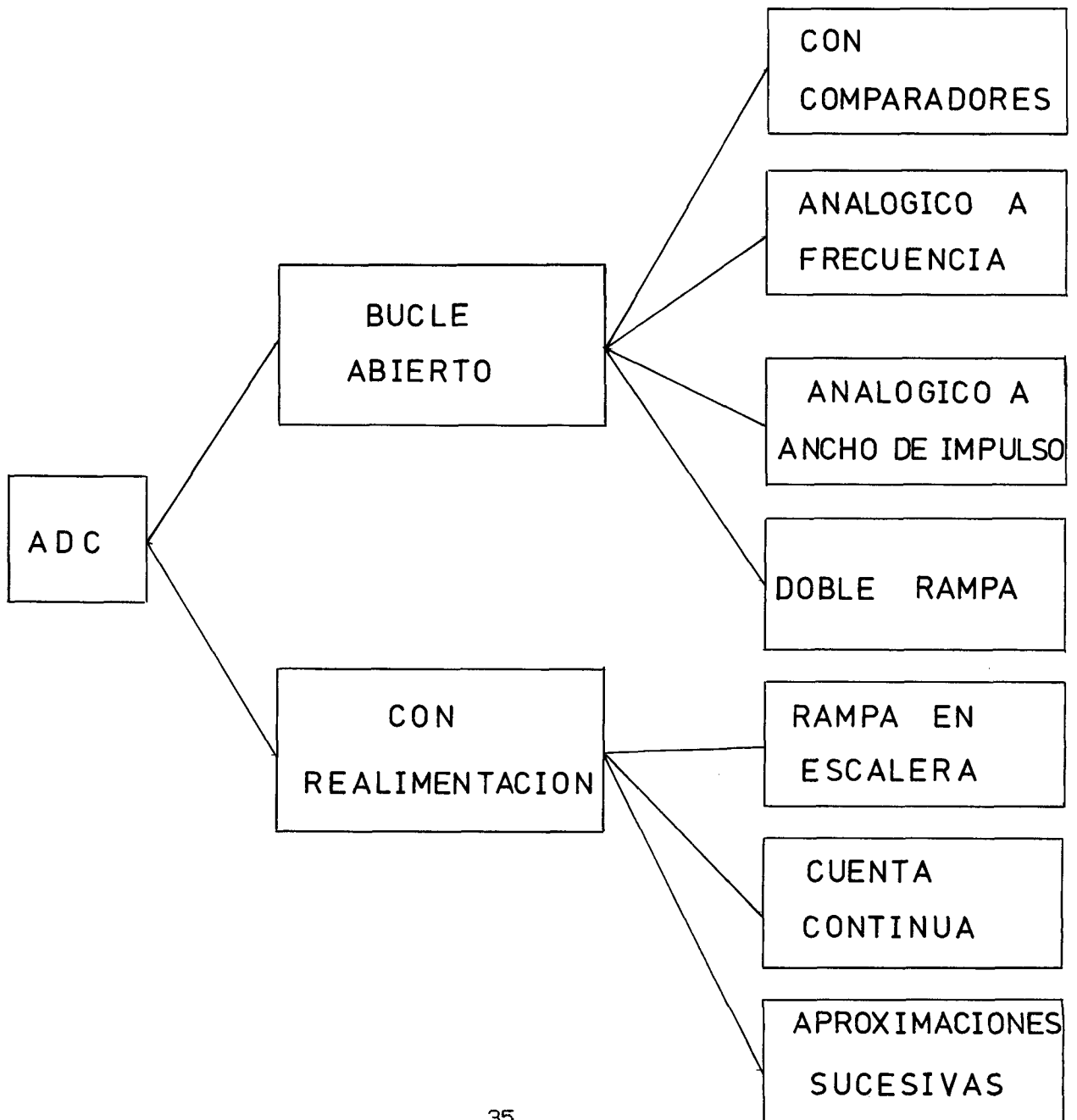
Para la codificación de la información se utilizan los códigos binarios que son aquellos que emplean dígitos binarios como medio de representar la información. El número de bits (n) que utilice cada código nos determinará la cantidad de símbolos que podamos codificar (2^n), de forma que si se desea codificar N símbolos, o intervalos de cuantificación en este caso, habrá que encontrar un número n de bits tal que se verifique:

$$2 < N \leq 2^n$$

Clasificación General de los Subsistemas

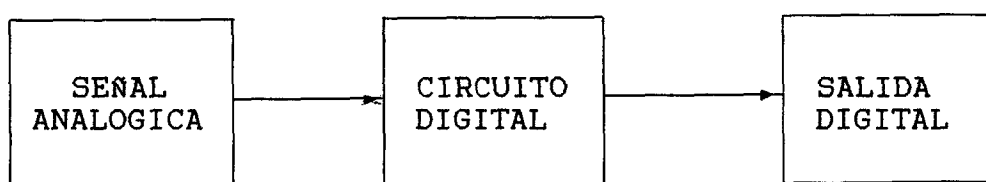
Convertidores A/D.

Existen muy diversas técnicas para realizar esta transformación analógico digital. Estas pueden ser agrupadas en dos tipos básicos: A/D de bucle abierto y A/D de bucle cerrado.



A) A/D de Bucle Abierto.

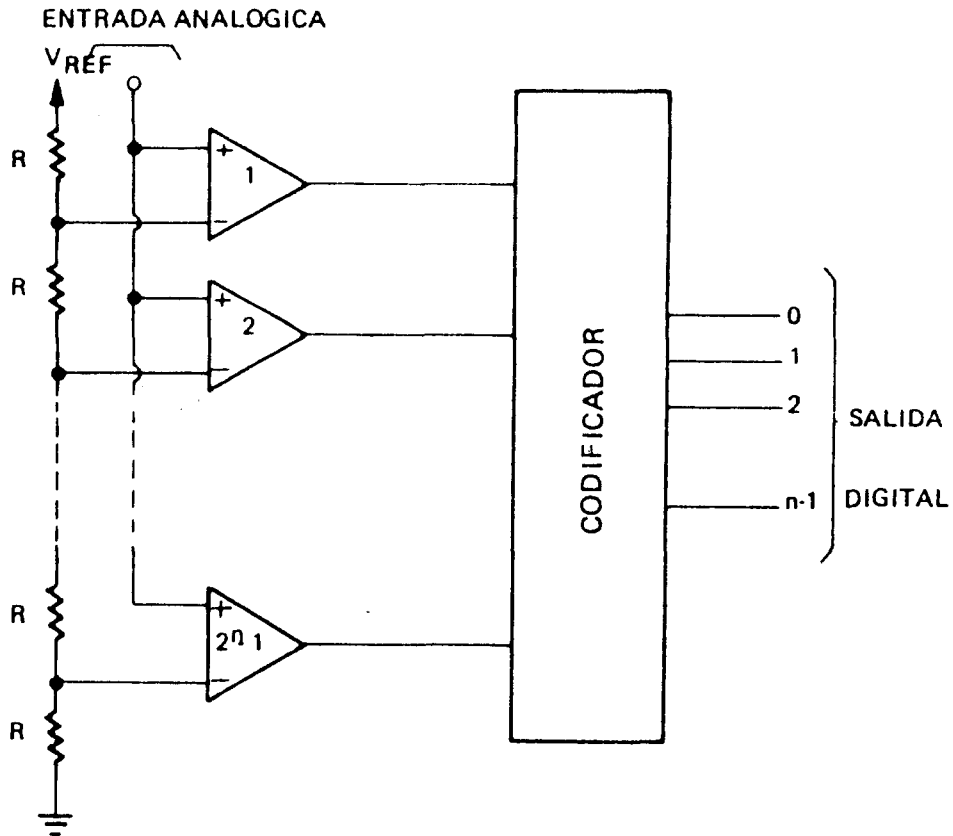
En los convertidores A/D de bucle abierto no existe realimentación interna, obteniéndose la información digital de forma directa al aplicar una señal analógica a su entrada:



Entre estos convertidores se encuentran los que convierten la señal analógica en frecuencia o en un impulso y los que emplean comparadores.

A1) Conversores A/D con Comparadores.

En este caso, los procesos de cuantificación y codificación están claramente separados. El primer paso se lleva a cabo mediante comparadores que discriminan entre un número finito de niveles de tensión. Estos reciben en sus entradas una señal analógica de entrada junto con una tensión de referencia, distinta para cada uno de ellos. Al entrar las tensiones de referencia escalonadas, es posible conocer si la señal de entrada está por

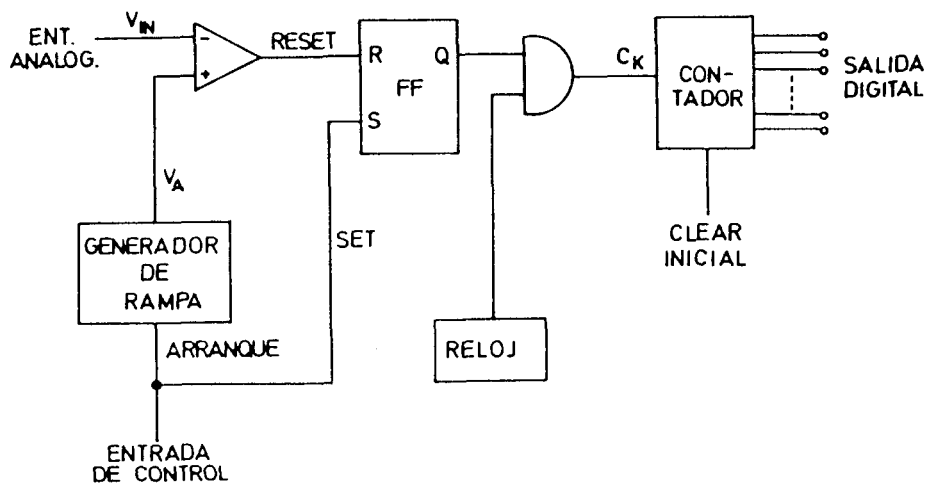


encima o por debajo de cada una de ellas, lo cual permitirá conocer el estado que le corresponde como resultado de la cuantificación.

A continuación será necesario un codificador que nos entregue la salida digital. Este convertidor es de alta velocidad, ya que el proceso de conversión es directo en lugar de secuencial, reduciéndose el tiempo de conversión necesario a la suma de los de propagación en el comparador y codificador.

Por contrapartida, si se quiere una resolución alta, su fabricación resulta cara.

A2-Convertidor A/D Analógico a Ancho de Impulso.



También llamado ADC con integrador de simple pendiente. Transforma la tensión desconocida en un periodo de tiempo que es medido mediante un reloj y un contador. Al recibir por la entrada de control la orden de iniciar la conversión el conversor comienza la generación de la rampa y pone a "1" la salida Q del biestable. Este nivel se mantiene hasta que la rampa supere el de la entrada analógica, la salida del flip-flop vuelve a 0 y el contador deja de contar los impulsos de frecuencia fija del reloj.

Al ser la duración del impulso en la salida del biestable función de la tensión de entrada, las

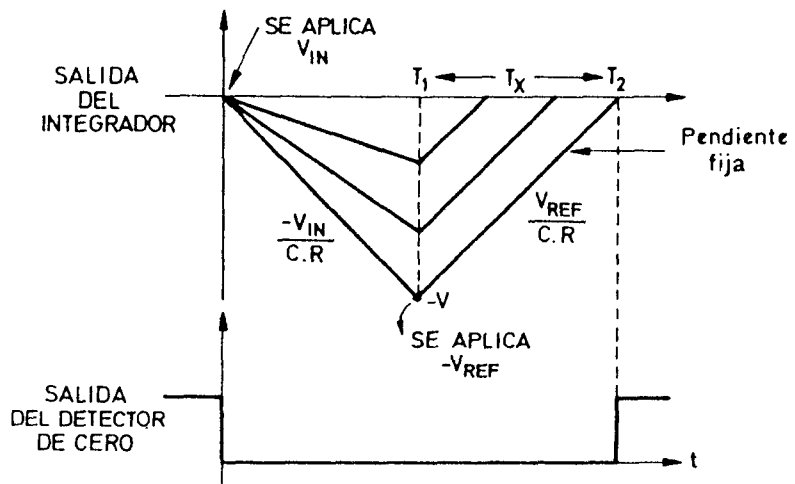
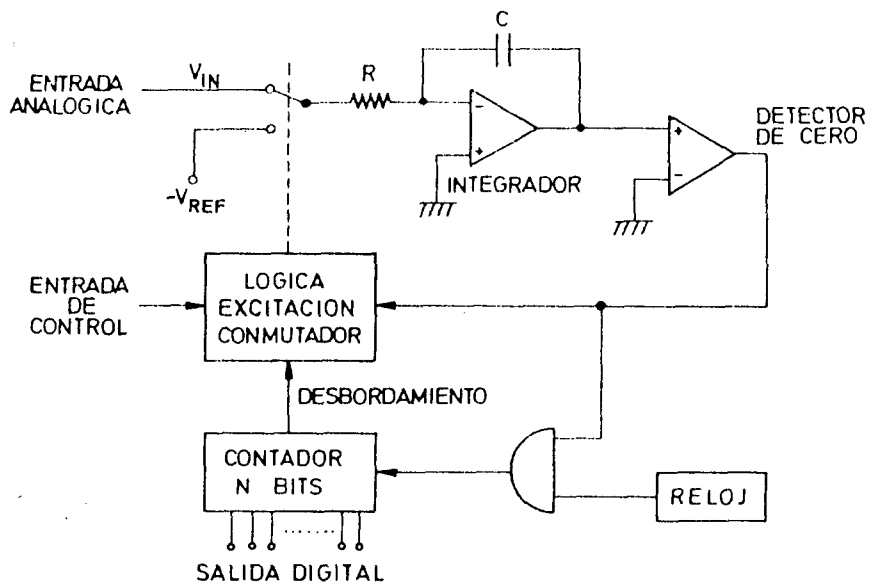
salidas del contador serán una representación digital de la misma.

Entre los inconvenientes que presenta este convertidor esta el offset de entrada del comparador de tensión, así como la posible variación de la frecuencia del reloj, por ejemplo a causa de la temperatura. Otro inconveniente es que el tiempo de conversión no es fijo, ya que es función del nivel de la tensión analógica de entrada.

A3) Convertidor A/D de Doble Rampa.

El principio de funcionamiento se basa en un integrador.

El proceso se inicia conectando la tensión de entrada al integrador durante un tiempo fijo T_1 , que emplea el contador en pasar del estado 000...0 hasta producirse un desbordamiento en la cuenta, es decir se llega a 2 impulsos en el reloj. Durante este intervalo de tiempo la salida del integrador se va haciendo negativa hasta alcanzar un valor mínimo en T_1 . La información de desbordamiento provoca la aplicación de una tensión de referencia ($-V_{ref}$) a la entrada del integrador, lo cual hace que la salida tienda a 0 voltios.



Durante el segundo tramo, en el que se mantiene la pendiente de la rampa constante, son de nuevo contados los impulsos del reloj, y la cuenta que se alcance en el momento en el que cruce por cero de la salida del integrador es la palabra digital equivalente buscada.

La ventaja de este convertidor es su precisión que es función de la linealidad de las rampas que entregue el integrador y de la precisión de la referencia. Como contrapartida, el tiempo de conversión es largo y variable.

B) A/D Realimentados o de Bucle Cerrado.

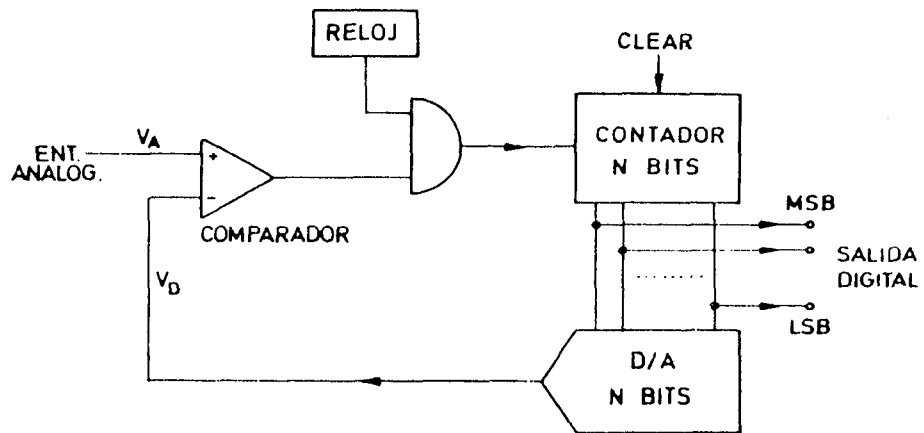
Los convertidores A/D de bucle cerrado son los que poseen un lazo de realimentación. En ellos, los procesos de cuantificación y codificación son simultáneos, obteniéndose una secuencia de números digitales que son convertidos a un valor analógico, el cual es comparado con la entrada.

B1) A/D con Contadores.

Dos son los conversores que pueden englobarse dentro de este tipo: Convertidores A/D con rampa en escalera y convertidores A/D de cuenta continua.

-ADC con Rampa en Escalera.

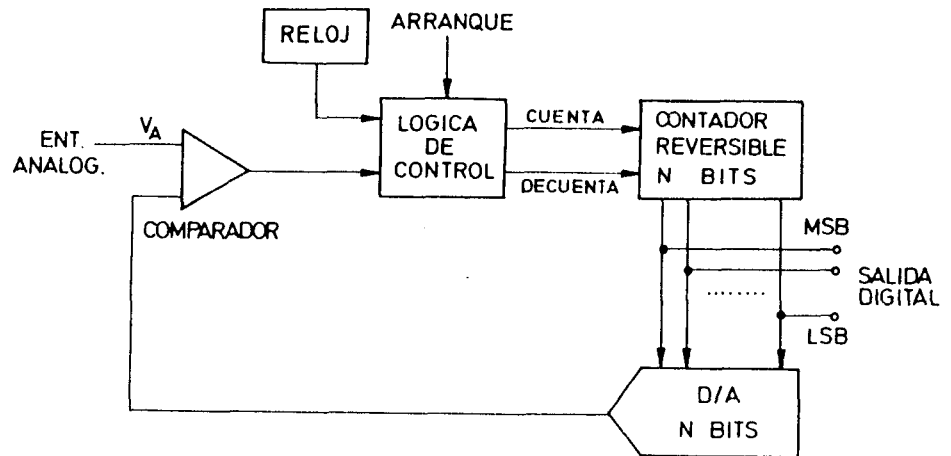
Consta de reloj, contador, conversor D/A y de un comparador como veremos en la figura. Cuando se desea comenzar la conversión, se inicializa el contador mediante la entrada "CLEAR" de puesta a cero. A partir de aquí el conversor D/A entrega una señal analógica, V_d , equivalente a la salida digital del contador la cual será comparada con la señal analógica de entrada, V_a .



En el momento en el que la tensión V_d supere ligeramente a V_a se inhibe el paso de impulsos al contador, teniéndose en la salida del mismo el valor digital buscado. Este proceso aunque simple presenta el grave inconveniente de tener que parar el contador por todos los estados desde el inicio.

Esto implica que los tiempos de conversión serán muy distintos dependiendo del valor analógico. No obstante, este método podría ser bueno para sistemas de alta resolución, aunque a su vez, esta mayor resolución se traduce en mayor tiempo de conversión, ya que el periodo de reloj debe tener un valor mínimo impuesto por la suma de los tiempos de propagación del contador y de respuesta del DAC (Convertidor D/A) y del comparador.

-ADC de Cuenta Continua.



Los problemas que presenta el ADC con rampa en escalera pueden reducirse mediante el ADC de cuenta continua, cuyo principio de funcionamiento es análogo al de rampa en escalera. La diferencia estriba en que en el de rampa continua la cuenta no va a partir siempre de cero, sino que trata de seguir continuamente a la señal analógica desde el último valor que tuviera.

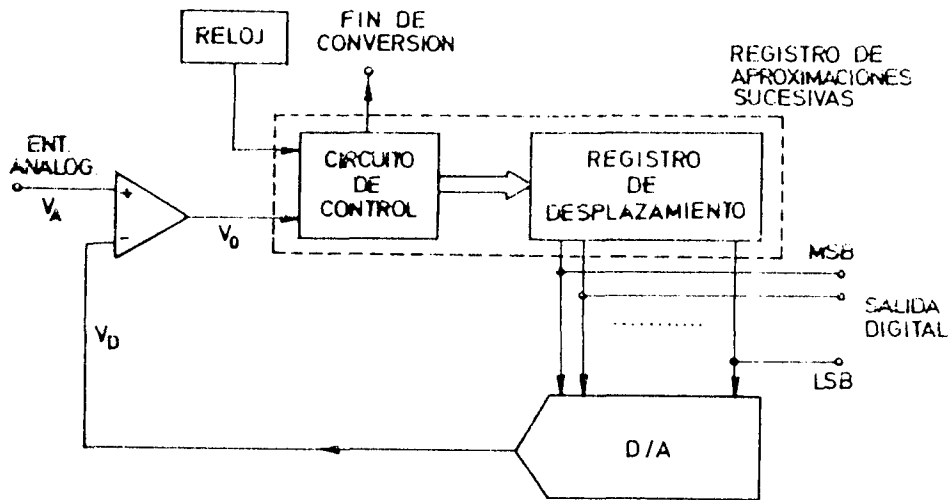
Este funcionamiento reducirá el tiempo necesario de conversión, puesto que la siguiente muestra, con gran probabilidad, estará próxima a la última. Para conseguir esto sólo será necesario modificar el circuito en el sentido de emplear un

contador reversible, lo que se muestra en la figura, que permita aumentar o disminuir el valor del estado de la cuenta. El circuito funciona de forma análoga al caso anterior, aplicandose impulsos de cuenta o decuenta al contador dependiendo de la salida que entregue el comparador.

B2) Convertidores A/D de Aproximaciones Sucesivas.

Este es el conversor más popular para la mayoría de las aplicaciones donde se requieren desde medias hasta altas velocidades de conversión. Con una estructura en bucle cerrado, el ADC trata de acercarse al valor final mediante aproximaciones sucesivas.

Para este fin se dispone de un registro de desplazamiento, que almacena la combinación digital, y un circuito secuencial de control, que suele recibir el nombre de registro de APROXIMACIONES SUCESIVAS.

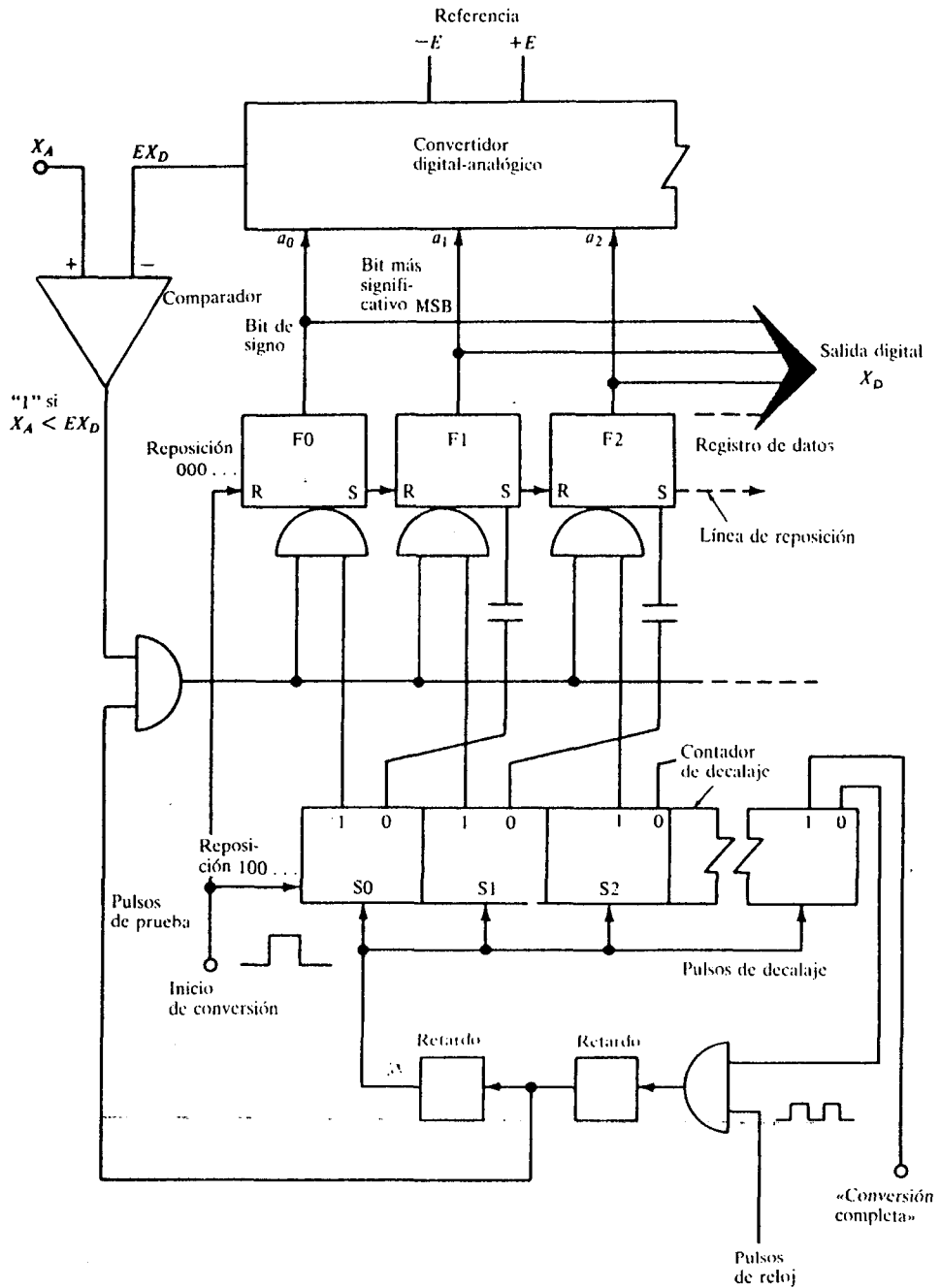


Para entender el funcionamiento de este tipo de conversores veamos el diagrama de bloques representado. El registro de aproximaciones sucesivas se inicializa con la combinación 100.....0, lo cual supone explorar si el valor analógico de entrada está por encima o por debajo del valor medio del rango.

Si por ejemplo V_A es mayor que dicho valor, se investigará dentro de la mitad superior añadiendo un 1 en la siguiente posición (1100.....0). Si aquí se obtiene que V_A es menor que el equivalente analógico de dicho valor digital, se quita el 1 de la segunda posición y se pasa a la siguiente (1010.....0). Este proceso finalizará cuando se halla estudiado el comportamiento del bit menos significativo, lo cual

es indicado a través de la línea de fin de conversión.

Veamos ahora un esquema de este conversor A/D un poco más detallado:



donde:

XA = entrada analógica

XD = salida digital

E = Tensión de referencia del convertidor D/A (DAC)

EXD= Tensión que se obtiene a la salida del convertidor D/A y que se compara directamente con la entrada XA.

Podemos describir el funcionamiento de este circuito de la siguiente forma:

Programamos el registro de desplazamiento del convertidor DAC con lógica digital para "ponderar" XA mediante sucesivas aproximaciones de dígitos binarios. Comenzamos con XD=0 para encontrar el signo de XA; Si $XA > 0$, a continuación ensayamos $EXD = E/2$ para ver si el dígito binario más significativo es 0 o 1, y continuamos hasta el dígito menos significativo. El tiempo de conversión requerido será "n" periodos de reloj, uno para cada bit en XD.

La salida digital XD, y así la tensión de salida del DAC, EXD, está colocada por el registro ADC (salida), que comprende "n" flip-flops (Fo, F1, F2,.....). Los valores sucesivos de XD serán programados por un 1 digital que pasa a través de un contador de decalaje de (n+1) bits (flip-flops So, S1, S2,.....).

El pulso de inicio de conversión repone el contador de decalaje a 1000.... y el registro del ADC

a 0000.... ; esto también libra los pulsos de verificación y los pulsos de decalaje producidos por el reloj del ADC. Inicialmente, $XD = 0000....$, si ahora $XA < EXD = 0$, la salida del comparador será 1 y el pulso de verificación (adecuadamente demorado para permitir a los flip-flops, DAC y comparador que se establezcan) establecerá el flip-flop del bit del signo F_0 a 1. El primer pulso de decalaje sigue y decala el contador a 0100.... Tenemos entonces:

$$XD = \begin{cases} 1100.... & \text{si } XA < 0 \\ 0100.... & \text{si } XA \geq 0 \end{cases}$$

Y la correspondiente salida del DAC ($-E/2$ o $E/2$) es comparada con la entrada analógica XA . Si (y sólo si) $XA < -E/2$ o $0 \leq XA < E/2$, el segundo pulso de verificación repondrá correctamente el flip-flop MSB F_1 a 0 . De lo contrario, el bit más significativo permanece, como así debe ser, a 1.

Los siguientes y sucesivos pulsos de decalaje y de verificación determinan de forma similar los otros bits de XD en orden de significancia decreciente. El pulso de decalaje n -ésimo decala el 1 en el flip-flop del contador de decalaje $(n+1)$ -ésimo, lo cual sirve como una indicación de hecho (DONE Flag) señalando la finalización de la conversión.

Especificaciones de los Convertidores A/D.

A-Rango de Entrada y Código de Salida.

Los ADC comerciales pueden admitir entradas analógicas unipolares o bipolares con rangos típicos de ± 10 voltios, ± 5 voltios o -10 voltios,, con códigos de salida: binario natural, binario desplazado, BCD, complemento a 1 o complemento a 2. Muchas veces los circuitos permiten programar el rango de entrada y el código de salida entre varias opciones.

B-Error de Cuantificación.

Este tipo de error ya fué mencionado al introducir los convertidores A/D, y aparece como consecuencia de que en un ADC de N bits la continuidad de la señal analógica es dividida en 2 rangos. De esta forma, todos los valores analógicos (que llamabamos muestras) dentro de un rango están representados por un único código digital, normalmente asignado al valor medio del mismo.

Existe pues, siempre un error de cuantificación de $\pm 1/2$ LSB, que junto con el ruido del sistema digital, "offset", etc.... se engloba dentro de las especificaciones de precisión.

C- Impedancia de Entrada.

Está comprendida entre pocos $K\Omega$ y decenas de $K\Omega$.

D- Resolución.

La resolución de un convertidor A/D se refiere al n° de bits decididos de modo que el incremento de la tensión de entrada siempre incrementa la salida digital

E- Linealidad.

Es la máxima desviación de la salida real respecto a la teórica, que sería una línea recta. Dependiendo del fabricante esta recta será o "la mejor" obtenida empíricamente, o la trazada entre los puntos extremos de la función de transferencia.

F- Precisión.

Es la desviación de la tensión de salida respecto al valor ideal esperado. Esta especificación incluye errores de ganancia, "offset", no linealidad de la red resistiva, ruido,....

G-Sensibilidad con la Temperatura.

Las características de linealidad, precisión, resolución, impedancia de entrada, no son parámetros fijos, sino que, por el contrario, cambian al hacerlo la temperatura ambiente.

La variación de éstos parámetros con respecto a la temperatura puede darse en $\mu V/^\circ C$ o en partes por millón (ppm) según del que se hable.

H-Tiempo de Conversión.

Es el tiempo requerido por el conversor para entregar la palabra digital equivalente a la entrada analógica. Como ya se vió, este tiempo puede variar mucho de unos convertidores a otros, siendo un valor típico el de 50 μs para velocidades medias, pudiendo llegar a varios nsg, los de baja velocidad, o reducirse a 50 nsg o menos en los ultrarápidos.

I-Relación de Conversión.

Es el recíproco del tiempo de conversión. Los convertidores ADC "instantáneos" nos permiten relaciones de conversión de al menos diez elevado a la ocho palabras por segundo, si se está satisfecho con 4 o 6

bits de resolución. Los convertidores en cascada vienen a continuación en velocidad con una resolución de 8 o 9 bits.

En un convertidor ADC de comparador y rampa y de integración la relación está limitada por los tiempos de respuesta del comparador y de los conmutadores analógicos. Convertidores de integración de 11 bits pueden convertir hasta 5000 palabras por segundo.

Las mejores exactitudes así como las relaciones de conversión más altas, son obtenidas por convertidores de aproximaciones sucesivas, predominantes en computación híbrida.

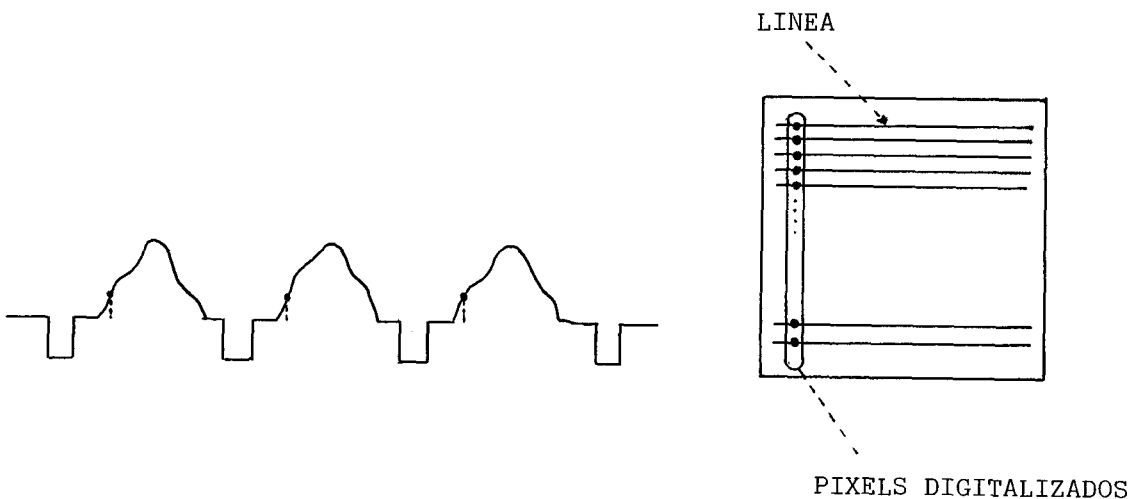
CIRCUITO ELECTRICO.

CIRCUITO ELECTRICO.

En primer lugar veremos, a grandes rasgos, el principio de funcionamiento del digitalizador, para ir más adelante entrando en detalle.

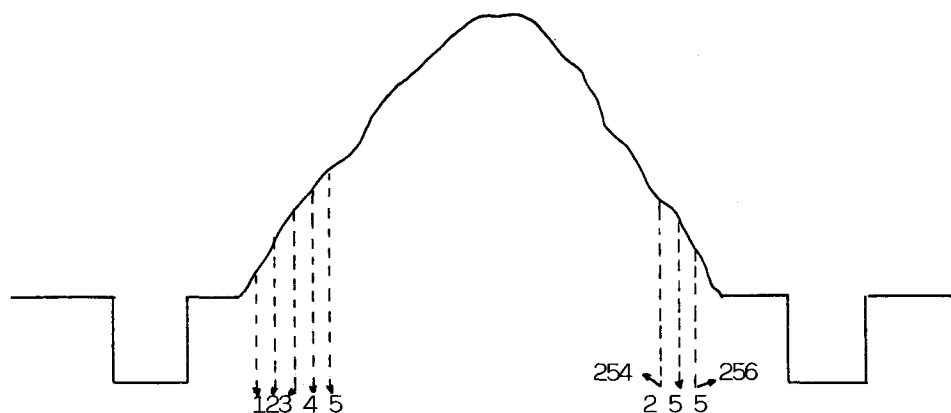
Este digitalizador es un aparato que transforma una imagen de vídeo monocroma congelada en palabras digitales, que representan la luminancia de las coordenadas (X, Y) de dicha imagen. Cada palabra está formada por 8 bits, por lo que podemos representar por cada pixel, 256 niveles de grises distintos. La resolución máxima de la imagen digitalizada será de $256 * 256$ pixels.

La fuente de vídeo a digitalizar ha de ser la misma durante unos segundos ya que el proceso no se efectúa en tiempo real. Cuando comienza la digitalización de la imagen, en cada campo se digitaliza un pixel por cada línea, es decir una columna.

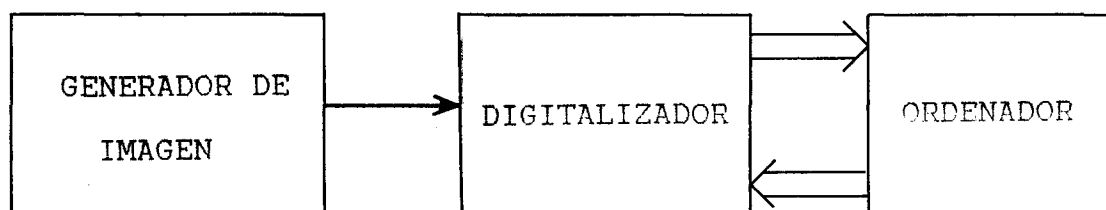


Por cada sincronismo de cuadro, (FS) se incrementa el tiempo que transcurre desde el sincronismo de línea (LS) y la captura del pixel. De este modo, en el primer cuadro, digitalizaremos la primera columna; en el segundo cuadro, la segunda columna; etc, hasta completar las 256 columnas.

Suponiendo ésta una línea genérica, en el primer cuadro se digitalizará el pto 1, en el segundo en el pto 2, y así hasta el 256.



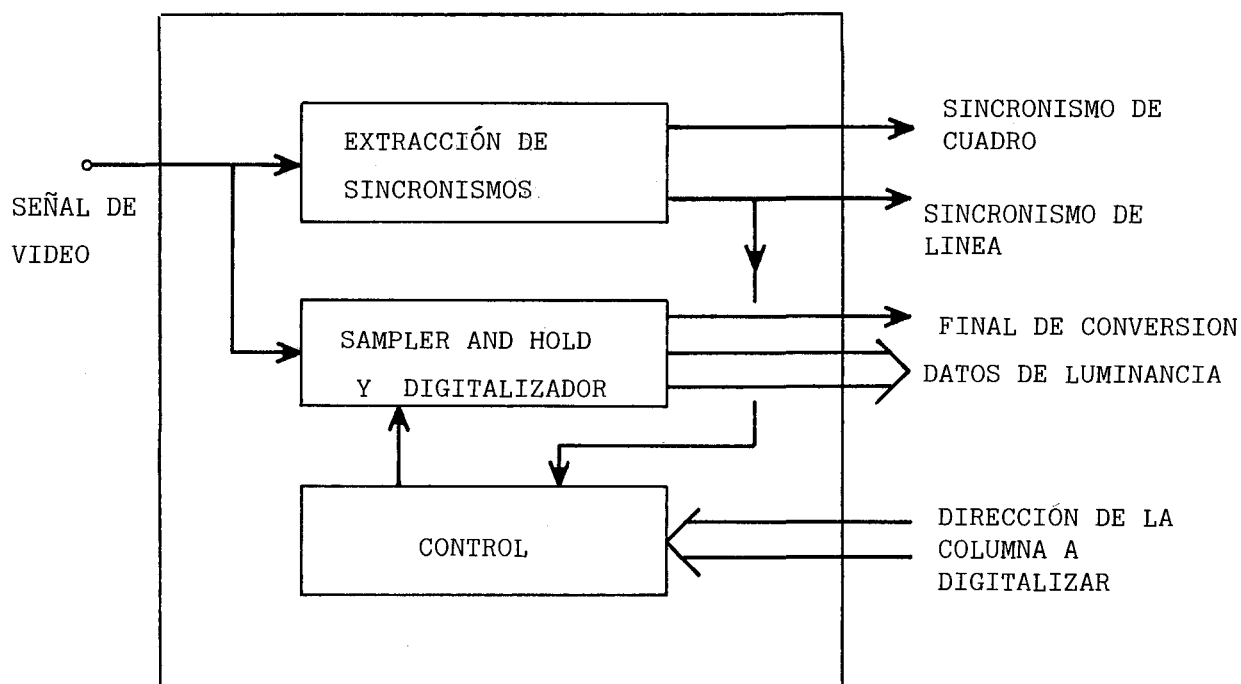
La interconexión del digitalizador se muestra en la siguiente figura:



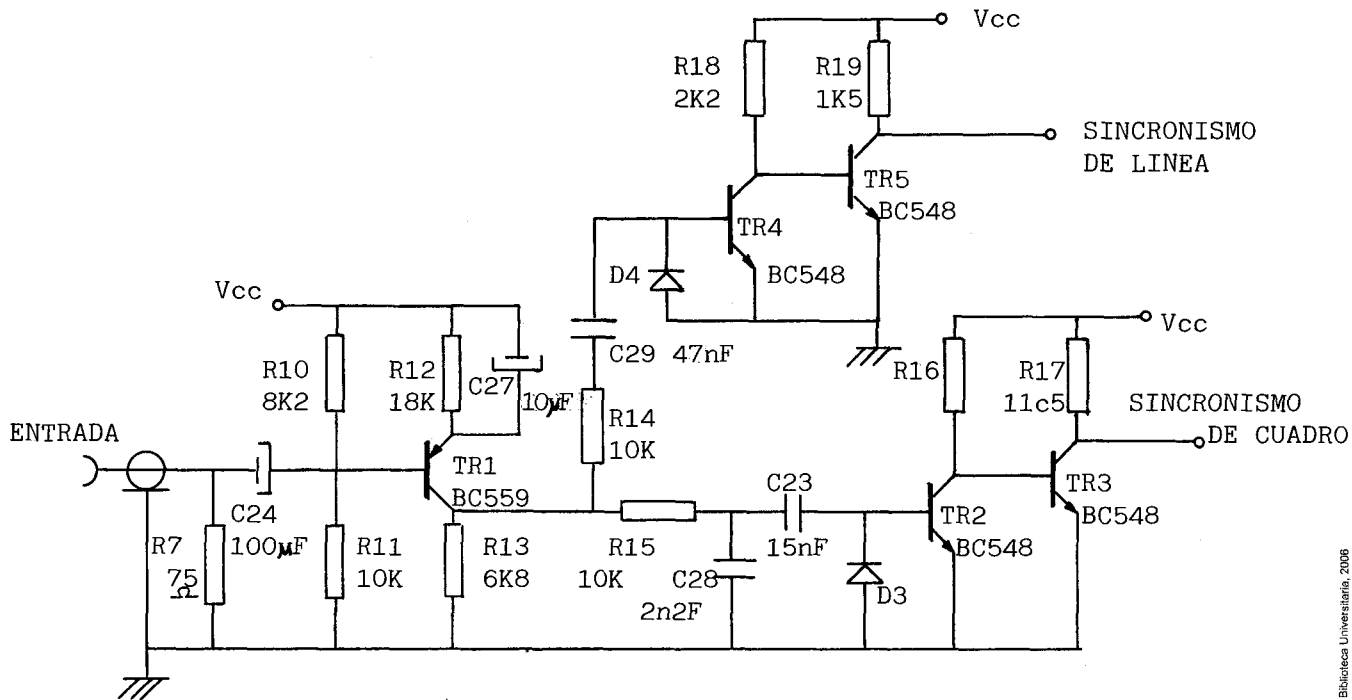
El ordenador se encargará del control del digitalizador, dictándole qué columna ha de digitalizar y recogiendo la información que este va generando. Progresivamente irá incrementando la columna hasta barrer la imagen a digitalizar por completo. De este modo, se requiere programar al ordenador en ensamblador, programa que será descrito más adelante, guardando la información de cada pixel en lugares consecutivos de memoria. Por ello, el ordenador reserva un bloque de memoria para almacenar la imagen.

Una vez presentada la filosofía de funcionamiento, veamos más detenidamente el digitalizador.

Lo podríamos dividir en tres bloques. Uno se encargaría de la extracción de sincronismos tanto de línea como de cuadro, para ser enviados al ordenador y al control del digitalizador. El bloque de control generará los impulsos con los retardos adecuados en cada línea, para que un tercer bloque capte el nivel de tensión en ese instante (Sampler and hold) y lo codifique en una palabra de 8 bits, que a continuación enviará al ordenador.

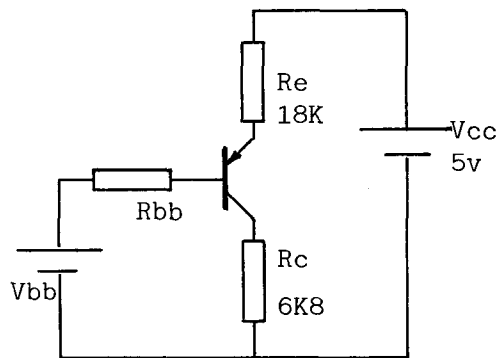
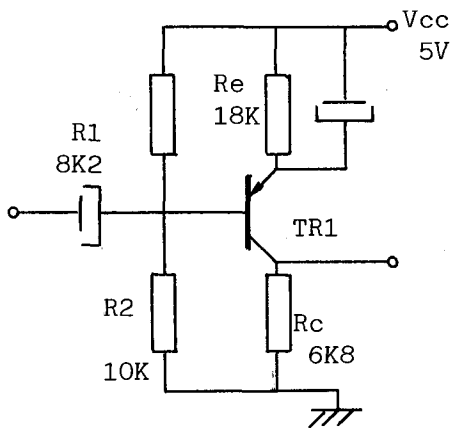


A) Bloque de extracción de sincronismos.



En la figura podemos observar este bloque, formado por varias partes. La señal de entrada es cargada con la impedancia característica del cable coaxial, 75 Ω . Esta se encuentra con una primera parte formada por un transistor PNP BC 559, montado en emisor común, que eleva el nivel de la señal y la invierte, de forma que a la salida de ésta etapa los impulsos de sincronismo serán positivos y la información de luminancia, serán niveles negativos.

Veamos el punto de trabajo de este transistor:



$$R_{bb} = R1 // R2 = \frac{R1 \cdot R2}{R1 + R2}; \quad R_{bb} = 4.5 \text{ K}\Omega$$

$$V_{bb} \approx V_{cc} \cdot \frac{R2}{R1 + R2}; \quad V_{bb} = 2.75 \text{ v.}$$

$$V_{cc} = I_e \cdot R_e - V_{ce} + I_c \cdot R_c$$

$$V_{cc} = I_e \cdot R_e - V_{be} - I_b \cdot R_{bb} + V_{bb}$$

Suponiendo que $I_c \approx I_e$

$$V_{cc} = I_c \cdot (R_e + R_c) - V_{ce}$$

$$V_{cc} = I_c \cdot R_e - V_{be} - I_b \cdot R_{bb} + V_{bb}$$

Aproximando $I_c = \beta \cdot I_b$

$$V_{cc} = \beta \cdot I_b \cdot R_e - I_b \cdot R_{bb} - V_{be} + V_{bb}$$

$$V_{cc} = (\beta \cdot R_e - R_{bb}) \cdot I_b - V_{be} + V_{bb}$$

$$V_{cc} + V_{be} - V_{bb} = I_b \cdot (\beta \cdot R_e - R_{bb})$$

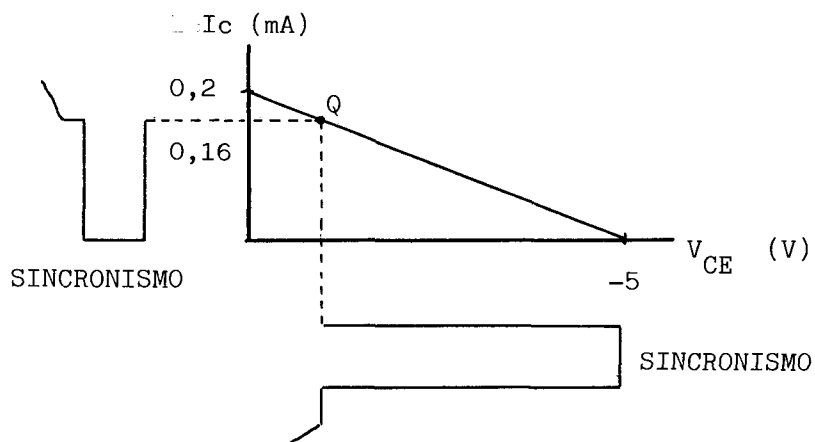
$$I_b = \frac{V_{cc} + V_{be} - V_{bb}}{\beta \cdot R_e - R_{bb}} \quad \begin{array}{l} V_{be} \approx 0.68 \text{ v.} \\ \beta \approx 100 \end{array}$$

$$I_b \approx 1.63 \text{ }\mu\text{A.} \quad ; \quad I_c \approx 0.163 \text{ mA.}$$

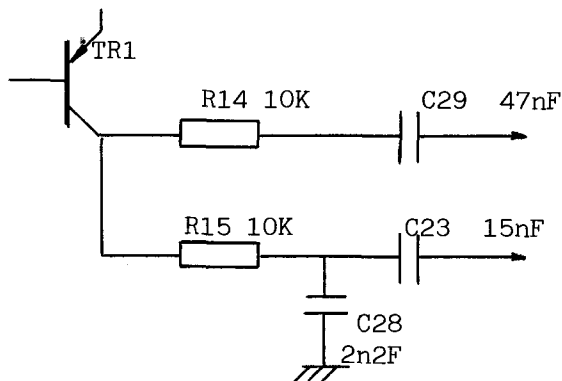
$$V_{ce} \approx -1 \text{ v.}$$

Nos preocupamos más de los sincronismos que de la señal de luminancia. El nivel de los sincronismos es de

0.25 voltios; como esta etapa tiene una ganancia considerable, la presencia de un sincronismo hace que el transmisor se corte rapidamente, obteniendo la máxima tensión que es la de la fuente = 5 V.

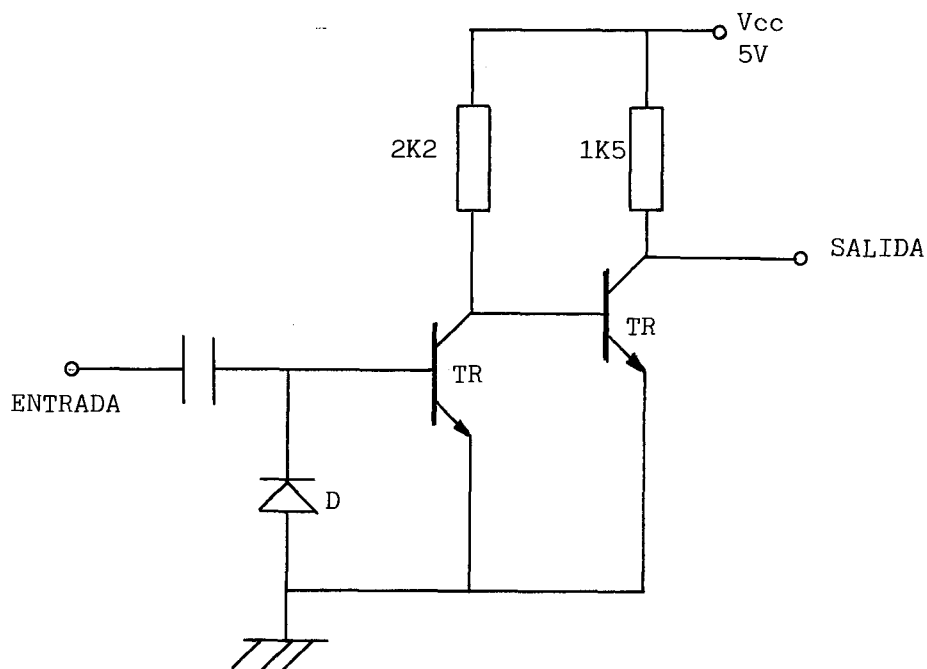


A partir de este punto la señal se divide atacando dos etapas en cascada, cuya misión es extraer los sincronismos de línea, por un lado, y los de cuadro por otro.



Ambas etapas son idénticas, a excepción de un condensador derivado a masa de 2.2 nF (C28) en la etapa de extracción de sincronismos de cuadro. La misión de este condensador es enviar a masa los impulsos de corta duración, dejando que prosperen los impulsos de cuadro.

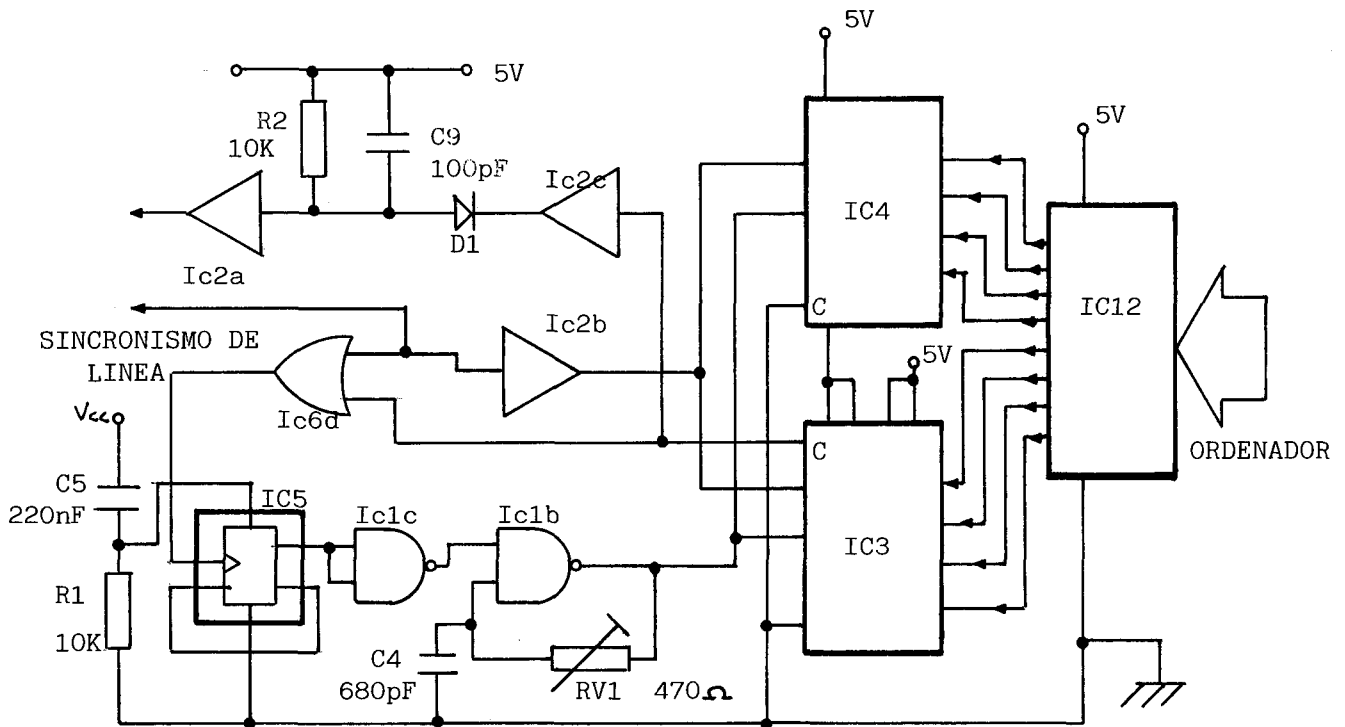
La etapa en cascada es la siguiente:



De la señal que llega, los niveles negativos (correspondientes a la información de luminancia) son mandados a tierra por el diodo. En ausencia de señal, el primer transistor se encuentra cortado y el segundo saturado. Con la llegada de un impulso positivo, entra en conducción rápidamente el primer transistor, cortando el segundo transistor.

A la salida de esta etapa tendremos pulsos de 5 Vp y de igual duración a la del impulso de sincronismo que llege.

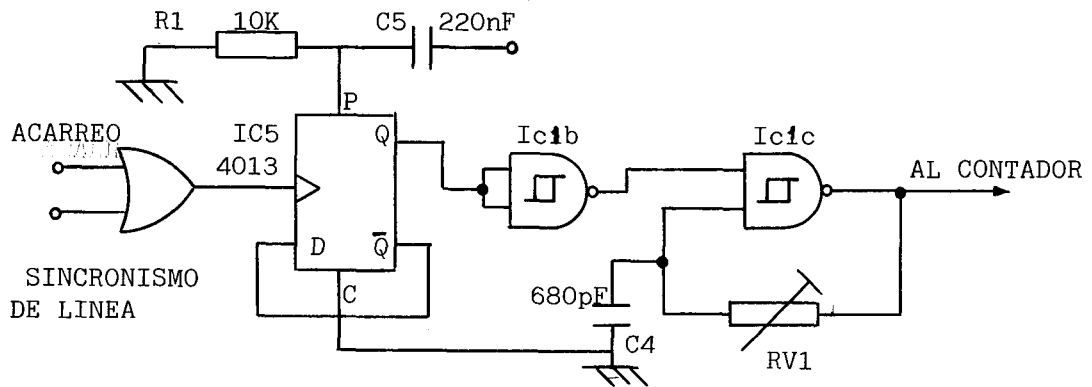
B) Bloque de control.



El circuito integrado 74HC373 está formado por 8 biestables tipo D, los cuales almacenarán la dirección enviada por el ordenador durante la digitalización de la columna completa. Las salidas de estos 8 latch están conectadas a las entradas de preset de 2 contadores, de 4 bits 74HC161, conectados en cascada (IC3, IC4). Con la llegada de un impulso de sincronismo de línea, la dirección almacenada en los registros del CI 74HC373 pasan al contador de 8 bits, IC3, IC4, previamente invertido (el impulso de sincronismo de línea) por el inversor IC2b.

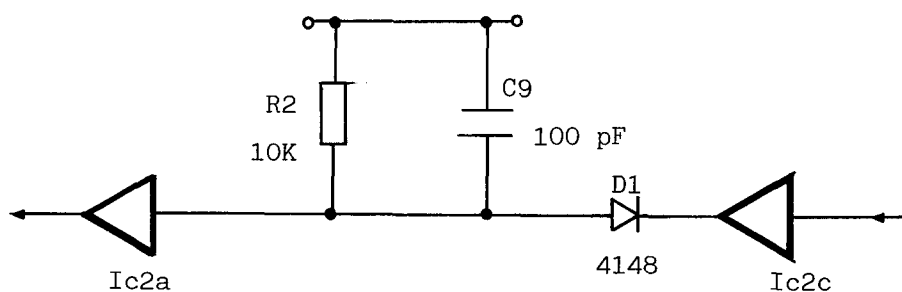
Este impulso de sincronismo también actúa sobre la entrada de reloj del biestable IC5 4013, a través de una puerta OR IC6d (4071). El biestable IC5 que

inicialmente está a nivel bajo pasa a estado alto.



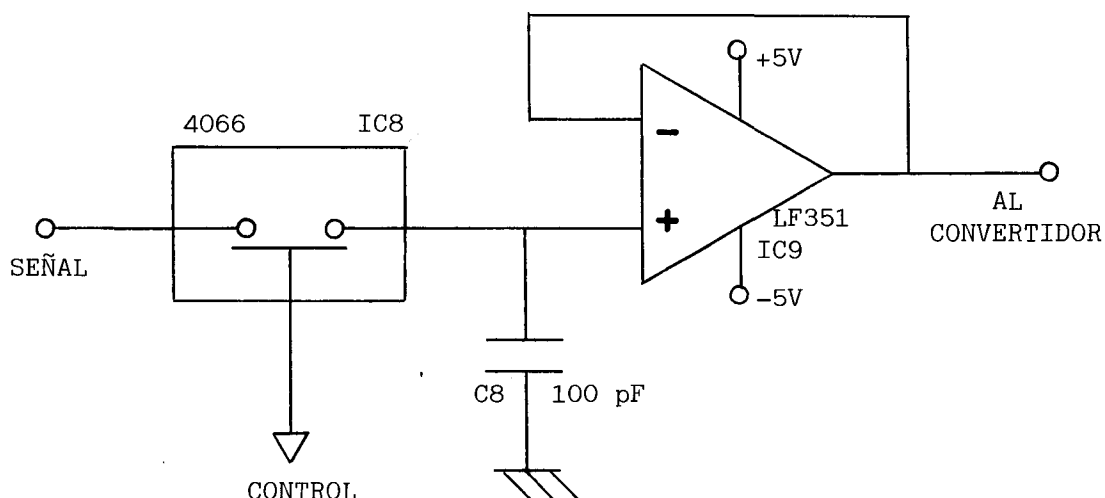
Este nivel alto se convierte en un nivel bajo al pasar por la puerta NAND IC1c utilizada como inversor Schmitt al cortocircuitar sus entradas, habilitando el oscilador formado por IC1b, RV1 y C4. La frecuencia de oscilación de éste viene dada por la constante de tiempo RC de RV1 y C4. Esta oscilación conectada a la entrada de reloj del contador IC3, IC4 determinará los retardos que se deben introducir en cada línea, desde el sincronismo de línea hasta que se atrape el nivel de tensión. Por tanto si hemos de tomar 256 muestras durante $64\mu\text{sg}$ que dura una línea, la frecuencia de oscilación deberá ser $(1 / (64\mu\text{sg} / 256)) = 4 \text{ MHz}$. Variando el valor de la resistencia ajustable RV1 podemos ajustar la oscilación a 4 MHz. Como se dijo anteriormente, los 4 MHz del oscilador estén conectados al contador IC3,4 que empezará a contar a partir del valor de la columna en curso previamente almacenada en éste.

En la transición de 1111 1111 a 0000 0000 se genera un acarreo que se utiliza para poner a 0 el biestable IC5 deshabilitando el oscilador. Además, este impulso de acarreo, invertirlo por IC2c carga el condensador C9, que al descargarse a través de R2, prolonga el impulso aproximadamente $1\mu\text{sg} = C9 \cdot R2$.



Este pulso prolongado negativo se invierte con IC2a y hace que el interruptor analógico IC8 se cierre, además de actuar sobre la entrada 'Start of conversion' del convertidor analógico digital de aproximaciones sucesivas ZN427 (IC10). El interruptor analógico y el convertidor analógico digital pertenecen al bloque que aún queda por describir.

C) Bloque de Sample and Hold y digitalización.



La señal de entrada, aparte de actuar sobre el bloque de extracción de sincronismos, lo hace también sobre el interruptor analógico 4066 (IC8). Previamente con la resistencia ajustable RV2, ajustamos el nivel de negros de la señal de vídeo.

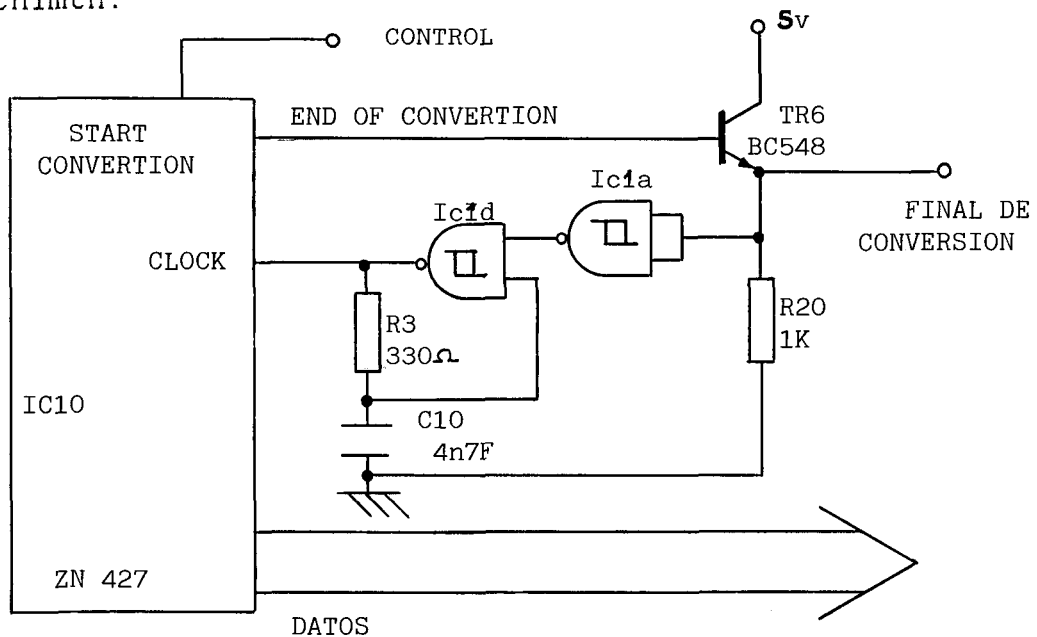
El circuito S&H lo forman el interruptor analógico IC8, el condensador de retención C8 y el amplificador operacional IC9.

La alimentación simétrica la conseguiremos con el circuito integrado 7660 (IC7), circuito doblador de tensión que nos proporciona +5 y -5 voltios.

Como vimos en el bloque de control, la salida del inversor IC2a actúa sobre la puerta del interruptor analógico, y después del inversor IC2f, actúa sobre la entrada "Start of conversión" del convertidor analógico

digital IC10. De este modo, cuando hay un acarreo del contador IC3,4 alargado por C9-R2, se cierra el interruptor analógico haciendo que se cargue el condensador de retención C8. Cuando el pulso finaliza, el interruptor se abre quedando atrapada la tensión en el condensador. Este valor de tensión se mantiene debido a la alta impedancia del amplificador operacional LF351 (IC9), conectado como seguidor de tensión de ganancia unidad y se aplica al convertidor analógico digital.

En este momento, la salida de final de conversión del convertidor IC10 pasa a nivel bajo, habilitando un nuevo oscilador formado por IC1d, R3 y C10, a través de TR6 e IC1a utilizado este último como inversor Schmitt.



La frecuencia de este oscilador es de $(1/(R3 \cdot C10)) = 645 \text{ KHz}$ que se aplica al convertidor ZN427. Después del noveno ciclo, el ADC pone la salida de final

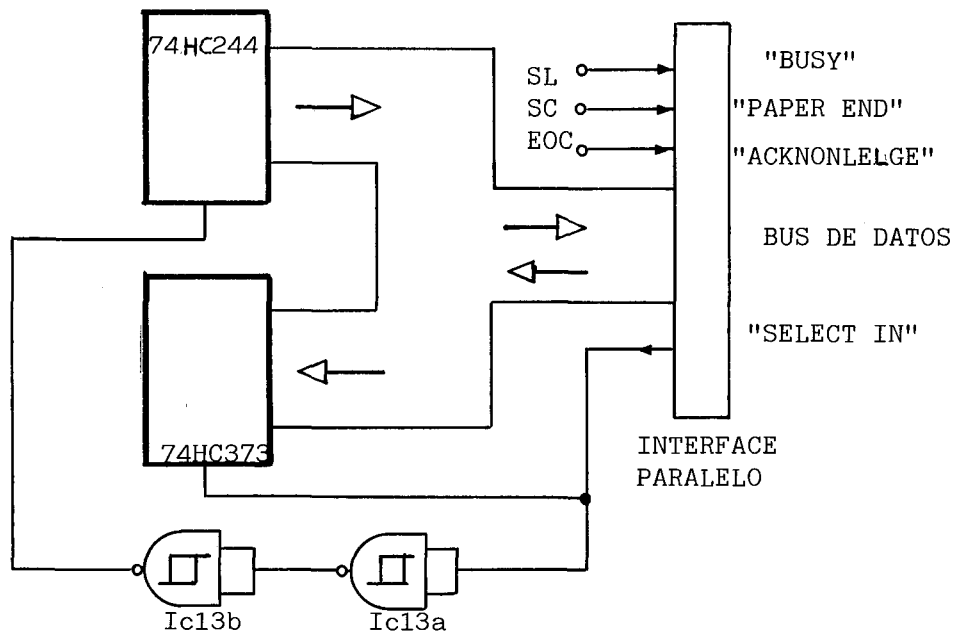
de conversión a estado alto, entrando en conducción TR6 y deshabilitando el oscilador. El emisor de TR6 también se puentea al ordenador, indicando un final de conversión cuando se produce una transición de "0" a "1" pudiéndose leer en este momento la salida del ADC.

Por último, la salida de datos del ADC se conecta al octuple buffer 74LS244 (IC11). En este momento se ha concluido todo el ciclo y el ordenador guarda la información. Con la llegada de un nuevo sincronismo de línea, se pone a "1" el biestable IC5, se habilita el primer oscilador y se repite todo el proceso.

Después de 256 impulsos de sincronismos de línea, el ordenador interpreta que ya ha almacenado la columna, decrementa esta y la almacena de nuevo en los 8 latch (IC12), empezando después del sincronismo de cuadro la conversión de la nueva columna.

Como el retardo introducido en cada línea para capturar el nivel de tensión está basado en el acarreo, producido por el contador IC3,4, la primera columna viene representada por el valor binario del número 256 y la correspondiente a la última columna, 1 en binario.

La conexión se hace a través del puerto paralelo que describiremos más adelante.

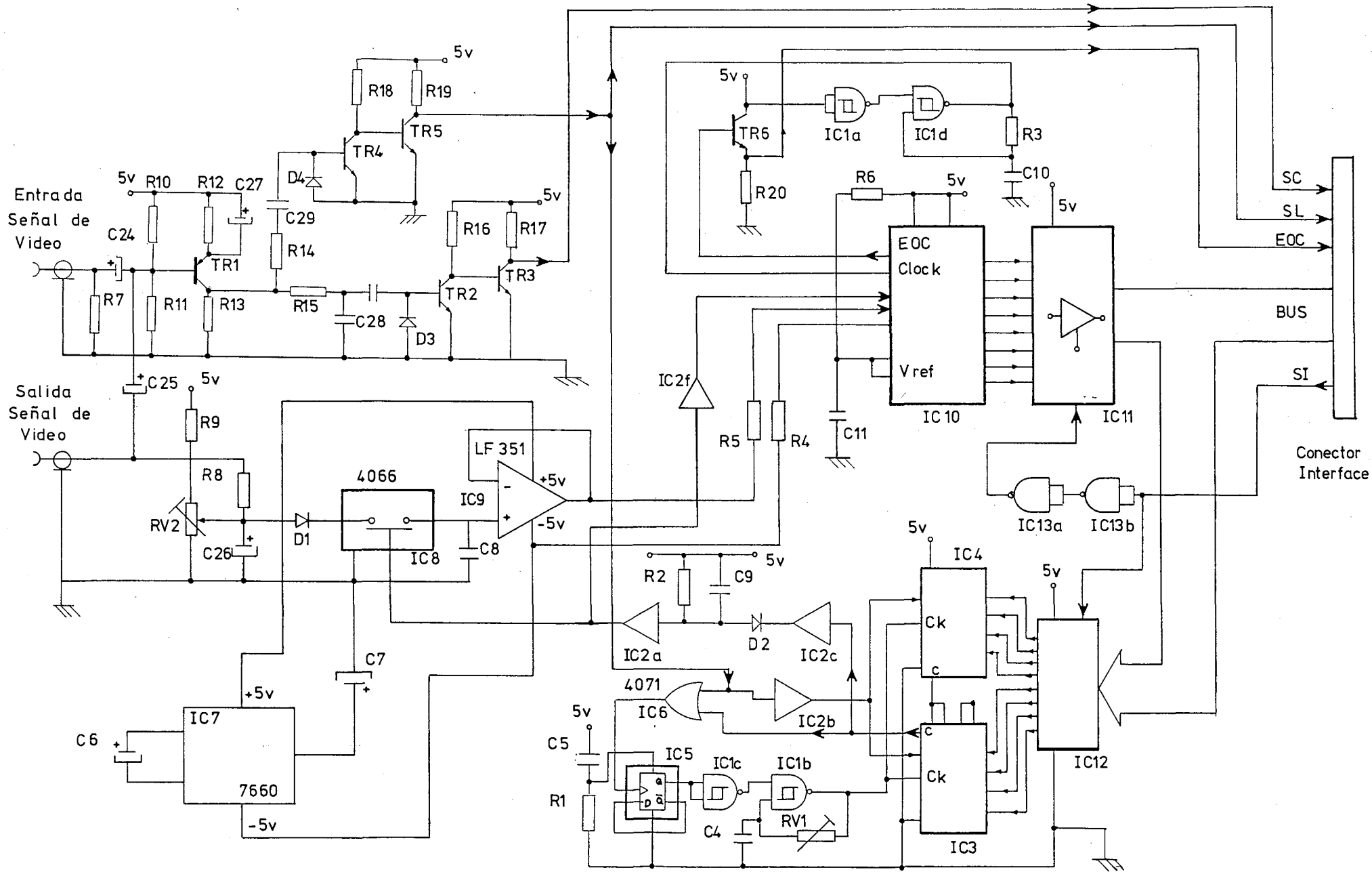


El puerto paralelo dispone de un bus de datos bidireccional de 8 bits y una serie de líneas de control. Como hemos visto, el puerto paralelo es principalmente para la conexión de impresoras, pero programandolo en ensamblador lo podemos emplear para otros fines.

Una de las líneas de control de las que dispone el interface la emplearemos para el control del bus, ya que por él vamos a enviar datos al ordenador y recibir direcciones del ordenador. Cuando esta línea se encuentre a alto nivel, el ordenador tendrá el bus libre para enviar una dirección y cuando se encuentre a nivel bajo, el digitalizador podrá enviar datos al ordenador.

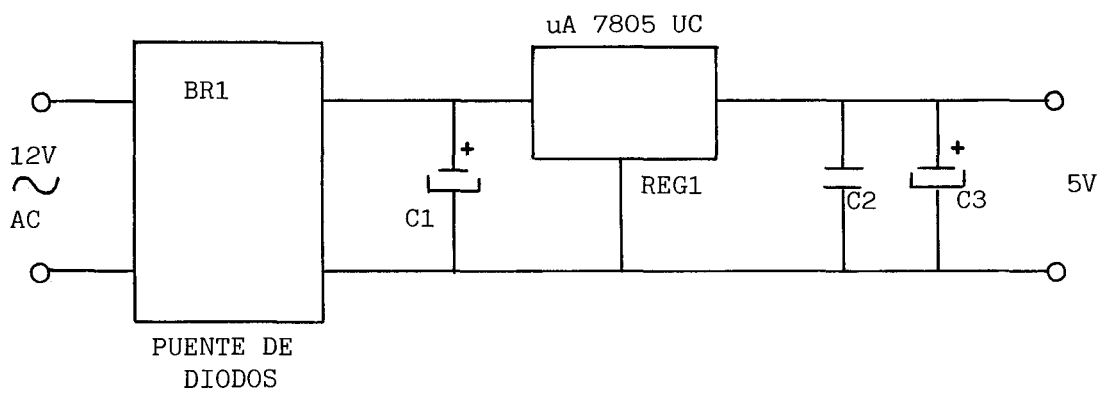
Inicialmente, el ordenador pone a nivel alto esta línea de control del bus, la cual va unida al "Load

Enable" de los 8 biestables tipo D del circuito integrado 74HC373 (IC12). También esta línea de control está unida a la entrada de validación del octuple buffer no inversor 74LS244 (IC11), a través de dos puertas NAND (empleadas como inversoras al cortocircuitar sus entradas). La misión de estas puertas es producir un cierto retardo entre el nivel bajo, que aparece en el octuple biestable IC12, el cual hace que retenga la información presente en el bus y el estado bajo que aparece en el octuple buffer IC11, que hace que sean transparente introduciendo en el bus un nuevo dato. De esta forma se evita que el octuple biestable tipo D capture una información distinta a la dirección que el ordenador envía.



D) Alimentación.

Para alimentar el conjunto del circuito utilizaremos un regulador de tensión monolítico de 5 voltios, montado en una configuración típica y con un transformador de 12 voltios, 1 Amperio.



El consumo del conjunto no es muy elevado debido a la utilización de circuitos integrados CMOS y de la serie 74HC. El regulador monolítico μ A7805 UC nos permite una máxima corriente de salida de 1 Amperio, utilizando un pequeño disipador de calor.

*PROGRAMACION EN
ENSAMBLADOR.*

PROGRAMACION EN ENSAMBLADOR.

Los microprocesadores Intel 8086 y 8088 se desarrollaron a partir de un procesador anterior: el 8080. Este ha sido el microprocesador de 8 bits de mayor éxito incluyendo el Zilog Z-80.

Tanto el 8086 como el 8088 poseen una arquitectura interna de 16 bits y son capaces de trabajar con operandos de 8 y 16 bits. Ambos microprocesadores tienen un bus de direcciones de 20 bits, que ofrece una capacidad de direccionamiento de un megabyte (1.048.576 bytes). El juego de instrucciones que comparten también es idéntico. La diferencia entre ambos es que aunque el 8088 responde a todas las instrucciones del 8086, sólo puede transferir datos de byte en byte, con un incremento considerable en los tiempos de ejecución.

Las longitudes de los datos con los que pueden trabajar estos microprocesadores son: 1, 4, 8 y 16 bits (bit, nibble, byte y palabra).

A) Segmentación.

Aunque tanto el 8086 como el 8088 utilizan un bus de direcciones de 20 bits, con una capacidad de direccionamiento de un megabyte, sus registros internos son de 16 bits (como veremos más adelante.) ofreciendo una capacidad máxima de direccionamiento de 64 Kbytes.

Para superar esta restricción, la memoria se divide en un número arbitrario de segmentos, cada uno de

los cuales contiene como máximo 64 Kbytes. Cada segmento empieza en una localización cuya dirección es divisible por 16 y que se conoce como dirección del segmento. Para acceder a los bytes individuales, o a sus palabras, se emplea una dirección adicional llamada desplazamiento u Offset, que apunta a la dirección de un byte exacto dentro del segmento de 64K designado por la dirección del segmento. Las direcciones se crean pues, combinando la dirección del segmento de 16 bits y una dirección relativa de 16 bits. La dirección del segmento se trata como si estuviese desplazada 4 bits. Cuando se añade la dirección relativa se obtiene una dirección completa de 20 bits, como se muestra en la figura:

1011 1011 1010 0011 0000	Dirección del segmento.
1011 1010 0110 0111	Dirección relativa.
<hr/>	
1100 0111 0100 1001 0111	Dirección segmentada de 20 bits.

B) Registros del 8088.

El 8088 para ejecutar instrucciones y llevar a cabo operaciones aritméticas y lógicas emplea registros de 16 bits.

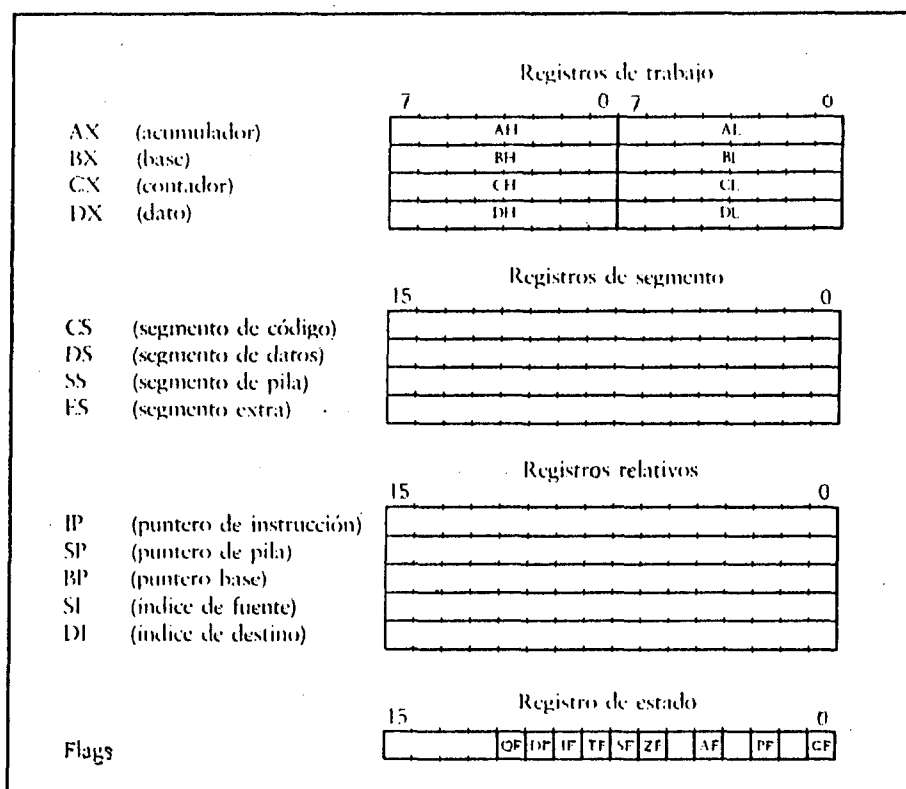
Hay 14 registros en total, cada uno de ellos con un objetivo determinado: Cuatro registros de trabajo que se utilizan temporalmente por los programas para almacenar resultados intermedios y los operandos que necesitan las operaciones aritméticas y lógicas.

Cuatro registros de segmento que almacenan las

direcciones de inicio de ciertos segmentos en la memoria.

Cinco registros relativos que almacenan las direcciones de segmento para indicar donde se sitúan los datos en la memoria.

Finalmente, hay un registro de estado, formado por 9 bits, que se emplea para registrar la información del estado del 8088 y de sus operaciones de control.



B1)Registros de trabajo.

Cuando un ordenador procesa datos, gran parte del tiempo el microprocesador lo emplea en traer y llevar datos a la memoria. Este tiempo se puede reducir en muchas ocasiones guardando en el mismo microprocesador los operandos más empleados y los resultados obtenidos. Hay cuatro registros de 16 bits con este propósito siendo reconocidos como: AX, BX, CX y DX. Cada uno de ellos se puede subdividir a su vez y direccionarse de forma separada, como si fuesen dos medios registros de 8 bits. Estos se conocen a su vez como: AH, BH, CH y DH si son de mayor orden y AL, BL, CL y DL si son de menor orden.

El empleo de unos u otros puede variar según el programador.

Aunque estos registros se encuentran disponibles para realizar cualquier tipo de trabajo, cada uno de ellos tiene asignado un uso específico. Por ejemplo:

- El registro AX es el acumulador y es el registro principal para realizar las operaciones aritméticas.
- El registro BX se utiliza para apuntar el comienzo de una tabla de memoria. También se emplea para almacenar la parte relativa de una dirección segmentada.
- El registro CX es utilizado como un contador de

repetición para control de bucles. Por ejemplo, la instrucción LOOP hace uso del mismo para almacenar el contador que indica el número de iteraciones del bucle. Ninguno de los demás registros realiza esta función.

B2)Registros de Segmento.

Como ya hemos visto, la dirección completa de una posición de memoria consiste en la dirección de un segmento de 64K y la dirección relativa dentro del segmento.

Los cuatro registros, llamados CS, DS, SS y ES, se utilizan para identificar cuatro segmentos de 64K específicos de memoria.

El registro CS localiza el segmento del código, es decir, la dirección a partir de la cual empieza el programa que está siendo ejecutado.

El registro DS localiza el segmento de datos, que es la dirección donde empieza el área de memoria que almacena los datos que se están empleando.

El registro SS localiza el segmento de pila, que es la dirección donde comienza un lugar de trabajo temporal que almacena parámetros y direcciones que utiliza el programa activo.

El cuarto registro de segmento, ES, apunta hacia un segmento extra, utilizado normalmente como suplemento del segmento de datos, con objeto de que puedan emplearse más de 64K de memoria para

almacenamiento.

B3)Registros Relativos.

Los cinco registros relativos se utilizan para localizar un byte en concreto, dentro de un segmento específico de 64K . El registro llamado Puntero de Instrucción (IP), indica cuál es la dirección donde se encuentra situada la instrucción que se va a ejecutar dentro del segmento de código.

Los programas no tienen acceso directo al registro IP, pero hay algunas instrucciones que cambian su contenido como es el caso del JMP y CALL. Hay dos más que se denominan registro de puntero de pila (SP) y puntero de base (BP), quienes proporcionan direcciones relativas dentro del segmento de pila. El registro BP se utiliza para indicar, partiendo de la posición inicial de la pila, donde se encuentra determinada información.

Los dos registros que quedan, denominados registros de índice, son el SI o índice de fuente y el DI o índice de destino. Estos se utilizan normalmente con otro registro (AX, BX, CX o DX) o con ciertas instrucciones, que proporcionan la dirección relativa del lugar inicial de un campo de datos comprendido en el interior de un segmento de datos.

B4)Registro de estado.

El último registro del 8088, llamado registro de estado, es una colección de bits individuales de control denominados indicadores o Flags. Están organizados en forma de registro, de manera que pueden ser guardados o recuperados como si fuesen un dato. Hay nueve indicadores de un bit en el registro de estado de 16 bits y los otros siete no se utilizan.

Estos indicadores se pueden dividir de forma lógica en dos grupos: seis indicadores de estado, que se utilizan para registrar la información del estado del microprocesador y tres indicadores de control que dirigen algunas de las instrucciones.

- CF Indicador de acarreo.
- OF Indicador de desbordamiento aritmético.
- ZF Indicador de cero.(Indica resultado cero o igual comparación).
- SF Indicador de signo.(Indica resultado o comparación negativa).
- PF Indicador de paridad.
- AF Indicador de acarreo auxiliar.(Indica ajuste necesario en operaciones aritméticas decimales codificadas en binario (BCD)).
- DF Indicador de dirección.
- IF Indicador de interrupción.(Controla si están habilitadas las interrupciones).
- TF Indicador de interrupción de Software.

C) Modos de direccionamiento.

Los modos de direccionamiento utilizados por el 8086 y el 8088 son los siguientes:

- De registro
- Inmediato
- Directo
- Indirecto
- Indirecto con base
- Indirecto con índice
- Indirecto con índice y base.

C1) Direccionamiento inmediato.

El operando es una constante situada después del código de instrucción. Por ejemplo, la instrucción `MOV BL,100` traslada el valor de la constante 100 al registro BL.

C2) Direccionamiento de registro.

Los operandos son los contenidos de registros indicados y en el caso de registros de datos, pueden tener longitud de byte o de palabra. Por ejemplo en,

```
MOV AH,BL
```

se transfiere el valor del byte en el registro BL al registro AH.

C3)Direccionamiento directo.

El operando está situado en una dirección indicada en la instrucción. Por ejemplo, `MOV AX,DIR` transfiere el valor de la dirección apuntada por la etiqueta `DIR` al registro `AX`. "`DIR`" es la dirección relativa, siendo la dirección del segmento la que indica `DS`.

C4)Direccionamiento indirecto.

El operando se encuentra en una dirección señalada por un registro índice o base como `BX`, `BP`, `SI` ó `DI`. Por ejemplo, `MOV [CX, BX]`. El operando origen se encuentra en la dirección segmentada determinada por el contenido del registro `BX` y el segmento `DS`.

C5)Registro indirecto con base.

El operando se encuentra en una dirección segmentada formada por la suma de un registro interno (`BX` ó `BP`) y un valor de desplazamiento de 8 ó 6 bits, además del segmento `DS`. Ejemplo: `MOV AX, [BX+DESP]`.

C6)Registro indirecto con índice.

Análogo al anterior pero con un registro índice (`SI` ó `DI`). Ejemplo: `MOV AX, [DI+DESP]`.

C7)Registro indirecto con base e índice.

En este caso la dirección relativa de la dirección segmentada se obtiene por la suma de dos

registros internos y opcionalmente un registro de desplazamiento de 8 ó 16 bits.

Ejemplo: MOV AX, [BX+DI+DESP].

D) Interrupciones.

Siempre que un dispositivo de hardware, o un programa, necesita la atención de la CPU envía una señal, o instrucción, llamada Interrupción, al microprocesador, identificando el tipo de tarea que quiere realizar. Cuando se recibe una señal de interrupción, se activa una rutina almacenada en memoria, llamada controlador o gestor de interrupción, asociada al número particular de la interrupción.

Después de que la rutina de tratamiento de la interrupción ha realizado su tarea, la actividad del ordenador normalmente continúa donde estaba cuando ocurrió la interrupción.

Hay tres tipos principales de interrupciones. En primer lugar, hay interrupciones generadas por la circuitería del ordenador en respuesta a algún acontecimiento, tal como la presión de una tecla en el teclado. Otras interrupciones son generadas por la CPU como resultado de alguna actuación inusual producida por el programa, como por ejemplo, una división por cero.

El tercer tipo de interrupciones son generadas deliberadamente por los programas como una forma de

llamar subrutinas almacenadas en RAM ó ROM. Estas interrupciones son muy utilizadas y se denominan interrupciones por Software. Las interrupciones guardan automáticamente los valores contenidos en los registros CS e IP, de forma que el ordenador puede volver al sitio donde estaba trabajando cuando se produjo dicha interrupción.

E) Instrucciones clasificadas por categorías.

E1) Propósito general.

MOV	Destino, Origen	:Tranferencia.
NOP		:Operación nula.
PUSH	Origen	:Introducir en la pila.
POP	Destino	:Extraer de la pila.
XCHG	Destino, Origen	:Intercambio.
XLAT	Tabla	:Traducción.

E2) E/S.

IN	Acumulador, Puerto	:Entrada.
OUT	Puerto, Acumulador	:Salida.

E3) Carga de direcciones.

LEA	Destino, Origen	:Carga la dirección efectiva.
LDS	Destino, Origen	:Carga el puntero usando DS.
LES	Destino, Origen	:Carga el puntero usando ES.

E4) Almacenamiento y Recuperación de indicadores.

LAHF		:Carga AH con los indicadores.
POPF		:Extrae los indicadores de la pila.
PUSHF		:Introduce los indicadores en la pila.
SAHF		:Almacena AH en el registro de indicadores.

E5) Aritméticas.

Suma:

AAA		:Ajuste ASCII para la suma.
ADC	Destino, Origen	:Suma con acarreo.
ADD	Destino, Origen	:Suma.
DAA		:Ajuste decimal para la suma.
INC	Destino	:Incremento.

Resta:

AAS		:Ajuste ASCII para la resta.
CMP	Destino, Origen	:Comparación.
DAS		:Ajuste decimal para la resta.
DEC	Destino	:Decremento.
NEG	Destino	:Negación.
SBB	Destino, Origen	:Resta con acarreo.
SUB	Destino, Origen	:Resta.

Multiplicación:

AMM	:Ajuste ASCII para la multiplicación.
IMUL Origen	:Multiplicación entera.
MUL Origen	:Multiplicación sin signo.

División:

ADD	:Ajuste ASCII para la división.
CBW	:Convierte byte en palabra.
CWD	:Convierte palabra en palabra doble.
DIV Origen	:División sin signo.
IDIV Origen	:División entera.

E6)Operaciones de cadena.

CMPS	:Compara cadenas.
INS	:Entrada de cadena.
LODS Origen	:Carga cadena.
MOVS Destino, Origen	:Transferencia de cadena.
OUTS	:Salida de cadena.
REP Destino, Origen	:Repetición.
REPE Destino, Origen	:Repetición mientras igual.
REPNE Destino	:Repetición mientras no igual.
REPZ Destino	:Repetición mientras no cero.
REPZ Destino, Origen	:Repetición mientras cero.
SCAS Cadena destino	:Explora cadena.
STOS Cadena destino	:Almacena cadena.

E7)Operaciones lógicas.

AND Destino, Origen :Intersección lógica.
NOT Destino :No lógico.
OR Destino, Origen :O lógico.
TEST Destino, Origen :Comparación lógica.
XOR Destino, Origen :O exclusivo lógico.

E8)Operaciones de rotación y desplazamiento.

RCL Destino, valor contador:Rotación a la izqda. con
acarreo.
RCR Destino, valor contador:Rotación a la dcha. con
acarreo.
ROL Destino, valor contador:Rotación a la izqda.
ROR Destino, valor contador:Rotación a la dcha.
SAL Destino, valor contador:Desplazamiento aritmé-
tico a la izqd.
SAR Destino, valor contador:Desplazamiento aritmé-
tico a la dcha.
SHL Destino, valor contador:Desplazamiento lógico
a la izqda.
SHR Destino, valor contador:Desplazamiento lógico
a la dcha.

E9)Operaciones de los indicadores.

CLC :Baja el indicador de acarreo.
CLD :Baja el indicador de dirección.
CLI :Baja el indicador de interrupción.
CMC :Complementa el indicador de

acarreo.

STC :Pone a uno el indicador de
acarreo.

STD :Pone a uno el indicador de
dirección.

STI :Pone a uno el indicador de
interrupción.

E10)Control del procesador.

ESC Código operación, addr.:Salida a un coprocesador.

HLT :Parada hasta interrupción o
reset.

LOCK :Bloqueo del bus.

WAIT :Espera hasta la activación de
la línea test.

E11)Transferencia de control.

Incondicional:

CALL Nombre-Rutina :Llamada a subrutina.

JMP Destino :Salto.

RET :Retorno de subrutina.

Condicional:

JA Destino (8 bits con signo):Salto si superior.

JAE :Salto si superior o igual.

JB :Salto si inferior.

JBE :Salto si inferior o igual.

JC	:Salto si acarreo.
JCXZ	:Salto si CX=0.
JE	:Salto si igual.
JG	:Salto si mayor que.
JGE	:Salto si mayor o igual.
JL	:Salto si menor.
JLE	:Salto si menor o igual.
JNA	:Salto si no superior.
JNAE	:Salto si no superior o igual.
JNB	:Salto si no inferior.
JNBE	:Salto si no inferior o igual.
JNC	:Salto si no acarreo.
JNE	:Salto si no igual.
JNG	:Salto si no mayor que.
JNGE	:Salto si no mayor o igual.
JNL	:Salto si no menor.
JNLE	:Salto si no menor o igual.
JNO	:Salto si no desbordamiento.
JNS	:Salto si no signo.
JNP	:Salto si no paridad.
JNZ	:Salto si distinto de cero.
JO	:Salto si desbordamiento.
JP	:Salto si paridad par.
JPE	:Salto si paridad igual.
JPO	:Salto si paridad impar.
JS	:Salto si signo.
JZ	:Salto si cero.

Control de bucle:

JCXZ	:Salto si CX=0.
LOOP Byte con signo	:Bucle.
LOOPE Byte con signo	:Bucle si igual.
LOOPNE Byte con signo	:Bucle si no igual.
LOOPNZ Byte con signo	:Bucle si no cero.
LOOPZ Byte con signo	:Bucle si cero.

Interrupciones:

INT n ($0 \leq n \leq 255$)	:Interrupción.
INTO	:Interrupción por desbordamiento.
IRET	:Retorno de interrupción.

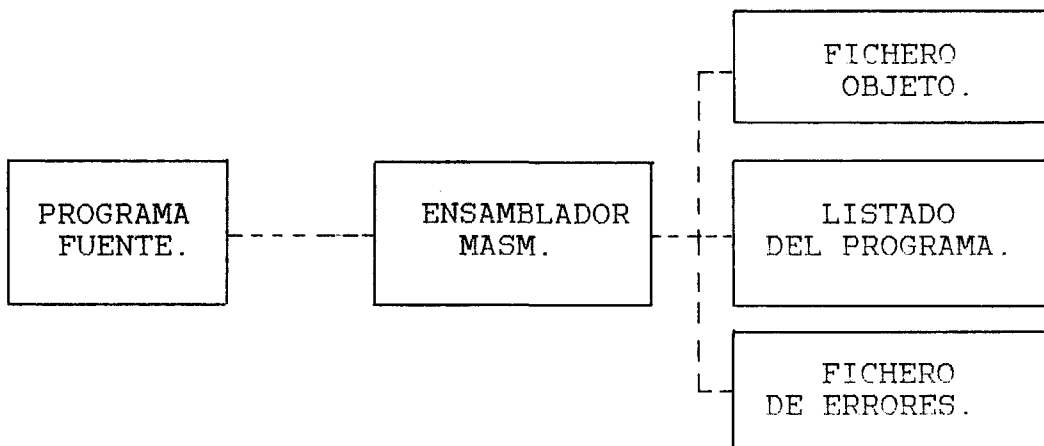
F) Introducción al macroensamblador para el microprocesador 8086 (MASM).

El ensamblador codifica instrucciones en código máquina y traduce un módulo fuente en un reposicionable código objeto. La entrada al ensamblador es el fichero fuente y la salida que produce consiste en tres ficheros posibles:

1-El fichero objeto que va a traducirse en código objeto por el LINK.

2-El fichero de listado que contiene un listado del código fuente, el código objeto generado por el ensamblador, mensajes de error y tabla de símbolos.

3-El fichero que contiene los mensajes de error y las líneas fuentes donde estos ocurren.



Veamos la secuencia de pasos a seguir para producir un código objeto. Suponiendo que es el fichero PROGRAMA el que queremos convertir en código objeto.

MASM PROGRAMA.ASM

Siendo MASM la llamada al macroensamblador y PROGRAMA.ASM el fichero que contiene el programa fuente (la extensión de este fichero ha de ser .ASM).

La respuesta del sistema es:

Objet filename [PROGRAMA.OBJ]:

Source listing [NUL.LIST]:

Cross reference [NUL.CRF]:

Los nombres que aparecen entre corchetes, es la opción por defecto, pudiendo colocar a continuación de los dos puntos el nombre del fichero que queremos que se cree.

Una vez obtenido el código objeto con el MASM, debemos crear el código ejecutable mediante LINK.

```
LINK PROGRAMA.OBJ
```

El sistema responde con:

Run file [PROGRAMA.EXE]:

List file [NUL.MAP]:

Libraries [.LIB]:

Los nombres que aparecen entre corchetes son los que el sistema asigna por defecto. El fichero PROGRAMA.EXE contiene el código listo para ejecutar. "List file", es un fichero que contiene un listado del programa e información de las variables y "Libraries" , son todas las librerías que podríamos unir a nuestro programa para obtener el código final.

F1)Directivos DB, DW, DD.

Estos directivos del ensamblador tienen dos funciones; definir variables e inicializar memoria.

DB define bytes.

DW define palabras (2 bytes).

DD define doble palabra (4 bytes).

Las variables se declaran con estos directivos y el nombre de estas se coloca a la izquierda. La dirección de cada variable queda definida pues, por el segmento dentro de la cual está definida su posición relativa dentro de este y el tipo de variable (DW, DB ó DD).

Veamos un ejemplo:

DATOS SEGMENT

```
Tabla1      DW 12 ; offset 0
             DW 34 ; offset 2
Número1     DB 5  ; offset 4
Pagos1      DW1314; offset 5
Pagos2      DD1718; offset 7
```

DATOS END.

Se puede reservar memoria sin tener que inicializarla o inicializarla a cualquier valor utilizando DUP.

```
DW 10 DUP (?);Reserva 10 palabra no inicializadas
DB13 5 DUP (1, 2, 4 DUP(3),2 DUP(10)); Se inicializan 50 bytes, 5 copias repetidas de 1, 2, 3, 3, 3, 3, 10, 10
```

```
COMENT1 DB 'BUENOS DIAS'
```

```
COMENT2 DB 128 DUP(' '); Inicializa 128 bytes a
```

blanco.

F2) Directivos SEGMENT/ENDS.

Nos permite crear segmentos, dando un nombre a cada segmento. Todas las instrucciones y las variables han de estar contenidos en ellos.

```
NOMBRE      SEGMENT      [ PARAMETRO ]
  .
  .
  .

NOMBRE      ENDS
```

El parámetro es opcional y a continuación comentamos algunos de ellos.

PUBLIC -Indica que este segmento está concatenado con otros que tengan el mismo nombre que se encuentren al hacer el link.

COMMON -Causa que este módulo comparta las mismas posiciones de memoria que otros segmentos con el mismo nombre, lo que implica que una misma posición de memoria se pueda referenciar con distintos nombres.

AT -Indica la dirección donde se va a asignar el segmento. Si no se coloca, el ensamblador establece la dirección.

ORG -Establece la primera línea donde se van a situar las instrucciones dentro del segmento.

F3)Directivo ASSUME.

Este directivo es obligatorio en todo programa y con él se le indica al ensamblador con qué segmentos se está trabajando. Ejemplo:

```
ASSUME:    DS : DATOS1
           ES : DATOS2
           SS : STACK1
           CS : CODIGO1
```

Donde DS, ES, SS y CS son los registros de segmentos y a su derecha tenemos el nombre de un segmento definido dentro del programa. Además del ASSUME dentro del segmento de código, tenemos que cargar los registros DS, ES y SS con las direcciones de comienzo de los segmentos. Para el registro CS no hace falta porque se encarga el propio ensamblador de hacerlo.

Si colocamos pues el directivo:

```
ASSUME CS:CODIGO1, DS:DATOS1, ES:DATOS2, SS:STACK1
```

dentro del segmento de código deberá aparecer:

```
CODIGO1 SEGMENT

START : MOV AX, DATOS1
       MOV DS, AX
       MOV AX, DATOS2
       MOV ES, AX
       MOV SS, STACK1
       MOV SS, AX
       :
```

Esta redundancia en cargar los registros DS, ES y SS es utilizada por el ensamblador para chequeos.

F4)Etiquetas.

La definición de etiquetas nos permite asignar nombres a posiciones de memoria independientemente del contenido de la memoria. Para etiquetar una instrucción hemos de terminar el nombre con dos puntos (:).

```
COMIENZO : MOV AX, CX
```

F5)Directivo END

Este directivo indica el final del programa fuente. Tiene que aparecer una sólo vez y ser la última instrucción.

Su formato es:

```
      .           .  
      :           :  
      .           .  
END           ó           END ETIQUETA.
```

F6)Estructura de un programa.

Veamos un resumen de la estructura de un programa escrito en ensamblador. De los distintos casos que se pueden dar, presentamos el más usual en el que las dimensiones de los segmentos del código de datos y el del Stack son menores de 64k bytes.

-El segmento PROGRAMA, contendrá las instrucciones del programa.

-El segmento DATOS contendrá los datos que se van a manipular.

-El segmento CUADRO, también almacenará datos como segmento extra.

-El segmento STACK, contendrá direcciones de retorno, datos temporales, parámetros de procedimientos, etc.

En este programa sólo existirá un segmento de código, uno de datos, uno extra y uno de Stack. No obstante nuestro programa puede ser el resultado de haber linkado varios módulos. En este caso se suele colocar al segmento el atributo de PUBLIC, para unir todos los segmentos con el mismo nombre definidos en diferentes módulos, en un segmento final para ejecución.

-Declaración del segmento de datos.

```
DATOS      SEGMENT      PUBLIC
:
:
:
DATOS      ENDS
```

-Declaración del segmento extra.

```
CUADRO     SEGMENT     PUBLIC
:
:
:
CUADRO     ENDS
```

-Declaración del segmento Stack.

```
STACK      SEGMENT      STACK
           DW            N DUP(?)
TOP-STACK  LABEL        WORDS
STACK      ENDS
```

N es el número de palabras reservadas para el Stack. El atributo STACK hace que este segmento sea público.

-Declaración del segmento de código.

```
PROGRAMA   SEGMENT      PUBLIC
ASSUME     CS:PROGRAMA, DS:DATOS, ES:CUADRO,
           SS:STACK

Start:     MOV     AX, DATOS
           MOV     DS, AX
           MOV     AX, CUADRO
           MOV     ES, AX
           MOV     AX, STACK
           MOV     SS, AX
           MOV     SP, OFFSET TOP-STACK
           .
           .
           . instrucciones en ensamblador.
           .
PROGRAMA   ENDS

           END     Start.
```

INTERFACE PARALELO.

INTERFACE PARALELO.

El interface paralelo puede estar situado en la placa madre del ordenador o en un slot de expansión. Está diseñado especialmente para la conexión con impresoras con el interface paralelo CONTRONIC, aunque también puede ser utilizado para la conexión de cualquier periférico que se ajuste a sus características de entrada/salida.

Tiene 12 salidas (Bus de datos, Strobe, Printer Init, Auto Feed y Select In) que pueden ser manipuladas por instrucciones de entrada/ salida de la impresora o el periférico en cuestión. También el interface tiene 5 entradas (Acknowledge, Busy, Paper End, Error y Select) que se pueden leer por instrucciones de entrada de la impresora o del periférico.

La entrada 'Acknowledge' se puede utilizar para generar una solicitud de interrupción.

Todas las señales de entrada/salida son a nivel TTL estandar:

- Líneas de datos: Cada línea de salidas de datos puede dar -2.6 mA y absorber 24 mA. Estas líneas no deben conectarse directamente a tierra.

- Líneas de control: Todas las líneas de salida de control están dispuestas en colector abierto, y cada una es capaz de absorber aproximadamente 7 mA.

- Líneas de estado: Cada una tiene un diodo Schottky

de baja potencia.

A) Conector del Interface.

Para la conexión de las señales con el exterior, el interface paralelo dispone de un conector de 25 pines D-Shell.

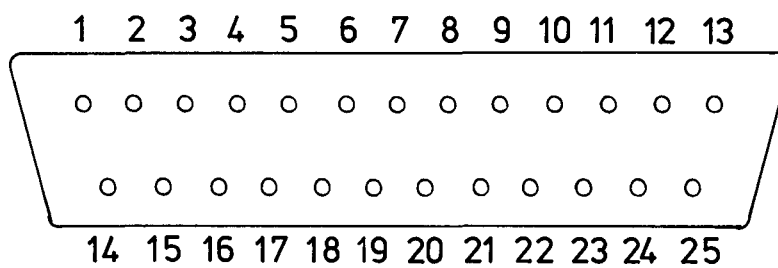
A continuación exponemos una descripción de las señales disponibles en éste conector.

<u>SEÑAL</u>	<u>I/O</u>	<u>DESCRIPCION</u>
PTD0 a PTD7	I/O	- Líneas de datos: Se utiliza para llevar y traer datos.
<u>STROBE</u>	I/O	- Strobe: Un pulso de nivel bajo se usa para transmitir un dato del interface a la impresora.
<u>AUTOFDX</u>	I/O	- Auto Feed: Un nivel bajo indica a la impresora bajar una línea después de imprimir.
<u>INIT</u>	I/O	- Inicializa la impresora: Un pulso de nivel bajo resetea la impresora, limpiando el buffer e inicializando su estado.
<u>SLCTIN</u>	I/O	- Select Input: un nivel bajo habilita a la impresora para que acepte un nuevo dato.
<u>ERROR</u>	I	- Error: Un nivel bajo indica

que se ha producido un error en la impresora.

SLCT	I	- Select.
PE	I	- Paper End: Un nivel alto indica que en la impresora no hay papel.
<u>ACK</u>	I	- Acknowledge: Un nivel bajo indica que el dato en curso ha sido aceptado, y que está preparada para aceptar uno nuevo.
BUSY	I	- Busy: Un nivel alto indica que la impresora no puede aceptar un dato nuevo.

<u>PIN</u>	<u>SEÑAL</u>	<u>PIN</u>	<u>SEÑAL</u>
1	$\overline{\text{STROBE}}$	14	$\overline{\text{AUTOFDX}}$
2	PTD0	15	$\overline{\text{ERROR}}$
3	PTD1	16	$\overline{\text{INIT}}$
:	.	17	$\overline{\text{SLCTIN}}$
:	.		
9	PTD7	18	GND
10	$\overline{\text{ACK}}$	19	GND
11	BUSY	.	.
12	PE	.	.
13	SLCT	25	GND

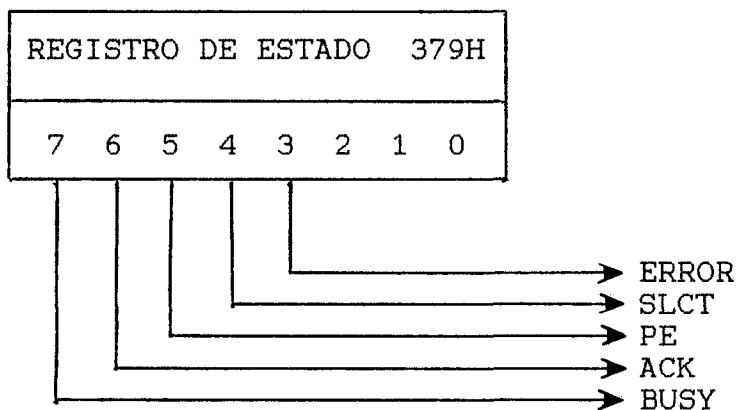
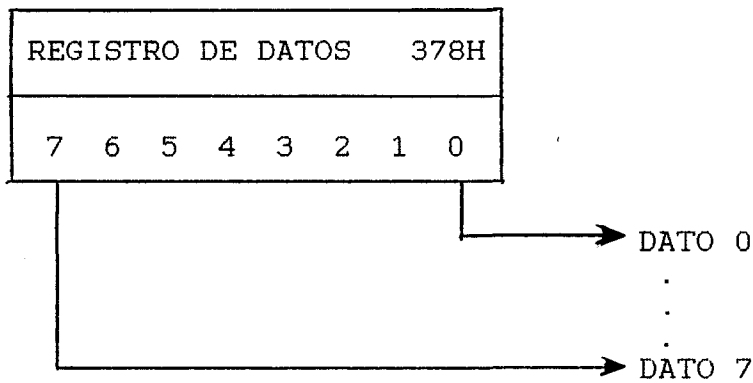


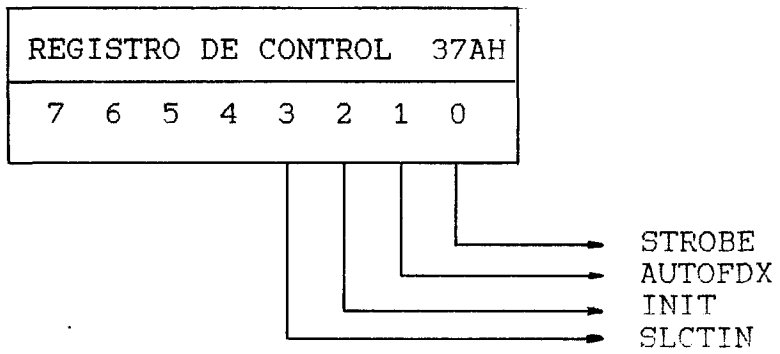
B) Programación de interface.

Para programar el interface paralelo, hemos de manejar tres registros: Uno de control, otro de salida/entrada de datos y otro registro de estado. Modificando los bits individuales de éstos registros, modificamos las salidas del interface.

Su localización en la memoria depende del puerto que tengamos configurado (LPT1 o LPT2).

	<u>LPT1</u>	<u>LPT2</u>
Entrada/Salida de datos	378H	278H
Estado	379H	279H
Control	37AH	37AH





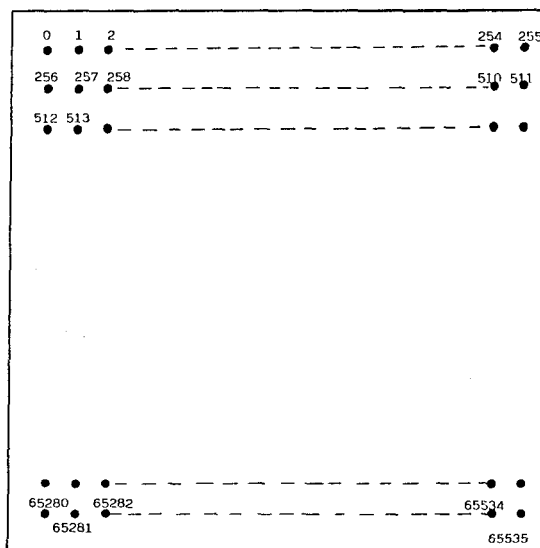
PROGRAMA.

PROGRAMA

Veremos a continuación el programa que controla al digitalizador y permite adquirir los datos de una imagen. Expondremos primero los módulos que lo componen, describiendo los organigramas, y los principios teóricos en que se basan, para pasar, por último, al programa completo.

A) Adquisición

Dándole un número consecutivo a cada pixel de la imagen digitalizada, el orden de llegada de información será en la primera columna, el dato correspondiente al pixel 0 , 256 , 512 ,65280 ; En la segunda columna, el pixel 1 , 257 , 513 ,65281 , y así hasta la última columna a la que llegarán los pixels 255 , 511 ,65535 .



Toda esta información la almacenaremos en memoria en forma ordenada, en lugares de memoria que correspondan a su orden. De esta manera, tendremos al completar la imagen en lugares consecutivos de memoria los pixels 0, 1, 2, 3,65534, 65535.

El organigrama del programa se muestra en la figura A:

1-En primer lugar reservamos dos variables que llamaremos COLUMNA y LINEA, que indicarán la columna y la línea con la que estamos trabajando. Inicializaremos estas variables teniendo en cuenta que la variable COLUMNA puede tomar los valores de 255 a 0, y la variable LINEA los valores 0 a 255.

2-Ponemos un 1 lógico en la línea reservada para el control del bus de datos del puerto paralelo, lo que indica que podemos enviar una dirección al digitalizador. Para este propósito, utilizaremos la línea SELECT IN del interface. Ponemos en el bus de datos la dirección que indica el registro COLUMNA y ponemos a nivel bajo (0 lógico) la línea de control del bus, habilitandolo para recibir información.

3-Testeando la línea reservada para transmitir los sincronismos de cuadro al ordenador (utilizaremos la línea PAPER END), esperamos a que aparezca uno. A partir de este momento, pasamos a testear la línea

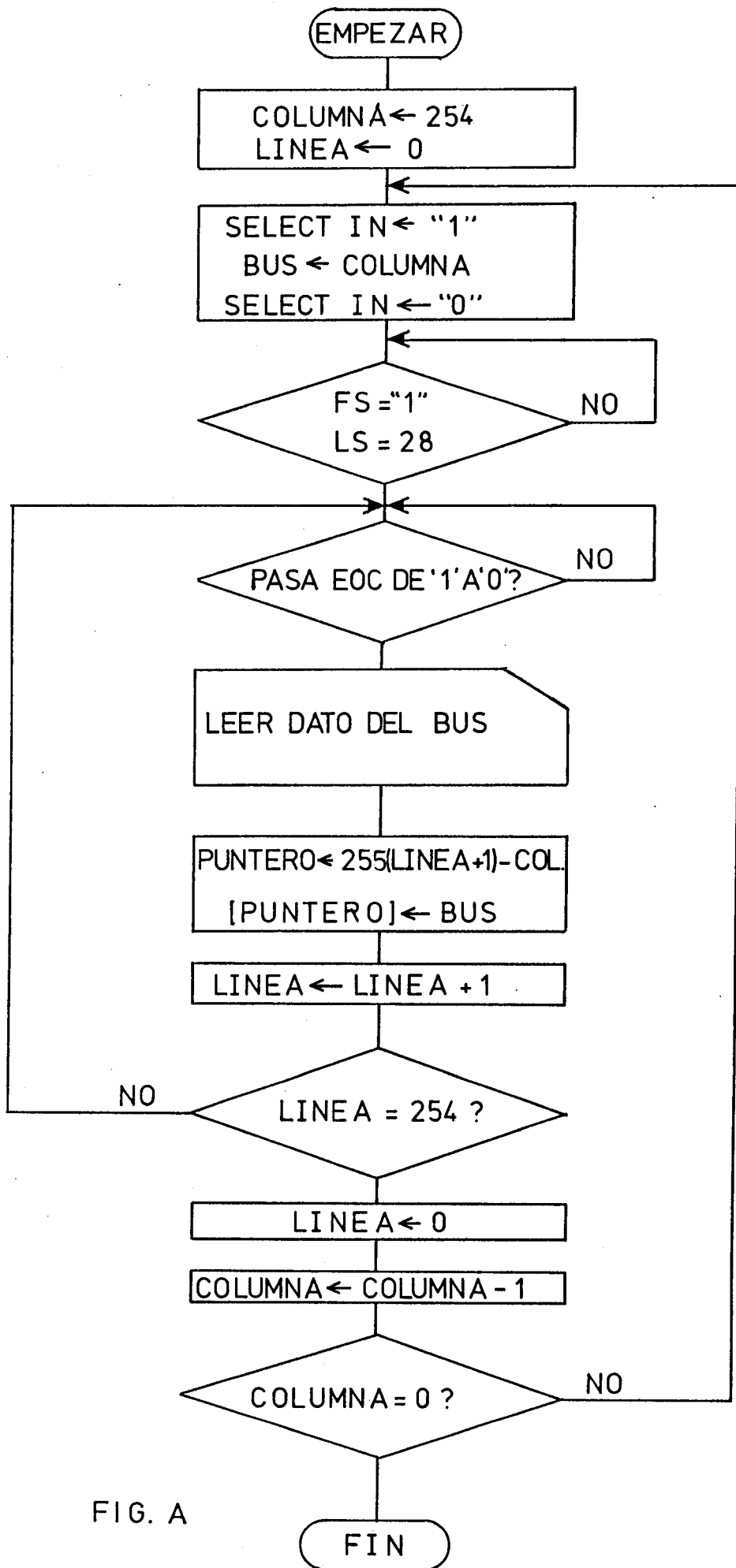


FIG. A

reservada para los sincronismos de línea, que será la denominada BUSY. Con un contador, esperamos que se presenten 28 sincronismos de línea, momento en el cual comenzamos la digitalización de la columna en curso. El motivo de esto es centrar la imagen digitalizada con respecto a la imagen fuente.

4-Testeamos la línea reservada para final de conversión (línea ACKNOWLEDGE del interface paralelo). Cada vez que en esta línea tengamos una transición de nivel alto (1 lógico) a nivel bajo (0 lógico), leemos el dato presente en el bus ya que se ha producido el final de conversión de un pixel. En el bloque de memoria reservado para almacenar la imagen, la posición de este dato leído viene dada por la siguiente ecuación:

$$255 \cdot (LINEA + 1) - COLUMNA$$

Este número sumado a la dirección de memoria donde comienza a estar almacenada la imagen, nos da la dirección donde hemos de guardar este dato.

5-Incrementamos el registro LINEA y lo comparamos con 254; Si LINEA es menor que 254 volvemos al punto 4. Si no, continuamos.

6-En este punto, LINEA es mayor que 254, por lo que para continuar con la siguiente columna hemos de hacer LINEA = 0, además de decrementar COLUMNA en una unidad.

7-Comparamos COLUMNA con 0; Si es mayor, transferimos el control al punto 2.

8-Si COLUMNA es 0 hemos concluido la digitalización de esta imagen y la tenemos almacenada en un bloque de memoria, a partir de una dirección determinada.

```

;*****
;*
;*          ADQUISICION DE DATOS DEL DIGITALIZADOR          *
;*
;*          ORDENADOR          *
;*
;*****
;
;
STACK      SEGMENT  STACK      ;Definimos una pila de 128 bytes.
           DW       128 DUP(?)
TOP        LABEL   WORD
STACK     ENDS
;
;
DATOS     SEGMENT  PUBLIC      ;Definimos un segmento de datos.
NLS       DW       0
LINEA    DW       0           ;Inicializamos la variable LINEA y COLUMNA
COLUMNA  DW       1 DUP(254); con 0 y 254 respectivamente.
COMENT1   DB       'DIGITALIZACION COMPLETA $'
DATOS     ENDS
;
;
PANTALLA SEGMENT  PUBLIC      ;Definimos un segmento extra de datos.
CUADRO   DB       65025 DUP(0)
PANTALLA ENDS
;
;
PROGRAMA SEGMENT  PUBLIC      ;Definimos el segmento código.
ASSUME   CS:PROGRAMA,DS:DATOS,ES:PANTALLA,SS:STACK
;
START:   MOV  AX,DATOS
         MOV  DS,AX
         MOV  AX,PANTALLA
         MOV  ES,AX
         MOV  AX,STACK
         MOV  SS,AX
         MOV  SP,OFFSET TOP
;
;
PRINCIPAL:MOV  DX,37AH          ;Cargamos en DX la direccion control.
           MOV  AL,4           ;Ponemos 0 en SELECT IN
           OUT  DX,AL
;
;
           MOV  DX,378H        ;Apuntamos a la direccion del
           ; registro de datos 378h.
           MOV  AX,COLUMNA     ;Pasamos al bus la columna
           OUT  DX,AL          ; en curso.
;
;
           MOV  DX,37AH        ;Apuntamos a la direccion del
           ; registro de control 37Ah.
           MOV  AL,12          ;Ponemos un 1 en SELECT IN.
           OUT  DX,AL
;
;

```

```

;
;
; ** DETECTAMOS QUE FS PASE DE ESTADO ALTO A BAJO. **
;
FS1:   MOV  DX,379H           ;Apuntamos al registro de
      IN   AL,DX             ; estado y testeamos el
      TEST AL,32             ; bit reservado para FS.
      JZ   FS1               ;Vuelve a testear si FS=0
FS2:   IN   AL,DX
      TEST AL,32
      JNZ  FS2

;
;
; ** DEJAMOS QUE PASEN LAS PRIMERAS 28 LINEAS. **
;
LS:    IN   AL,DX             ;LS lo detecta el ordenador al revés.
      TEST AL,128
      JNZ  LS
      INC  NLS
      CMP  NLS,28
      JLE  LS
      MOV  NLS,0

;
;
EOC1:  MOV  DX,379H
      IN   AL,DX             ;Detecta en el registro
      TEST AL,64             ; de estado cuando hay
      JNZ  EOC1              ; una transicion de

EOC2:  IN   AL,DX             ; 0 a 1 en el bit
      TEST AL,64             ; reservado para EOC
      JZ   EOC2

;
;
      MOV  DX,378H           ;Apuntamos al registro de
      IN   AL,DX             ; datos
      MOV  CL,AL             ;Guarda el dato en CL.

;
      MOV  BX,LINEA          ;Se efectua la operacion
      INC  BX                 ; 255(LINEA+1)-COLUMNNA.
      MOV  AX,255
      MUL  BX
      MOV  DX,COLUMNNA
      SUB  AX,DX              ;AX indica el puntero donde
                              ; almacenamos el dato.
      MOV  SI,AX
      MOV  [OFFSET CUADRO+SI],CL

;
;
      INC  LINEA              ;Incrementamos LINEA.

;
      CMP  LINEA,254          ;Compara 254 con LINEA.

```

```

JLE EOC1          ;Salta a EOC1 si 254>LINEA.
MOV LINEA,0       ;Pone a 0 LINEA.
;
DEC COLUMNNA      ;Decrementa COLUMNNA.

CMP COLUMNNA,0    ;Compara 0 con COLUMNNA.
JL CONTINUAR      ;Salta a PRINCIPAL si
JMP PRINCIPAL     ; 0<COLUMNNA.
;
CONTINUAR:MOV DX,OFFSET COMENT1
MOV AH,09H
INT 21H
;
;
MOV AH,00         ;Espera a que se presione una tecla
INT 16H          ; para continuar.
;
;
;
MOV AH,4CH       ;Devuelve el control al DOS.
INT 21H
;
;
PROGRAMA ENDS
END START

```

B) Adaptadores gráficos. Modos de video.

Para producir la imagen de vídeo, la mayoría de los PC's (incluyendo AT y XT) requieren un adaptador de vídeo, esto es, una tarjeta especial que normalmente se conecta en uno de los conectores de expansión del ordenador.

El adaptador de vídeo conecta el ordenador al monitor a través de un chip llamado Controlador de CRT. El adaptador también tiene un conjunto de puertos de E/S programables, una ROM generadora de caracteres y memoria RAM para almacenar la información del visualizador.

Hay varios tipos de adaptadores de vídeo, pero los que IBM originalmente adoptó para el PC son: el adaptador color/gráficos (CGA) y el adaptador monocromo. Los generadores de vídeo operan fundamentalmente según dos modos diferentes: modo texto y modo gráfico. El modo texto sólo permite visualizar caracteres, aunque algunos de estos caracteres se pueden ajustar para producir dibujos simples. El modo gráfico se utiliza principalmente para producir dibujos complejos, pero puede reproducir caracteres de texto.

El adaptador color/gráficos puede operar en ambos modos, texto y gráficos, para producir dibujos y caracteres en varios formatos y colores. Este está diseñado para trabajar con todo tipo de pantallas, desde televisores estándar a monitores de color de alta resolución.

El adaptador monocromo, por el contrario, sólo puede operar en modo texto utilizando un conjunto de caracteres ASCII, alfanuméricos y gráficos, que tiene guardados y visualizándolos en un sólo color.

Para superar estas limitaciones algunos fabricantes de hardware han ofrecido variaciones del adaptador monocromo, tales como el adaptador de vídeo Hércules. Este combina las capacidades gráficas (aunque no de color) del adaptador color/gráficos con las del visualizador de textos del adaptador monocromo, además de añadir características propias.

Otro adaptador, el 'Enhanced Graphics Adapter' (EGA) puede crear de forma similar gráficos en una pantalla monocroma.

Los adaptadores color/gráficos y monocromos almacenan la información de visualización mediante imagen mapeada en memoria. Se denomina de esta forma debido a que cada dirección de la memoria de pantalla corresponde a una localización específica de la misma. La circuitería del visualizador lee repetidamente la información de la memoria y la sitúa en la pantalla.

Los modos de pantalla definen las características del visualizador incluyendo la cantidad de texto que puede ser visualizado, la resolución o detalles de los gráficos y los colores.

El adaptador color/gráficos (CGA) permite varias opciones de formato diferentes en los modos de textos y gráficos. El adaptador monocromo ofrece un único formato de texto en un color. El Enhanced Graphic Adapter soporta una gran variedad de formatos.

Modo	Tipo	Tamaño	Colores	Adaptador	Visualización
0	Texto	40 × 25	16 (grises)	CGA, EGA, PCjr	Color mejorado
1	Texto	40 × 25	16 en primer plano, 8 de fondo	CGA, EGA, PCjr	Color mejorado
2	Texto	80 × 25	16 (grises)	CGA, EGA, PCjr	Color mejorado
3	Texto	80 × 25	16 en primer plano 8 de fondo	CGA, EGA, PCjr	Color mejorado
4	Gráficos	320 × 200	4	CGA, EGA, PCjr	Color mejorado
5	Gráficos	320 × 200	4 (grises)	CGA, EGA, PCjr	Color mejorado
6	Gráficos	640 × 200	2	CGA, EGA, PCjr	Color mejorado
7	Texto	80 × 25	b/n	EGA, A. monocr.	Monocromo
8	Gráficos	160 × 200	16	PCjr	Color mejorado
9	Gráficos	320 × 200	16	PCjr	Color mejorado
10	Gráficos	640 × 200	4	PCjr	Color mejorado
11	Aparentemente interno al EGA				
12	Aparentemente interno al EGA				
13	Gráficos	320 × 200	16	EGA	Color mejorado
14	Gráficos	640 × 200	16	EGA	Color mejorado
15	Gráficos	640 × 350	b/n	EGA	Monocromo
16	Gráficos	640 × 350	64	EGA	Color mejorado

El color puede ser usado en cualquier modo de visualización excepto en el previsto para el adaptador monocromo (modo 7). A través de los modos disponibles con el adaptador color/gráficos, podemos elegir de 2 a 16 combinaciones de color, incluyendo una selección de tonos grises, llamados modo de color suprimido. La resolución de la imagen está definida por el número de líneas, o líneas de barrido, de la parte superior a la inferior y por el número de pixels de izquierda a derecha de cada línea.

El control de modo de vídeo se puede hacer a través de una interrupción del software, ya que estos están controlados por la ROM-BIOS. La interrupción que controla los modos es la 16 (10 hex).

B1) Mapeado de los pixels.

Cuando se utiliza un modo gráfico, los pixels son almacenados como una serie de bits con una correlación uno a uno entre los bits de la memoria y los pixels de la pantalla.

El adaptador color/gráficos organiza la pantalla en 200 líneas, numeradas de 0 a 199. El número de pixels de cada línea depende del modo que se utilice. Las columnas de pixels para los modos gráficos de baja, media y alta resolución están numeradas de 0 a 159, 319 ó 639 respectivamente.

El almacenamiento de las filas está dividido en 'bancos' de líneas, que ocupan posiciones de memoria contiguas. Para los modos 4, 5, 6 y 8 hay dos bancos. El primero almacena la memoria para las líneas pares 0, 2, 4,198; y el segundo las impares. Los modos 9 y 10 tienen 4 bancos con las líneas alternadas:

banco1	0, 4, 8.....196
banco2	1, 5, 9.....197
banco3	2, 6, 10....198
banco4	3, 7, 11....199

Estos bancos de líneas son similares a las 'páginas' en modo texto.

La cantidad de memoria utilizada para soportar cada pixel varía con el modo. El modo 6 utiliza 1 bit

que puede seleccionar dos colores. Los modos 4, 5 y 10 utilizan dos bits seleccionando 1 entre cuatro colores, y los modos 8 y 9 usan cuatro bits que seleccionan 16 colores.

Excepto para el modo 10, que se trata especialmente, los bits necesarios para cada pixel en cada fila se cogen de la memoria en orden consecutivo. Por ejemplo en el modo 6, que utiliza un bit por pixel los 8 bits del primer byte de la memoria de pantalla controlan los primeros 8 pixels de la pantalla. El primer bit (es más significativo) controla el primer pixel y así sucesivamente.

En el modo 4 con dos bits por pixel los 8 bits de cada byte controlan 4 pixels. En el modo 8, cada byte controla 2 pixels. Debido a esta organización de la información gráfica en el buffer de vídeo, el dibujo de un pixel implica la modificación de bits individuales en la memoria. Los modos de dos, cuatro y dieciseis colores requieren que se modifiquen uno, dos o cuatro bits respectivamente para cambiar un sólo pixel. Estas operaciones pueden llegar a consumir una enorme cantidad de tiempo del microprocesador.

Otra forma de dibujar un pixel en la pantalla gráfica es mediante una interrupción de software, indicando la columna, la fila y el color. De este modo simplificamos la programación aunque no

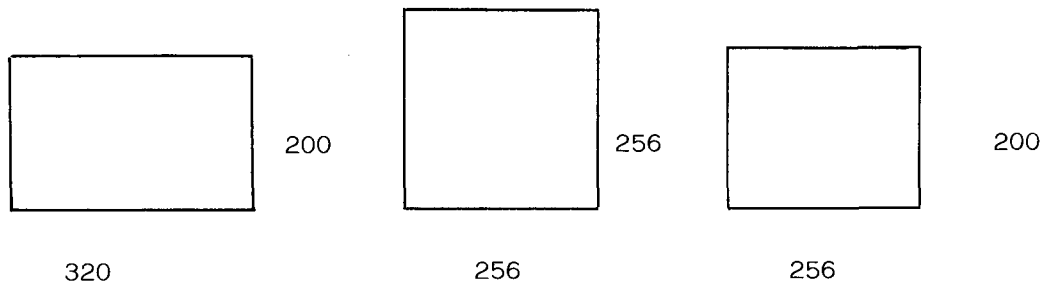
reducimos el tiempo empleado por el microprocesador.

C) Presentación en pantalla.

Para la presentación en pantalla de la imagen digitalizada almacenada en memoria, con una tarjeta gráfica CGA o EGA, tenemos que hacerlo pixel a pixel. Para ello elegiremos el modo de vídeo n° 5 o 6, los cuales tienen 320*200 pixels y disponen de 4 o 16 tonos de grises para cada pixel.

Como la información de luminancia almacenada de cada pixel de nuestra imagen es de 8 bits (1 byte), disponemos de 256 tonos de grises (2^8). Debido a las limitaciones de la tarjeta gráfica del PC sólo podemos representar 4 o 16 tonos de grises, es decir 2 o 4 bits, por lo que perdemos 6 bits de información. De los 8 bits de que disponemos, nos quedamos con los 2 o 4 bits más significativos.

Otro detalle es que la cuadrícula de la que disponemos consta de 320*200 pixels y nuestra imagen es de 256*256 pixels. Por tanto a la hora de presentar ésta imagen en un PC con tarjeta EGA o CGA no sólo perderemos información de luminancia sino que se nos recorta la imagen.



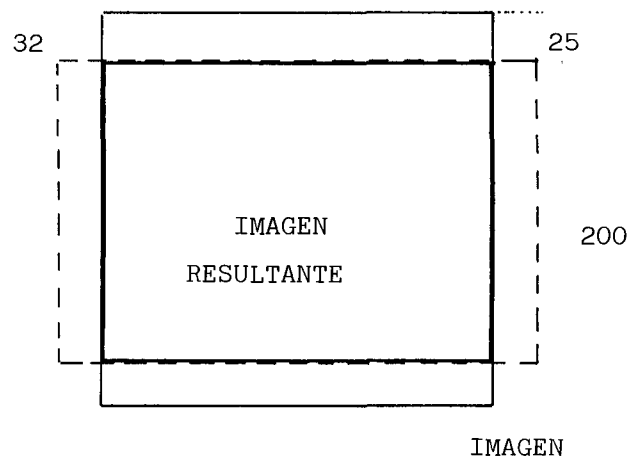
En resumen, veremos una cuadrícula de 256×200 donde cada pixel puede tomar 1 de 4 tonos de grises.

Para que la imagen quede centrada en la pantalla, en el eje horizontal empezaremos a escribir a partir del pixel situado en la posición $(320 - 256) / 2 = 32$.

El primer pixel a dibujar será el situado en la línea n° 25

$$255 * 25 = 5375$$

pixel n° 5375



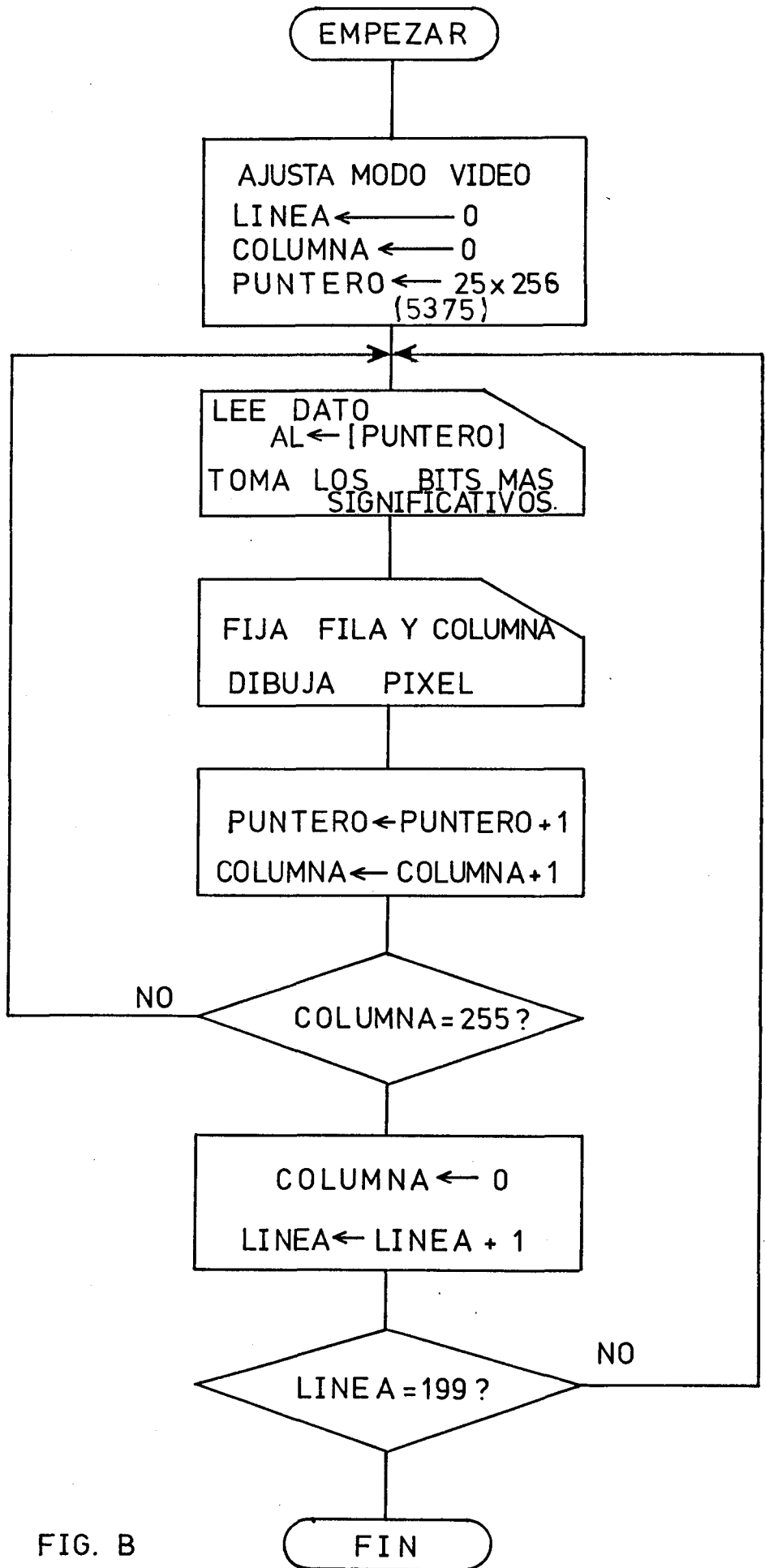


FIG. B

Visto esto podemos ver el organigrama (figura B).

1-Ajustamos el modo de vídeo . También inicializamos los registros LINEA y COLUMNA a 0. Inicializamos un nuevo registro PUNTERO que indicará la posición del pixel con el que estamos trabajando. Como ya hemos visto el valor inicial de este registro ha de ser 5375 por ser el primer pixel a dibujar.

2-Leemos el dato contenido en la dirección que indica el registro PUNTERO y nos quedamos con los 2 o 4 bits más significativos, según la tarjeta empleada.

3-Dibujamos el pixel con las coordenadas FILA, COLUMNA y el tono de gris que indique el dato obtenido en el punto 2.

4-Incrementamos el registro PUNTERO y COLUMNA.

5-Comparamos el registro COLUMNA con 255. Si es menor transferimos el control al punto 2 y si no seguimos adelante ya que una línea ha sido dibujada.

6-Ponemos a cero el registro COLUMNA e incrementamos el registro LINEA.

7-Comparamos el registro LINEA con 199. Si es menor volvemos al punto 2, y si no hemos concluido la presentación en pantalla de la imagen almacenada en la memoria.

```

;*****
;*
;*          PRESENTACION EN PANTALLA
;*
;*****
;
;
STACK      SEGMENT  STACK      ;Definimos una pila de 128 bytes.
           DW       128 DUP(?)
TOP        LABEL    WORD
STACK     ENDS
;
;
DATOS      SEGMENT  PUBLIC      ;Definimos un segmento de datos.
LINEA     DW       0           ;Inicializamos la variable LINEA y COLUMNA
COLUMNA   DW       1 DUP(254); con 0 y 254 respectivamente.
COMENT2   DB       'PRESENTACION CONCLUIDA $'
PUNTERO   DW       0
DATOS     ENDS
;
;
PANTALLA  SEGMENT  PUBLIC      ;Definimos un segmento extra de datos.
CUADRO    DB       65025 DUP(0)
PANTALLA  ENDS
;
;
PROGRAMA  SEGMENT  PUBLIC      ;Definimos el segmento código.
ASSUME    CS:PROGRAMA,DS:DATOS,ES:PANTALLA,SS:STACK
;
START:    MOV     AX,DATOS
           MOV     DS,AX
           MOV     AX,PANTALLA
           MOV     ES,AX
           MOV     AX,STACK
           MOV     SS,AX
           MOV     SP,OFFSET TOP
;
;
           MOV     AH,00        ;Servicio 0,modo de video.
           MOV     AL,05        ;Modo 5, graficos 320*200 4 grises
                               ;(CGA).
                               ;Modo 13, 16 grises (EGA).
           INT     16          ;Petición del servicio de video.
;
;
           MOV     LINEA,00     ;Inicializamos las variables LINEA,
           MOV     COLUMNA,31   ; COLUMNA, y preparamos el puntero
           MOV     PUNTERO,1    ; para extraer datos.
;
BUCLE:    MOV     SI,PUNTERO    ;Extrae el dato.
           MOV     AL,[OFFSET CUADRO+SI+6885]
           MOV     CX,4
           SHR     AL,CL        ;Dejamos los 2 bits mas significaticos

```

```

;para CGA o 4 para EGA.
MOV DX,LINEA
MOV CX,COLUMNA
MOV AH,12 ;Servicio 12,escritura de un pixel.
INT 16 ;Solicita servicio de video.
;
INC PUNTERO
INC COLUMNA
CMP COLUMNA,285 ;Compara COLUMNA con 255.
JLE BUCLE ;Salta a BUCLE si 255>=COLUMNA.
;
MOV COLUMNA,31 ;Reinicializa COLUMNA.
INC LINEA ;Incrementa LINEA.
CMP LINEA,199 ;Compara 199 con LINEA.
JLE BUCLE ;Salta a BUCLE si 199>=LINEA.
;
;
;
MOV AH,00
INT 16H
;
MOV AH,00 ;Servicio de video.
MOV AL,03
INT 16 ;Solicita servicio.
;
;
luis: MOV DX,OFFSET COMENT2
MOV AH,09H
INT 21H
;
;
MOV AH,4CH ;Devuelve el control al DOS.
INT 21H
;
;
PROGRAMA ENDS
END START

```

PRESUPUESTO.

PRESUPUESTO.

Evaluaremos a continuación el coste aproximado de los componentes necesarios para la realización del circuito.

UN.	COMPONENTE	PRECIO UNIDAD	TOTAL
20	Resistencia 1/4 w.	10	200
2	Resistencias ajustables	50	100
1	Condensador electr. 680 μ F 40V.	200	200
1	" " 4,7 μ F 63V.	20	20
2	" " 1 μ F 100V.	20	40
2	" " 10 μ F 50V.	20	40
1	" " 100 μ F 10V.	50	50
1	" " 470 μ F 16V.	50	50
1	Cond. Tántalo 4,7 μ F 16V.	50	50
2	" " 22 μ F 16V.	100	100
13	Condensador 100nF Ceram.	20	260
1	" 47nF "	15	15
1	" 680pF "	15	15
2	" 100pF "	15	30

UN.	COMPONENTE	PRECIO UNIDAD	TOTAL
1	Cond. Poliester Metal. 220nF	50	50
1	" " " 4,7nF	50	50
1	" " " 15nF	50	50
1	" " " 2,2nF		
4	Diodos 1N4148	15	60
1	Transistor PNP BC 559	50	50
5	Transistor NPN BC 548	50	250
1	Puente rectificador W01	150	150
1	Regulador de Tensión μ A 7805 UC	150	150
1	Circuito integrado 74HC132	125	125
1	" " 74HC14	100	100
2	" " 74HC161	125	125
1	" " 74HC373	125	125
1	" " 74HC00	100	100
1	" " 74LS244	125	125
1	" " 4013BE	100	100
1	" " 4071BE	100	100
1	" " 4066BE	100	100
1	" " 7660	1000	1000
1	Amplificador operacional LF 351	150	150
1	Convertidor A/D ZN 427	4500	4500
13	Zócalos para C.I.	30	390
1	Disipador de calor	150	150

UN.	COMPONENTE	PRECIO UNIDAD	TOTAL
1	Placa de circuito impreso fibra de vidrio doble cara.	1400	1400
1	Transformador 12V. 1A.	1000	1000
1	Conector.	150	150
1	Diodo Led.	50	50
1	Interruptor.	150	150
1	Caja metálica.	1700	1700
1	Conector Centrónico.	500	500
2	Conectores RCA.	75	150
1	Portafusibles.	150	150
	Cable plano.		200
		TOTAL:	14620 ptas.

APENDICES.

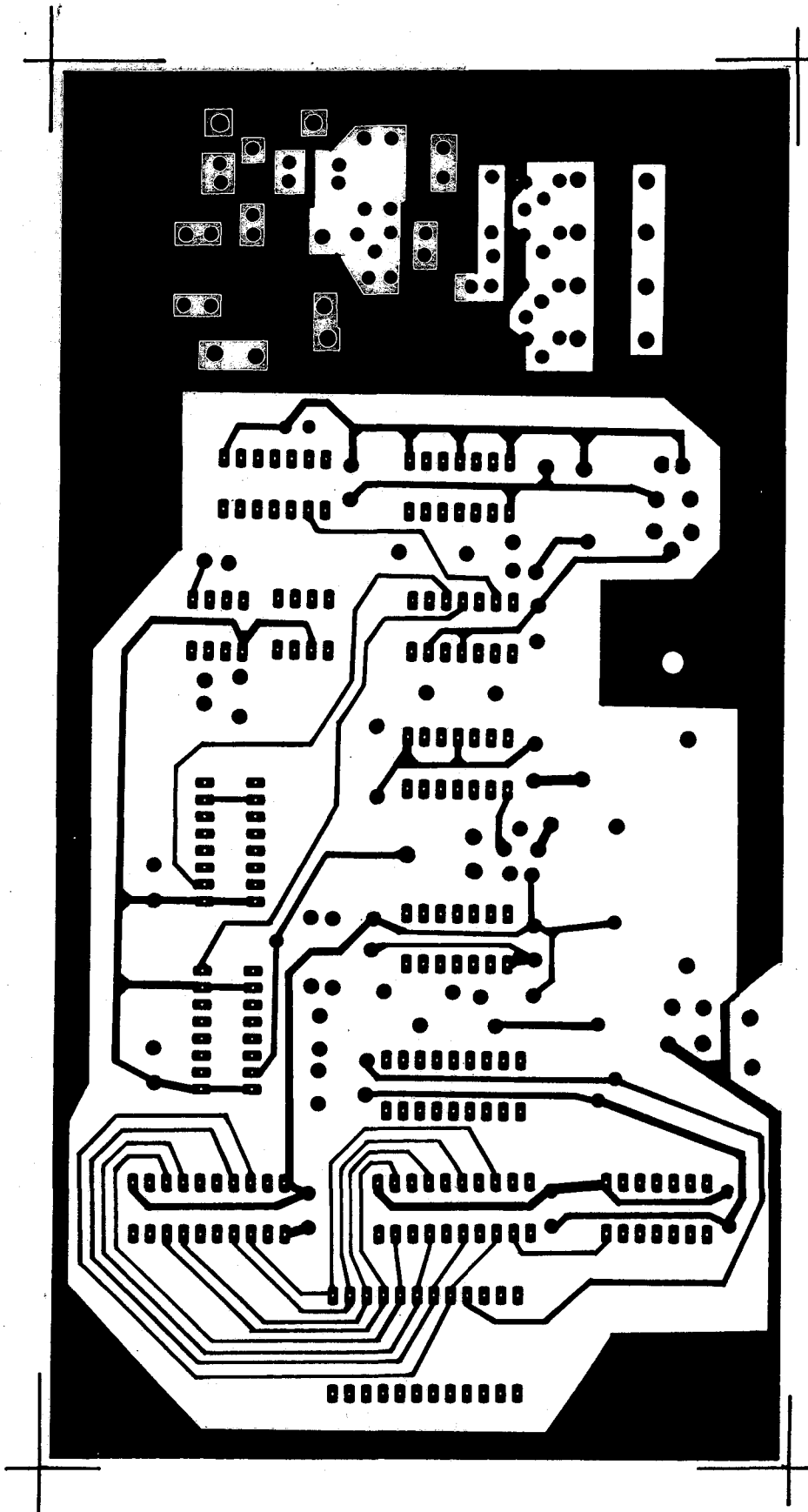
A1) REALIZACION DEL CIRCUITO.

REALIZACION DEL CIRCUITO ELECTRICO.

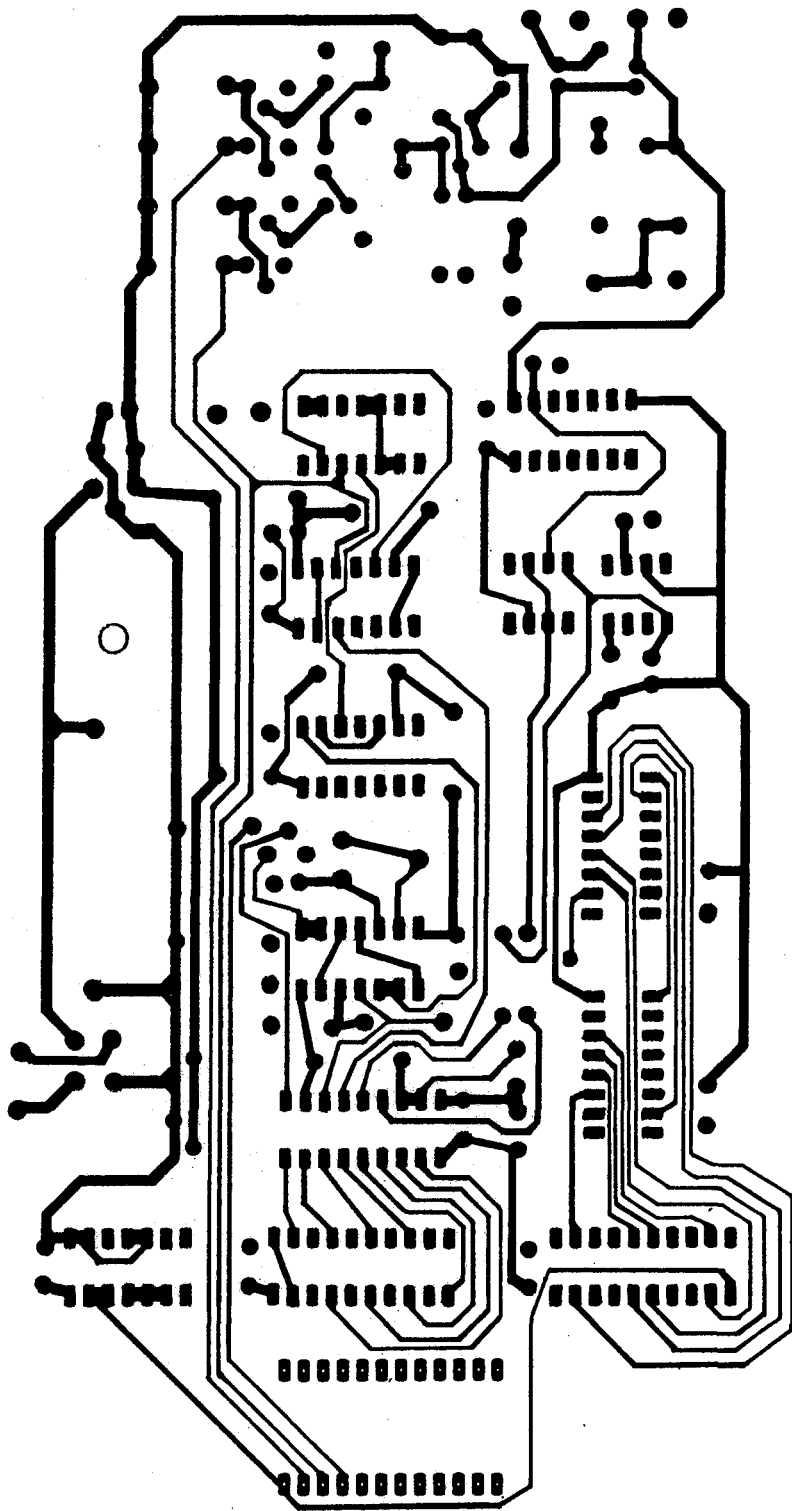
En éste apartado veremos primero las placa de circuito impreso y luego la relación de componentes utilizados.

A) Placa de Circuito Impreso.

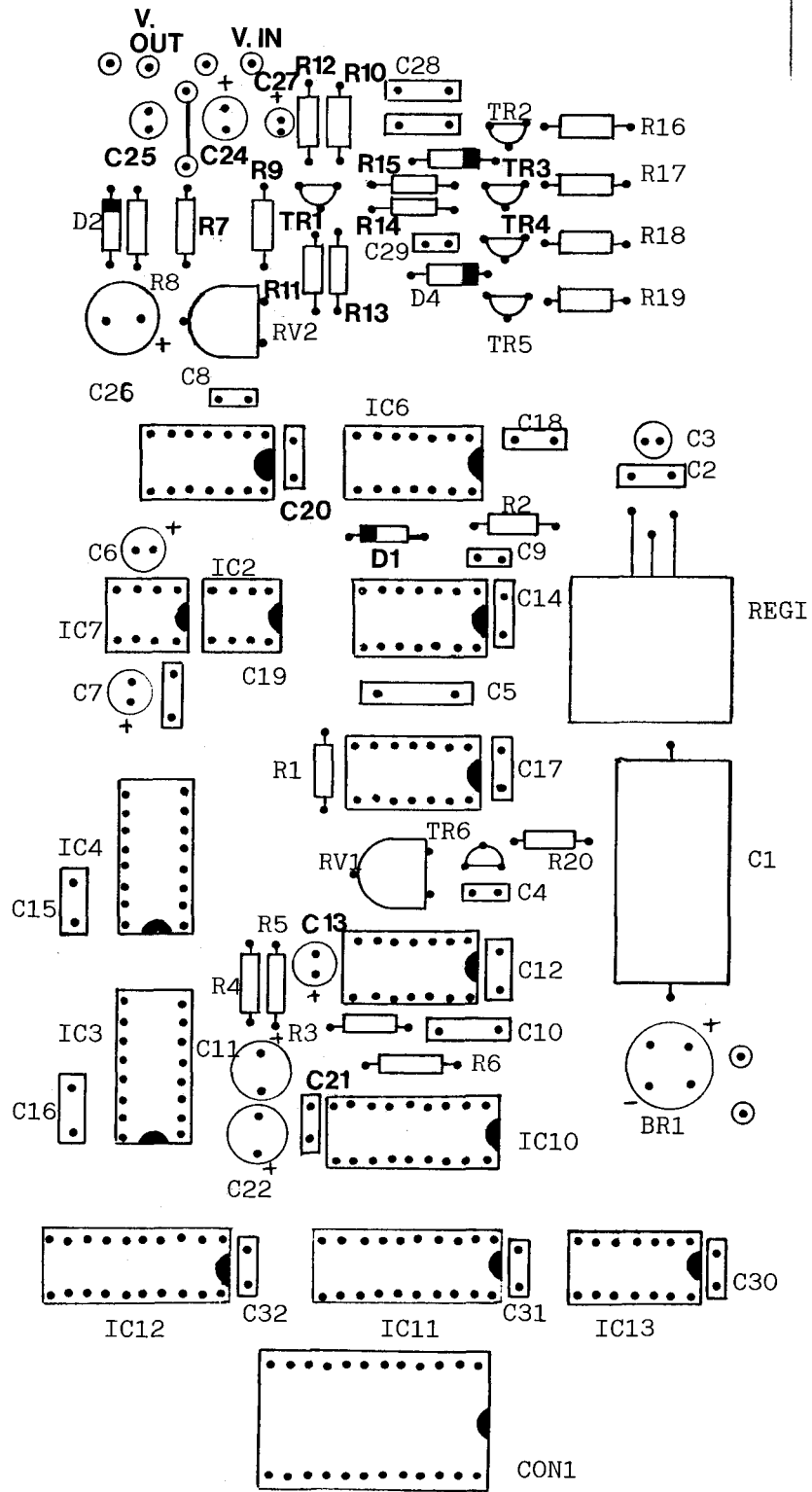
Emplearemos una placa de fibra de vidrio de doble cara. La cara A será la de colocación de los componentes.



LADO DE COLOCACION DE COMPONENTES
CARA A

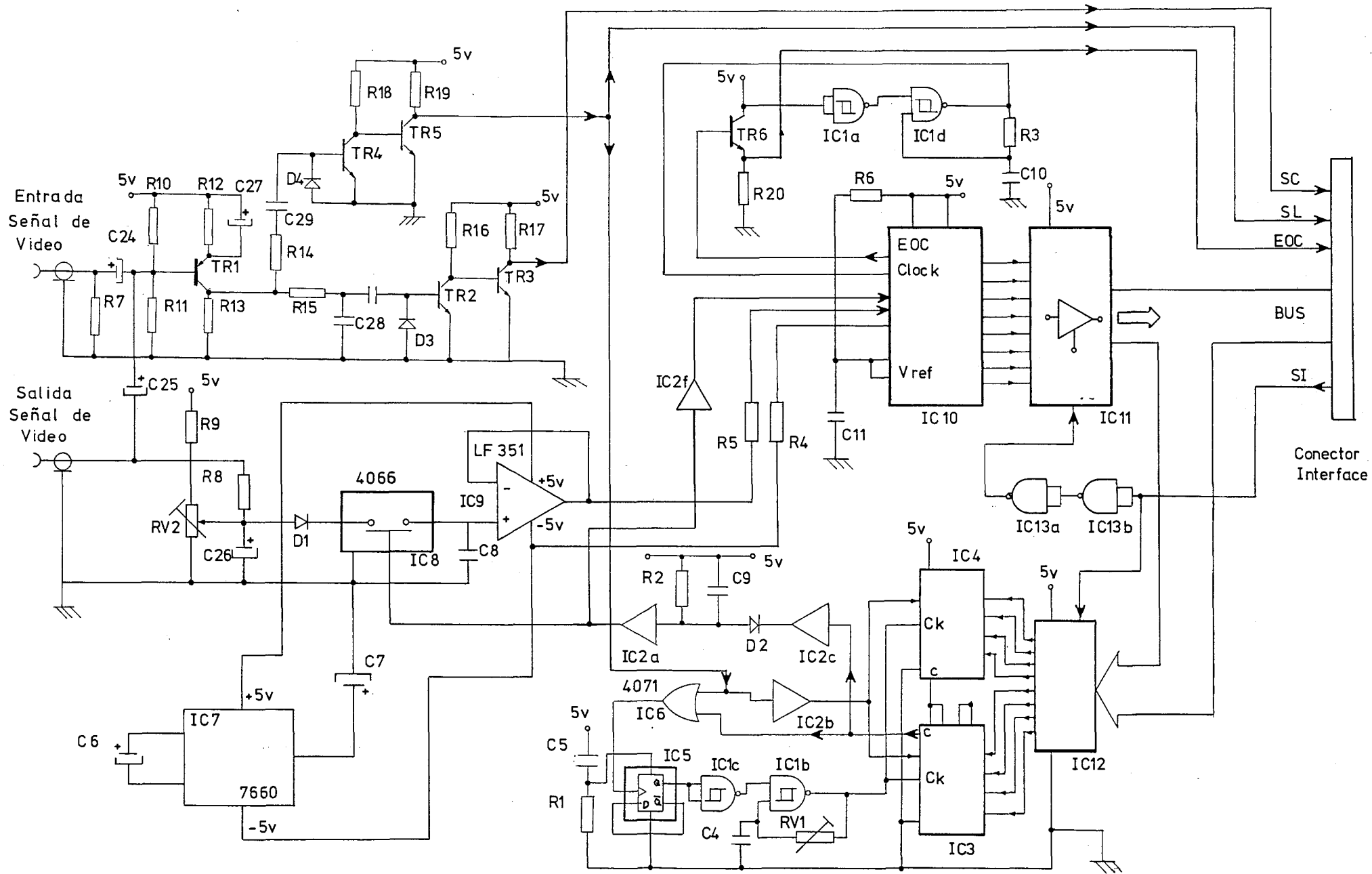


CARA B



COMPONENTES SOBRE LA PLACA DEL CIRCUITO IMPRESO

CARA A



B) Componentes.

B1) Resistencias.

R1	10K Ω
R2	10K Ω
R3	330 Ω
R4	82K Ω
R5	3K9 Ω
R6	390 Ω
R7	75 Ω
R8	100K Ω
R9	1K Ω
R10	8,2K Ω
R11	10K Ω
R12	18K Ω
R13	6,8K Ω
R14	10K Ω
R15	10K Ω
R16	2,2K Ω
R17	1,5K Ω
R18	2,2K Ω
R19	1,5K Ω
R20	1K Ω
RV1	470 Ω Ajustable.
RV2	1K Ω Ajustable.

B2) Condensadores.

C1	680 μ F	40V	Electrolítico.
C2	100nF		Cerámico disco.
C3	4,7 μ F	63V	Electrolítico.
C4	680pF		Cerámico.
C5	220nF		Poliéster Metalizado.
C6	1 μ F	100V	Electrolítico.
C7	10 μ F	50V	Electrolítico.
C8	100pF		Cerámico.
C9	100pF		"
C10	4.7nF		Poliéster Metalizado.
C11	4.7 μ F	16V	Tantalo.
C12	100nF		Cerámico disco.
C13	22 μ F	16V	Tantalo.
C14	100nF		Cerámico disco.

C15	100nF	Cerámico disco.
C16	100nF	" "
C17	100nF	" "
C18	100nF	" "
C19	100nF	" "
C20	100nF	" "
C21	100nF	" "
C22	22 μ F	16V Tantaló.
C23	15nF	Poliéster Metalizado.
C24	100 μ F	10V Electrolítico.
C25	1 μ F	100V Electrolítico.
C26	470 μ F	16V Electrolítico.
C27	10 μ F	50V Electrolítico.
C28	2.2nF	Poliéster Metalizado
C29	47nF	Cerámico disco.
C30	100nF	" "
C31	100nF	" "
C32	100nF	" "

B3) Semiconductores.

D1	1N4148
D2	1N4148
D3	1N4148
D4	1N4148
TR1	BC559
TR2	BC548
TR3	BC548
TR4	BC548
TR5	BC548
TR6	BC548
BR1	W01 1.5A. 100V.

B4) Circuitos Integrados.

REG1	μ A 7805 UC
IC1	74132
IC2	74HC14
IC3	74HC161
IC4	74HC161
IC5	4013BE
IC6	4071BE
IC7	7660

IC8	4066BE
IC9	LF351
IC10	ZN427
IC11	74LS244
IC12	74HC373
IC13	74HC06

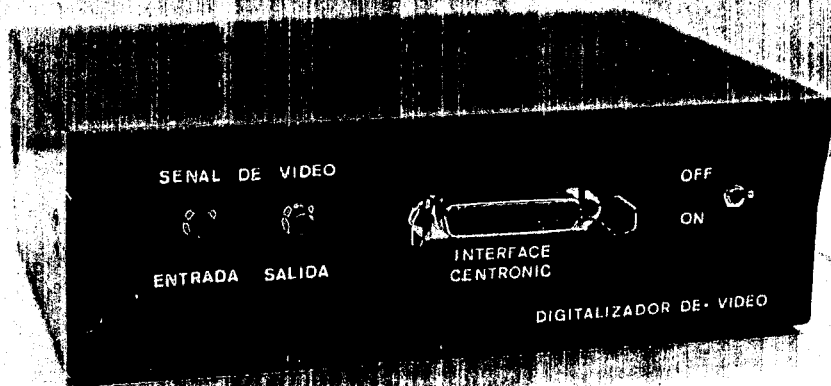
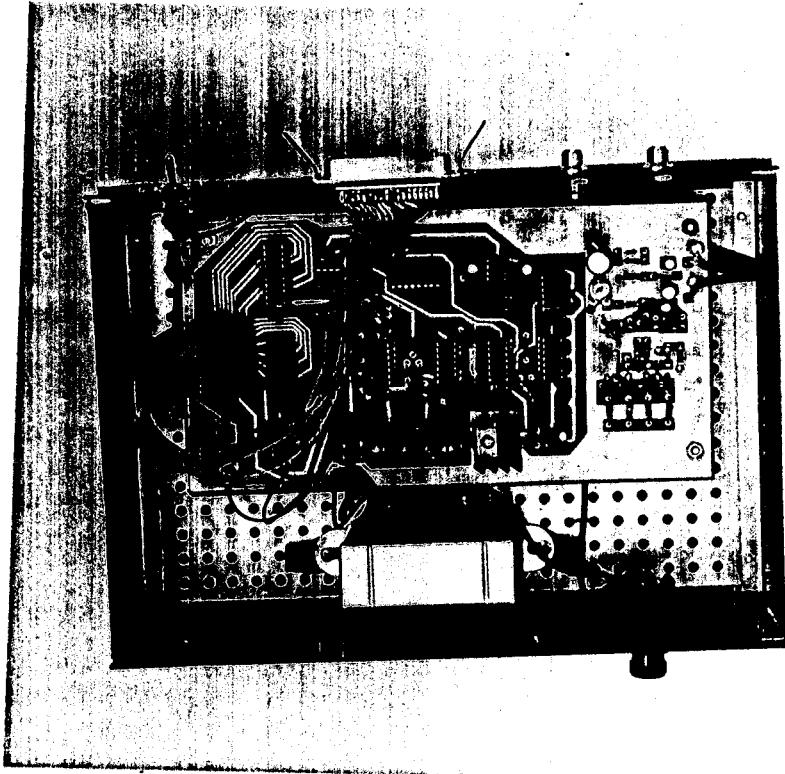
B5) Varios

Transformador 12V. 1A.

Disipador de calor.

CON1 Conector cable plano.

A2) VISTA DEL EQUIPO.



A3) PROGRAMA COMPLETO.

PROGRAMA COMPLETO.

Describamos en principio y a grandes rasgos el funcionamiento del programa completo que ha de controlar al digitalizador. La imagen como hemos visto queda almacenada en un bloque de memoria. Vamos a dotar al programa de la posibilidad de guardar ese bloque de memoria en un fichero con el fin de almacenar dicha imagen, por ejemplo en disketes. Con posterioridad podremos recuperar estos ficheros, almacenar estos bloques en memoria y volver a ver las imágenes.

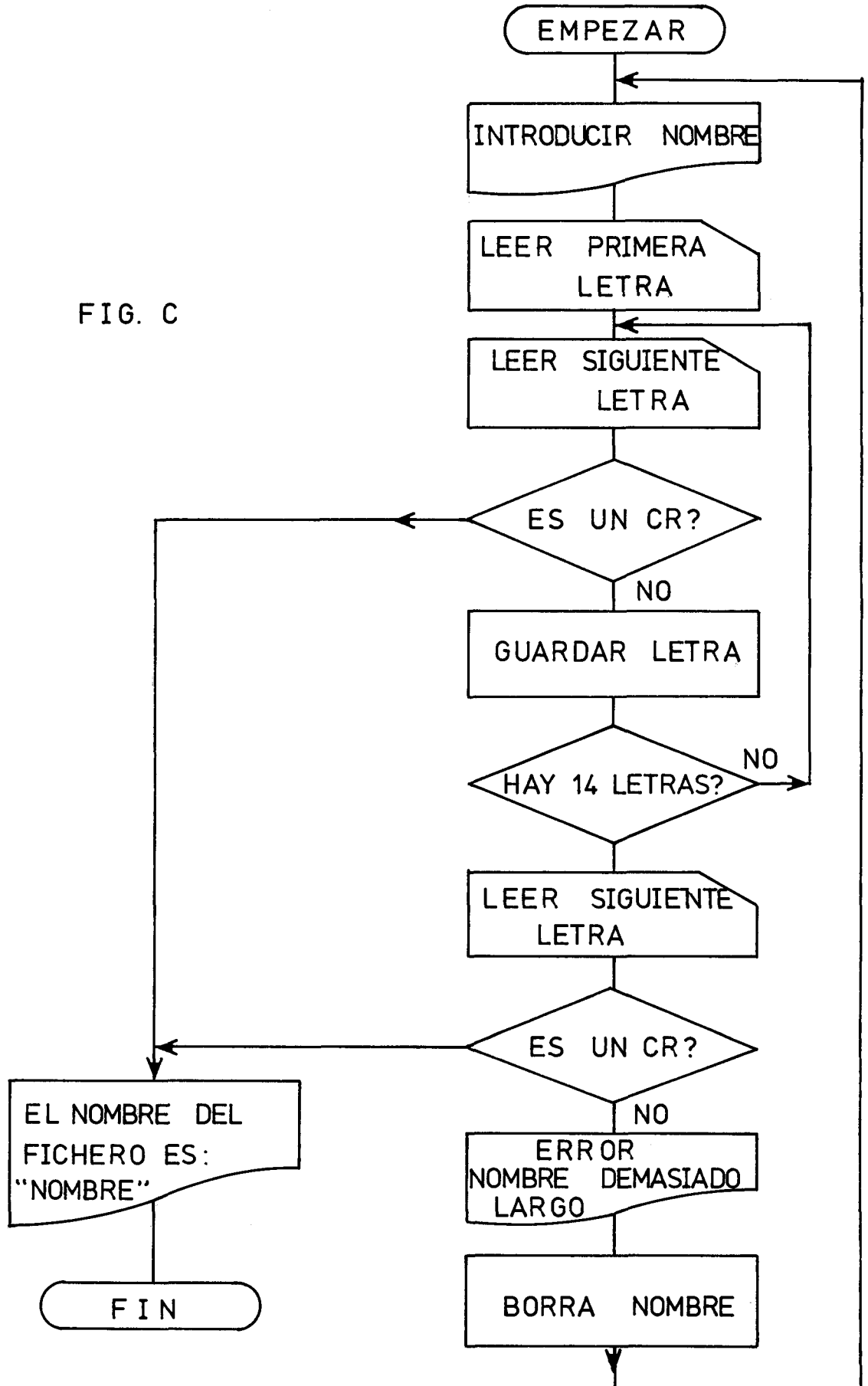
El hecho de guardar las imágenes en ficheros nos permite manejarlas con facilidad, ya sea para transmitir las a otro ordenador distante con un modem, o para almacenarlas para un posterior procesado.

Tenemos que ver pues, como se gestionan los ficheros desde el ensamblador, desde leer el nombre de éste, crearlo, hasta guardar o recuperar datos de él.

Para la obtención del nombre del fichero, emplearemos un programa como el que mostramos en el organigrama de la figura C.

Con este programa leemos un nombre para el fichero, como máximo de 14 caracteres (dos para la unidad donde se ubica, 8 para el nombre, uno para el

FIG. C



caracter "." y tres para la extensión: Ejemplo A:FICHERO1.EXT) desde el teclado y los almacenaremos en el registro NOMBRE.

Los programas escritos en lenguajes de alto nivel necesitan unicamente abrir los ficheros, con lo que se realiza todo el trabajo preparatorio para operaciones de fichero. Sin embargo en ensamblador se deben asignar zonas especiales de memoria para utilizarla en E/S de ficheros. El DOS utiliza dos métodos de acceso a ficheros; el llamado bloque de control de fichero (FCB) y el método Handle. Para nuestro propósito utilizaremos el método handle ya que el FCB sólo puede acceder a los ficheros del directorio actual, mientras el Handle puede buscar un fichero en cualquier lugar.

Para crear, leer o escribir en ficheros secuenciales emplearemos las distintas posibilidades que nos ofrece la interrupción de software 21H.

El organigrama del programa completo para gestionar el digitalizador y los ficheros, es el que mostramos en la figura D.

Empezamos imprimiendo en pantalla un pequeño menú con las opciones ofrecidas, que son:

- digitalizar una imagen
- ver una imagen almacenada en un fichero
- salir del programa.

Leemos el nombre del fichero con el que vamos a

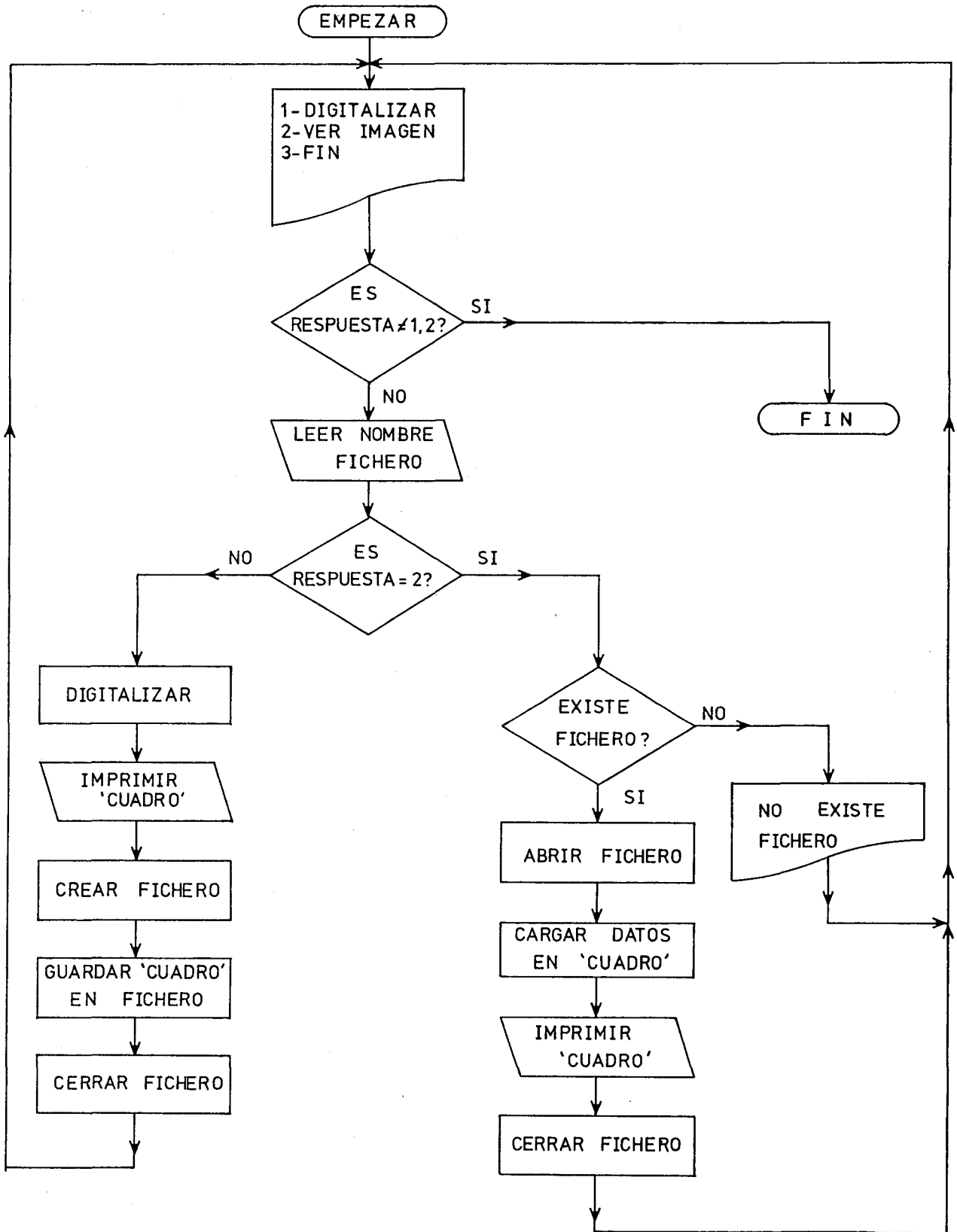


FIG. D

trabajar y seleccionamos el camino elegido. Si se opta por la digitalización, digitalizamos la imagen, obteniendola en un bloque de memoria determinada del ordenador. La sacamos en pantalla dibujando pixel a pixel los datos de este bloque de memoria y creamos un fichero con el nombre anteriormente leído, guardamos el bloque de memoria en dicho fichero, y al final, cerramos el fichero; todo esto ayudándonos de las interrupciones de software. A continuación volvemos al menú principal.

Si lo que hemos elegido es ver una imagen almacenada en algún fichero, en primer lugar vemos si el fichero existe. Si este no se encuentra, enviamos un mensaje y volvemos al menú principal. Por el contrario, si encontramos el fichero, lo abrimos, pasamos su información al bloque de memoria con el que vamos a trabajar e imprimimos la imagen por pantalla. Cerramos el fichero y volvemos al menú principal.

Siempre que efectuemos alguna operación, testeando algunos flags, podemos ver si en algún momento se produce un error o suceso anómalo. Si esto ocurriese, sacaríamos en pantalla un mensaje de error y transferiríamos el control al menú principal.

Vistas estas líneas generales pasemos a ver el programa completo en ensamblador:

```

;*****
;*****
;*****
;*
;*          PROGRAMA CONTROLADOR DEL DIGITALIZADOR
;*
;*
;*
;*
;*          Luis Guerrero Aroca 1990
;*
;*****
;*****
;*****
;
;
;
;
STACK      SEGMENT  STACK      ;Definimos una pila de 128 bytes.
           DW       128 DUP(?)
TOP
STACK      LABEL    WORD
           ENDS
;
;
DATOS      SEGMENT  PUBLIC      ;Definimos el segmento de datos.
LINEA     DW       0
COLUMNNA  DW       0
NLS       DW       0
PUNTERO   DW       0
HANDLE    DW       0
RESPUES   DB       0
NOMBRE    DB       14 DUP(' ')
           DB       '$'
;
COMENT1   DB       ' *          PULSAR OPCION DESEADA      * $'
COMENT2   DB       ' *          1 -DIGITALIZAR          * $'
COMENT3   DB       ' *          2 -VER IMAGEN           * $'
COMENT4   DB       ' *          3 -SALIR AL DOS          * $'
COMENT5   DB       'INTRODUCIR EL NOMBRE DEL FICHERO $'
COMENT6   DB       'NOMBRE DEMASIADO LARGO $'
COMENT7   DB       'NO EXISTE FICHERO $'
COMENT8   DB       'ERROR AL ABRIR FICHERO $'
COMENT9   DB       'ERROR EN PUNTERO DE FICHERO $'
COMENT10  DB       'ERROR AL LEER FICHERO $'
COMENT11  DB       'ERROR AL CERRAR FICHERO $'
COMENT12  DB       'PRESENTACION CONCLUIDA $'
COMENT13  DB       'DIGITALIZACION COMPLETA $'
COMENT14  DB       'ERROR AL CREAR FICHERO $'
COMENT15  DB       'ERROR AL ESCRIBIR EN FICHERO $'
COMENT17  DB       ' *          * $'
COMENT16  DB       ' ***** $'
DATOS     ENDS
;
;
PANTALLA  SEGMENT  PUBLIC      ;Definimos un segmento extra para datos.
CUADRO    DB       65025 DUP(0)

```

```

PANTALLA ENDS
;
;
PROGRAMA SEGMENT PUBLIC ;Definimos el segmento codigo.
ASSUME CS:PROGRAMA,DS:DATOS,ES:PANTALLA,SS:STACK
;
START: MOV AX,DATOS
MOV DS,AX
MOV AX,PANTALLA
MOV ES,AX
MOV AX,STACK
MOV SS,AX
MOV SP,OFFSET TOP

;
;
;*****
;***** PRESENTACION *****
;*****
;
EMPEZAR: CALL LIMPIADO
CALL CRLF
MOV DX,OFFSET COMENT16
CALL LISTAR
CALL CRLF
MOV DX,OFFSET COMENT1
CALL LISTAR
CALL CRLF
MOV DX,OFFSET COMENT17
CALL LISTAR
CALL CRLF
MOV DX,OFFSET COMENT2
CALL LISTAR
CALL CRLF
MOV DX,OFFSET COMENT3
CALL LISTAR
CALL CRLF
MOV DX,OFFSET COMENT4
CALL LISTAR
CALL CRLF
MOV DX,OFFSET COMENT16
CALL LISTAR
CALL CRLF
CALL CRLF
CALL CRLF

;
CALL LEER_C ;Espera contestación.
MOV RESPUES,AL ;Guardamos la contestación en RESPUES.
CMP AL,'1'
JZ SEGUIR1
CMP AL,'2'
JZ SEGUIR1
MOV AH,4CH ;Se ha pulsado una tecla distinta de 1,2
INT 21H ; y devolvemos el control al DOS.

;
SEGUIR1: CALL LIMPIADO ;Limpiamos la pantalla.

```

```

;
;
;*****
;*****  LECTURA DEL NOMBRE DEL FICHERO  *****
;*****
;
        JMP  LIMPIA
;
;
COMIENZO: MOV  DX,OFFSET COMENT5;DX apunta al primer comentario.
          CALL LISTAR          ;Listamos el mensaje por la pantalla.
          CALL CRLF
          CALL LEER_C          ;Leemos un caracter del teclado.
;
          MOV  BX,OFFSET NOMBRE;Cargamos la primera letra.
          MOV  [BX],AL
;
          MOV  CX,1            ;CX es el contador de los caracteres
;                                ; leidos.
VOLVER_D: CALL LEER_C
          CALL CR              ;Vemos si se introduce un CR.
          MOV  BX,OFFSET NOMBRE
          ADD  BX,CX
          MOV  [BX],AL
          INC  CX
          CMP  CX,14
          JNZ  VOLVER_D
;
          CALL LEER_C          ;Tiene que ser un CR, si no, se ha
;                                ; producido un error.
          CALL CR
          CALL CRLF
          MOV  DX,OFFSET COMENT6
          CALL LISTAR
          CALL CRLF
;
LIMPIA:  MOV  CX,0
          MOV  AL,' '          ;Inicializamos los caracteres del
ININOM:  MOV  BX,OFFSET NOMBRE; nombre a blanco.
          ADD  BX,CX
          MOV  [BX],AL
          INC  CX
          CMP  CX,14
          JNZ  ININOM
          JMP  COMIENZO
;
SEGUIR2: CALL CRLF
          MOV  AL,RESPUES
          CMP  AL,'1'
          JNZ  SEGUIR11       ;Salta a DIGIT1 si se pulsó 1.
          JMP  DIGIT1
;
;
;
;

```

```

;*****
;***** VER IMAGEN *****
;*****
;
;
SEGUIR11: MOV AH,4EH ;Función comprobar si hay algún fichero
          LEA DX,NOMBRE ; con el mismo nombre.
          INT 21H
          JNC SEGUIR3
          MOV DX,OFFSET COMENT7
          CALL LISTAR
          CALL ERROR ;No existe fichero.
          JMP EMPEZAR

;
SEGUIR3: MOV AH,3DH ;Función abrir fichero.
          MOV AL,02
          LEA DX,NOMBRE
          INT 21H
          JNC SEGUIR4
          MOV DX,OFFSET COMENT8
          CALL LISTAR ;Error al abrir fichero.
          CALL ERROR
          JMP EMPEZAR

;
SEGUIR4: MOV HANDLE,AX
          MOV AH,42 ;Situa el puntero del fichero al
          MOV AL,02 ; principio.
          MOV CX,0
          MOV DX,0
          INT 21H
          JNC SEGUIR5
          MOV DX,OFFSET COMENT9
          CALL LISTAR ;Error de puntero del fichero.
          CALL ERROR
          JMP EMPEZAR

;
SEGUIR5: MOV AH,3FH ;Función leer fichero.
          MOV BX,HANDLE
          MOV CX,65025
          LEA DX,CUADRO
          INT 21H
          JNC SEGUIR6
          MOV DX,OFFSET COMENT10
          CALL LISTAR ;Error de lectura.
          CALL ERROR
          JMP EMPEZAR

;
SEGUIR6: MOV BX,HANDLE ;Función cerrar fichero.
          MOV AH,3EH
          INT 21H
          JNC SEGUIR7
          MOV DX,OFFSET COMENT11
          CALL LISTAR ;Error al cerrar fichero.
          CALL ERROR
          JMP EMPEZAR

```

```

;
SEGUIR7: CALL MONITOR      ;** IMPRIMIR EN PANTALLA **
;
        JMP  EMPEZAR
;
;
;*****
;*****      DIGITALIZAR      *****
;*****
;
DIGIT1:  MOV  COLUMNA,254
        MOV  LINEA,0
;
;
DIGIT:   MOV  DX,27AH      ;Cargamos en DX la dirección control.
        MOV  AL,4         ;Ponemos 1 en SELECT IN.
        OUT  DX,AL
;
        MOV  DX,378H      ;Apuntamos a la dirección del
                        ; registro de datos 378h.
        MOV  AX,COLUMNA   ;Pasamos al bus la columna
        OUT  DX,AL        ; en curso.
;
        MOV  DX,37AH      ;Apuntamos a la dirección del
                        ; registro de control 37Ah.
        MOV  AL,12        ;Ponemos un 0 en SELECT IN.
        OUT  DX,AL
;
;
;
;** DETECTAMOS QUE FS PASE DE ESTADO ALTO A BAJO. **
;
FS1:     MOV  DX,379H      ;Apuntamos al registro de
        IN   AL,DX        ; estado y testeamos el
        TEST AL,32        ; bit reservado para FS.
        JZ   FS1          ;Vuelve a testear si FS=0
FS2:     IN   AL,DX
        TEST AL,32
        JNZ FS2
;
;
;
;** DEJAMOS QUE PASEN LAS 28 PRIMERAS LINEAS. **
;
LS:      IN   AL,DX
        TEST AL,128
        JNZ LS            ;LS lo detecta el ordenador al revés.
        INC  NLS
        CMP  NLS,28
        JLE LS
        MOV  NLS,0
;
EOC1:    MOV  DX,379H
        IN   AL,DX        ;Detecta en el registro
        TEST AL,64        ; de estado cuando hay
        JNZ  EOC1        ; una transición de

```

```

EOC2:   IN   AL,DX           ; 1 a 0 en el bit
        TEST AL,64         ; reservado para EOC
        JZ   EOC2
;
;
        MOV  DX,378H       ;Apuntamos al registro de
        IN   AL,DX         ; datos y leemos el dato.
        MOV  CL,AL        ;Guarda el dato en CL.
;
        MOV  BX,LINEA     ;Se efectua la operación
        INC  BX           ; 255(LINEA+1)-COLUMNA.
        MOV  AX,255
        MUL  BX
        MOV  DX,COLUMNA
        SUB  AX,DX        ;AX indica el puntero donde
                        ; almacenamos el dato.
        MOV  SI,AX
        MOV  [OFFSET CUADRO+SI],CL
;
;
        INC  LINEA        ;Incrementamos LINEA.
;
        CMP  LINEA,254    ;Compara 254 con LINEA.
        JLE  EOC1        ;Salta a EOC1 si 254>=LINEA.
        MOV  LINEA,0      ;Pone a 0 LINEA.
;
        DEC  COLUMNA
;
        CMP  COLUMNA,0    ;Compara 0 con COLUMNA.
        JL   CONTINUAR   ;** ** Salta a DIGIT si 0<=COLUMNA.
        JMP  DIGIT
;
CONTINUAR:CALL CRLF
        MOV  DX,OFFSET COMENT13
        CALL LISTAR
        CALL CRLF
;
;
        MOV  AH,00        ;Espera a que se presione una tecla
        INT  16H         ; para continuar.
;
        CALL MONITOR     ;**** IMPRIMIR EN PANTALLA ****
;
        MOV  AH,3CH      ;Función crear fichero.
        MOV  CX,0        ;Atributo normal.
        LEA  DX,NOMBRE
        INT  21H
        JNC  SEGUIR8
        MOV  DX,OFFSET COMENT14
        CALL LISTAR     ;Error al crear fichero.
        CALL ERROR
        JMP  EMPEZAR
;
SEGUIR8: MOV  HANDLE,AX

```



```

MOV BX,AX
MOV AH,40H ;Función escribir en fichero.
MOV CX,65025
LEA DX,CUADRO
INT 21H
JNC SEGUIR9
;
ERROR_DIS:MOV DX,OFFSET COMENT15
CALL LISTAR ;Error al grabar en fichero.
CALL ERROR
JMP EMPEZAR
;
SEGUIR9: CMP CX,65025
JNE ERROR_DIS
MOV AH,3EH ;Función cerrar fichero.
MOV BX,HANDLE
INT 21H
JC SEGUIR10
JMP EMPEZAR
;
SEGUIR10: MOV DX,OFFSET COMENT11
CALL LISTAR ;Error al cerrar fichero.
CALL ERROR
JMP EMPEZAR
;
;
;
;*****
;***** FIN DEL PROGRAMA *****
;*****
;
;
;*****
;***** PROCEDIMIENTOS *****
;*****
;
;
;**** LEE CARACTER DEL TECLADO ****
LEER_C: MOV AH,01
INT 21H
RET
;
;
;**** ESCRIBE CARACTERES EN PANTALLA ****
LISTAR: MOV AH,09
INT 21H
RET
;
;
;**** DETECTA UN CARRY RETURN ****
CR: CMP AL,0DH
JNZ SEGUIR12

```

```

                JMP  SEGUIR2
SEGUIR12: RET
;
;
;****  PRODUCE UN CARRY RETURN Y DESPLAZA UNA LINEA  ****
CRLF:  MOV  AH,02
        MOV  DL,0DH
        INT  21H
        MOV  DL,0AH
        INT  21H
        RET
;
;
;****  PARA MENSAJES DE ERROR  ****
ERROR:  MOV  AH,02
        MOV  DL,0DH
        INT  21H
        MOV  DL,0AH
        INT  21H
        MOV  AH,01
        INT  21H          ;Espera que se presione una tecla.
        RET
;
;
;****  LIMPIADO DE PANTALLA  ****
LIMPIADO: MOV  AH,00
        MOV  AL,03
        INT  16
        RET
;
;
;****  PRESENTACION EN PANTALLA  ****
MONITOR: MOV  AH,00          ;Servicio 0,modo de video.
        MOV  AL,05          ;Modo 5, graficos 320*200 4 grises.
                                ;Modo 13, 16 grises.
        INT  16          ;Petición del servicio de video.
;
;
        MOV  LINEA,00        ;Inicializamos las variables LINEA,
        MOV  COLUMNA,31      ; COLUMNA, y preparamos el puntero
        MOV  PUNTERO,1       ; para extraer datos.
;
BUCLE:  MOV  SI,PUNTERO      ;Extrae el dato.
        MOV  AL,[OFFSET CUADRO+SI+6885]
        MOV  CX,6
        SHR  AL,CL          ;Dejamos los 2 bits mas significaticos
                                ; si es CGA o 4 si es EGA.
        MOV  DX,LINEA
        MOV  CX,COLUMNA
        MOV  AH,12          ;Servicio 12,escritura de un pixel.
        INT  16          ;Solicita servicio de video.
;
        INC  PUNTERO
        INC  COLUMNA
        CMP  COLUMNA,285    ;Compara COLUMNA con 255.

```

```

        JLE  BUCLE          ;Salta a BUCLE si 255>=COLUMNNA.
;
MOV  COLUMNNA,31          ;Reinicializa COLUMNNA.
INC  LINEA                ;Incrementa LINEA.
CMP  LINEA,199           ;Compara 199 con LINEA.
JLE  BUCLE                ;Salta a BUCLE si 199>=LINEA.
;
;
;
MOV  AH,00
INT  16H
;
MOV  AH,00                ;Servicio de video.
MOV  AL,03
INT  16                    ;Solicita servicio.
;
;
MOV  DX,OFFSET COMENT12
CALL LISTAR
CALL CRLF
;
MOV  AH,00
INT  16H
;
RET
;
;
PROGRAMA  ENDS
          END  START

```

A4) CARACTERISTICAS DE CIRCUITOS EMPLEADOS.

LF 351

National Semiconductor

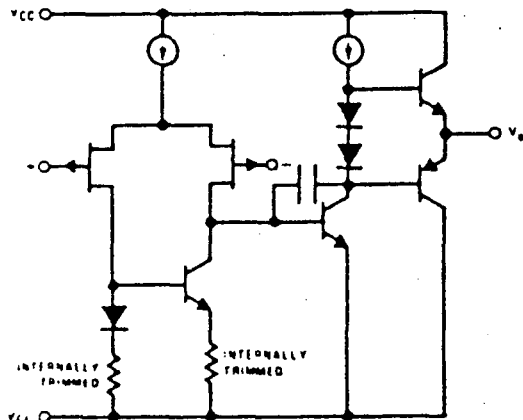
Amplificadores operacionales de banda ancha con entrada JFET

Descripción general

El LF 351 es un amplificador operacional de bajo coste y gran velocidad, con entrada a JFET y tensión de offset ajustada internamente (tecnología BI-FET II). El dispositivo requiere una pequeña corriente de alimentación, pese a lo cual mantiene su gran ancho de banda y una buena velocidad de respuesta. Además los experimentados transistores de entrada JFET de alta tensión acarrearán una baja corriente de polarización de entrada, así como un reducido offset. El LF 351 es compatible pin a pin con el popular 741, y utiliza el mismo circuito de ajuste de la tensión de offset. Esto permite la inmediata sustitución por el LF 351 en cualquier circuito diseñado para el 741.

El LF 351 puede ser empleado en integradores de alta velocidad, convertidores D/A rápidos, circuitos de muestreo y retención y otras aplicaciones que requieran baja tensión de offset, baja corriente de polarización, elevada impedancia de entrada, elevada velocidad de respuesta y gran ancho de banda. El dispositivo presenta bajo ruido y pequeña deriva de la tensión de offset, pero en aquellas aplicaciones en las que estas características sean críticas puede emplearse el LF 356. Si la corriente de alimentación debe ser reducida es preferible, sin embargo, el LF 351.

Esquema simplificado



Valores máximos absolutos

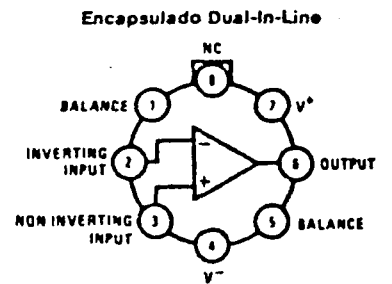
Tensión de alimentación	± 18 V
Potencia disipada (nota 1)	500 mW
Margen de temperatura de empleo	0°C a +70°C
T _j (MAX)	115°C
Tensión de entrada diferencial	± 30 V
Márgen de tensión de entrada (nota 2)	± 15 V
Duración del cortocircuito a la salida	Continuo
Margen de temperaturas de almacenaje	-65°C a +150°C
Temperatura durante la soldadura (10 segundos)	300°C

Características:

- Tensión de offset autoajustada 2 mV
- Baja corriente de polarización de entrada 50 pA
- Baja tensión de ruido de entrada 16 nV/√Hz
- Baja corriente de ruido de entrada 0.01 pA/√Hz
- Gran ancho de banda 4 MHz
- Gran velocidad de respuesta 13 V/μs
- Baja corriente de alimentación 1,8 mA
- Alta impedancia de entrada 10¹² Ω
- Reducida distorsión armónica total A_v = 10
R_L = 10 k, V_O = 20 V_{p-p}.
BW = 20 Hz-20 kHz 0.02%
- Baja frecuencia de corte de ruido 50 Hz
- Corto tiempo de subida al 0,01% 2 μs

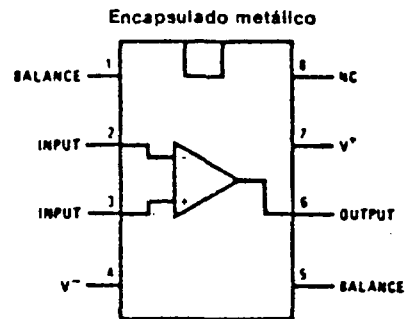


Diagramas de conexión



TOP VIEW

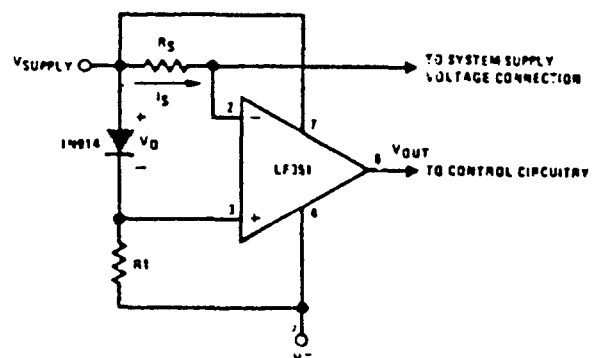
Nota. Pin 4 conecta a la caja



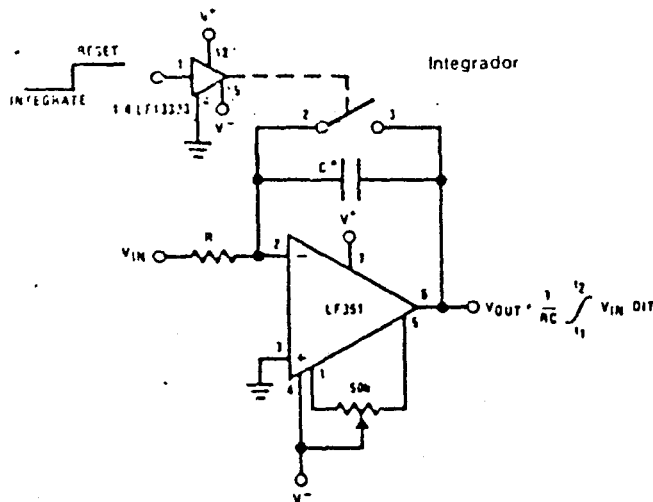
TOP VIEW

Aplicaciones típicas

Indicador/Limitador de corriente de alimentación



- Los conmutadores Vout a nivel alto cuando R_SI_S > V_D



- * Condensador de pequeña capacidad
- * El potenciómetro de 50k, se utiliza para el ajuste grueso de VOS

Características eléctricas en DC (Nota 3)

SYMBOL	PARAMETER	CONDITIONS	LF351A			LF351B			LF351			UNITS
			MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
V _{OS}	Input Offset Voltage	R _S = 10 kΩ, T _A = 25°C Over Temperature		1	2		3	5		5	10	mV
ΔV _{OS} /ΔT	Average TC of Input Offset Voltage	R _S = 10 kΩ		10			10			10		μV/°C
I _{OS}	Input Offset Current	T _J = 25°C (Notes 3, 4) T _J ≤ 70°C		25	50		25	100		25	100	μA
I _B	Input Bias Current	T _J = 25°C (Notes 3, 4) T _J ≤ 70°C		50	100		50	200		50	200	μA
R _{IN}	Input Resistance	T _J = 25°C		10 ¹²			10 ¹²			10 ¹²		Ω
A _{VOL}	Large Signal Voltage Gain	V _S = ±15V, T _A = 25°C V _O = ±10V, R _L = 2 kΩ Over Temperature	50	100		50	100		25	100		V/mV
V _O	Output Voltage Swing	V _S = ±15V, R _L = 10 kΩ	-12	+13.5		-12	+13.5		-12	+13.5		V
V _{CM}	Input Common Mode Voltage Range	V _S = ±15V	-11	+15		-11	+15		-11	+15		V
CMRR	Common Mode Rejection Ratio	R _S ≤ 10 kΩ	80	100		80	100		70	100		dB
PSRR	Supply Voltage Rejection Ratio	(Note 5)	80	100		80	100		73	100		dB
I _S	Supply Current		1.9	2.8		1.9	2.8		1.9	3.4		mA

Características eléctricas en AC (Nota 3)

SYMBOL	PARAMETER	CONDITIONS	LF351A			LF351B			LF351			UNITS
			MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
SR	Slew Rate	V _S = ±15V, T _A = 25°C		13			13			13		V/μs
GBW	Gain Bandwidth Product	V _S = ±15V, T _A = 25°C		4			4			4		MHz
e _n	Equivalent Input Noise Voltage	T _A = 25°C, R _S = 100Ω, f = 1000 Hz		18			16			16		nV/√Hz
i _n	Equivalent Input Noise Current	T _J = 25°C, f = 1000 Hz		0.01			0.01			0.01		pA/√Hz

Nota 1. Para trabajar a temperatura elevada, debe reducirse la resistencia térmica del dispositivo a 150° C/W entre unión y ambiente, o 45° C/W entre unión y cápsula.

Nota 2. Salvo especificación en contra, la máxima tensión negativa de entrada, es igual a la tensión de alimentación negativa.

Nota 3. Estas especificaciones han sido fijadas para V_S = ±15 V y 0°C ≤ T_A ≤ +70°C. V_{OS}, I_B e I_{OS}.

medidas a V_{CM} = 0

Nota 4. Las corrientes de polarización de entrada son las corrientes de carga de la unión, y su valor se incrementa aproximadamente al doble cada vez que la temperatura de la unión (T_J) aumenta 10°C. Debido al limitado tiempo de prueba de la producción, las corrientes de polarización de entrada medidas, se relacionan con la temperatura de la unión. En régimen normal, la temperatura de la unión se eleva por encima de la temperatura ambiente debido a la disipación de potencia interna, P_D. T_J = T_A + θ_{JA} P_D, donde θ_{JA} es la resistencia térmica desde la unión al ambiente. Se recomienda el uso de disipador si la corriente de polarización de entrada va a ser mantenido al mínimo.

Nota 5. La relación de rechazo de la tensión de alimentación, se mide para las dos alimentaciones, aumentando o disminuyendo simultáneamente según la forma de uso.

74 HC / HCT 14

HEX INVERTING SCHMITT TRIGGER

FEATURES

- Output capability: standard
- I_{CC} category: SSI

GENERAL DESCRIPTION

The 74HC/HCT14 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7.

The 74HC/HCT14 provide six inverting buffers with Schmitt-trigger action. They are capable of transforming slowly changing input signals into sharply defined, jitter-free output signals.

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t_{PHL}/t_{PLH}	propagation delay nA to nY	$C_L = 15 \text{ pF}$ $V_{CC} = 5 \text{ V}$	11	15	ns
C_I	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per gate	notes 1 and 2	7	8	pF

$GND = 0 \text{ V}$; $T_{amb} = 25 \text{ }^\circ\text{C}$; $t_r = t_f = 6 \text{ ns}$

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in μW):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f_i = input frequency in MHz

f_o = output frequency in MHz

$\sum (C_L \times V_{CC}^2 \times f_o)$ = sum of outputs

C_L = output load capacitance in pF

V_{CC} = supply voltage in V

2. For HC the condition is $V_I = GND$ to V_{CC}
For HCT the condition is $V_I = GND$ to $V_{CC} - 1.5 \text{ V}$

ORDERING INFORMATION/PACKAGE OUTLINES

PC74HC/HCT14P: 14-lead DIL; plastic (SOT-27).

PC74HC/HCT14T: 14-lead mini-pack; plastic (SO-14; SOT-108A).

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 3, 5, 9, 11, 13	1A to 6A	data inputs
2, 4, 6, 8, 10, 12	1Y to 6Y	data outputs
7	GND	ground (0 V)
14	V_{CC}	positive supply voltage

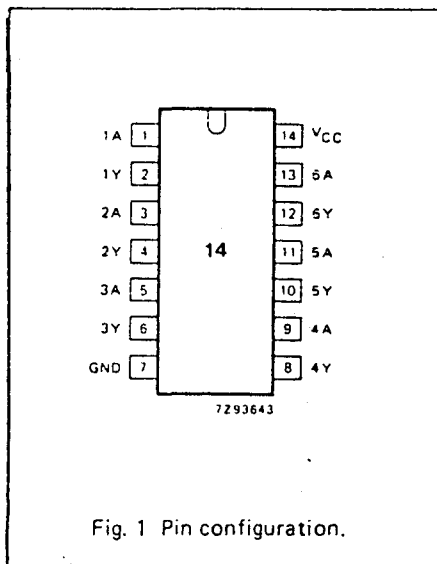


Fig. 1 Pin configuration.

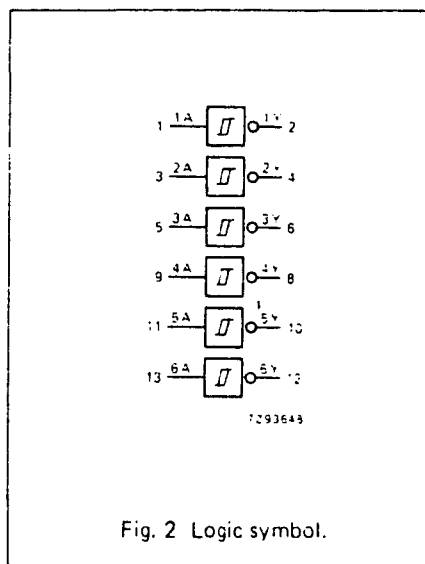


Fig. 2 Logic symbol.

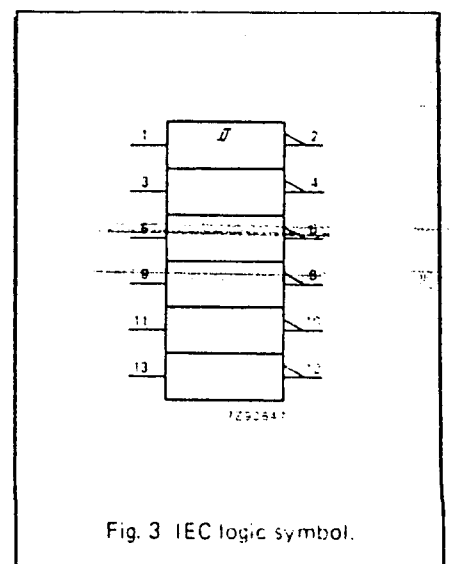


Fig. 3 IEC logic symbol.

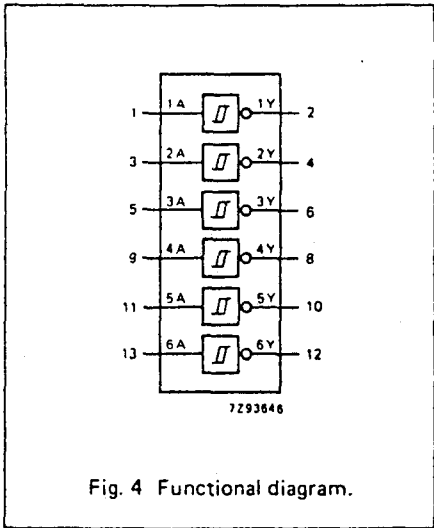


Fig. 4 Functional diagram.

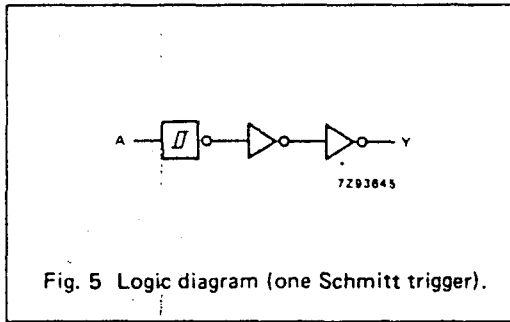


Fig. 5 Logic diagram (one Schmitt trigger).

FUNCTION TABLE

INPUT	OUTPUT
nA	nY
L	H
H	L

H = HIGH voltage level
L = LOW voltage level

APPLICATIONS

- Wave and pulse shapers
- Astable multivibrators
- Monostable multivibrators

DC CHARACTERISTICS FOR 74HC

For the DC characteristics see chapter "HCMOS family characteristics", section "Family specifications". Transfer characteristics are given below.

Output capability: standard
 I_{CC} category: SSI

Transfer characteristics for 74HC

Voltages are referenced to GND (ground = 0 V)

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HC									V _{CC} V	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
V _{T+}	positive-going threshold	0.7 1.7 2.1		1.5 3.15 4.2	0.7 1.7 2.1	1.5 3.15 4.2	0.7 1.7 2.1	1.5 3.15 4.2	V	2.0 4.5 6.0	Figs 6 and 7	
V _{T-}	negative-going threshold	0.3 0.9 1.2		1.0 2.2 3.0	0.3 0.9 1.2	1.0 2.2 3.0	0.3 0.9 1.2	1.0 2.2 3.0	V	2.0 4.5 6.0	Figs 6 and 7	
V _H	hysteresis (V _{T+} - V _{T-})	0.2 0.4 0.6		1.0 1.4 1.6	0.2 0.4 0.6	1.0 1.4 1.6	0.2 0.4 0.6	1.0 1.4 1.6	V	2.0 4.5 6.0	Figs 6 and 7	

AC CHARACTERISTICS FOR 74HC

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HC									V _{CC} V	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
t _{PHL} / t _{PLH}	propagation delay nA, nB to nY		39 14 11	125 25 21		155 31 26		190 38 32	ns	2.0 4.5 6.0	Fig. 8	
t _{THL} / t _{TLH}	output transition time		19 7 6	75 15 13		95 19 16		110 22 19	ns	2.0 4.5 6.0	Fig. 8	

74 HC/HCT 14

DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see chapter "HCMOS family characteristics", section "Family specifications". Transfer characteristics are given below.

Output capability: standard
 I_{CC} category: SSI

Note to HCT types

The value of additional quiescent supply current (ΔI_{CC}) for a unit load of 1 is given in the family specifications. To determine ΔI_{CC} per input, multiply this value by the unit load coefficient shown in the table below.

INPUT	UNIT LOAD COEFFICIENT
nA	0.3

Transfer characteristics for 74HCT

Voltages are referenced to GND (ground = 0 V)

SYMBOL	PARAMETER	T _{amb} (°C)						UNIT	TEST CONDITIONS		
		74HCT							V _{CC} V	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
V _{T+}	positive-going threshold	1.2 1.4		1.9 2.1	1.2 1.4	1.9 2.1	1.2 1.4	1.9 2.1	V	4.5 5.5	Figs 6 and 7
V _{T-}	negative-going threshold	0.5 0.6		1.2 1.4	0.5 0.6	1.2 1.4	0.5 0.6	1.2 1.4	V	4.5 5.5	Figs 6 and 7
V _H	hysteresis (V _{T+} - V _{T-})	0.4 0.4		- -	0.4 -	0.4 -	0.4 -	0.4 -	V	4.5 5.5	Figs 6 and 7

AC CHARACTERISTICS FOR 74HCT

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)						UNIT	TEST CONDITIONS		
		74HCT							V _{CC} V	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
t _{PHL} / t _{PLH}	propagation delay nA, nB to nY		18	31		39		47	ns	4.5	Fig. 8
t _{THL} / t _{TLH}	output transition time		7	15		19		22	ns	4.5	Fig. 8

TRANSFER CHARACTERISTIC WAVEFORMS

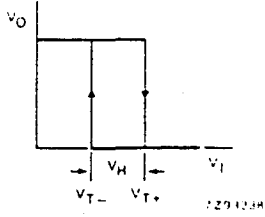


Fig. 6 Transfer characteristic.

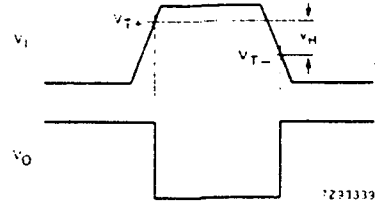


Fig. 7 Waveforms showing the definition of V_{T+} , V_{T-} and V_H ; where V_{T+} and V_{T-} are between limits of 20% and 70%.

AC WAVEFORMS

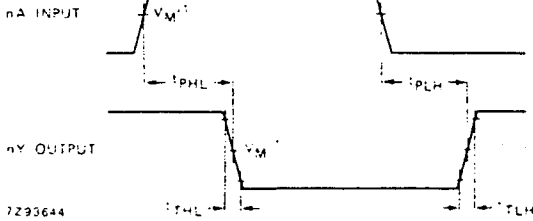


Fig. 8 Waveforms showing the input (nA, nB) to output (nY) propagation delays and the output transition times.

Note to AC waveforms

- (1) HC : $V_M = 50\%$; $V_I = \text{GND to } V_{CC}$.
- HCT: $V_M = 1.3 \text{ V}$; $V_I = \text{GND to } 3 \text{ V}$.

74 HC/HCT 132

QUAD 2-INPUT NAND SCHMITT TRIGGER

FEATURES

- Output capability: standard
- ICC category: SSI

GENERAL DESCRIPTION

The 74HC/HCT132 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7.

The 74HC/HCT132 contain four 2-input NAND gates which accept standard input signals. They are capable of transforming slowly changing input signals into sharply defined, jitter-free output signals.

The gate switches at different points for positive and negative-going signals. The difference between the positive voltage V_{T+} and the negative voltage V_{T-} is defined as the hysteresis voltage V_H .

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t_{PHL}/t_{PLH}	propagation delay nA, nB to nY	$C_L = 15 \text{ pF}$ $V_{CC} = 5 \text{ V}$	11	17	ns
C_i	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per gate	notes 1 and 2	24	20	pF

GND = 0 V; $T_{amb} = 25^\circ\text{C}$; $t_r = t_f = 6 \text{ ns}$

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in μW):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

- f_i = input frequency in MHz
- f_o = output frequency in MHz
- $\sum (C_L \times V_{CC}^2 \times f_o)$ = sum of outputs
- C_L = output load capacitance in pF
- V_{CC} = supply voltage in V

2. For HC the condition is $V_i = \text{GND to } V_{CC}$
For HCT the condition is $V_i = \text{GND to } V_{CC} - 1.5 \text{ V}$

ORDERING INFORMATION/PACKAGE OUTLINES

PC74HC/HCT132P: 14-lead DIL; plastic (SOT-27).

PC74HC/HCT132T: 14-lead mini-pack; plastic (SO-14; SOT-108A).

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 4, 9, 12	1A to 4A	data inputs
2, 5, 10, 13	1B to 4B	data inputs
3, 6, 8, 11	1Y to 4Y	data outputs
7	GND	ground (0 V)
14	V_{CC}	positive supply voltage

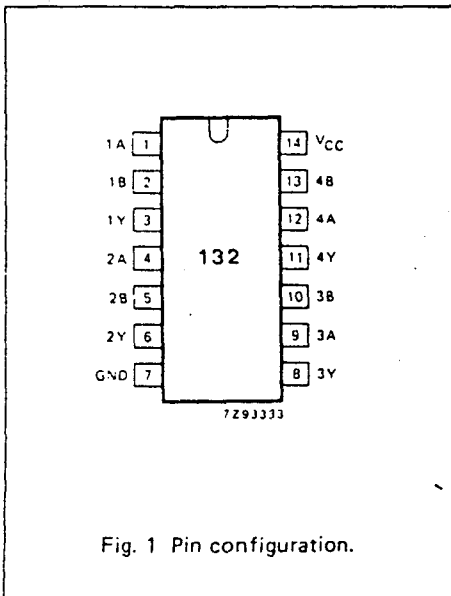


Fig. 1 Pin configuration.

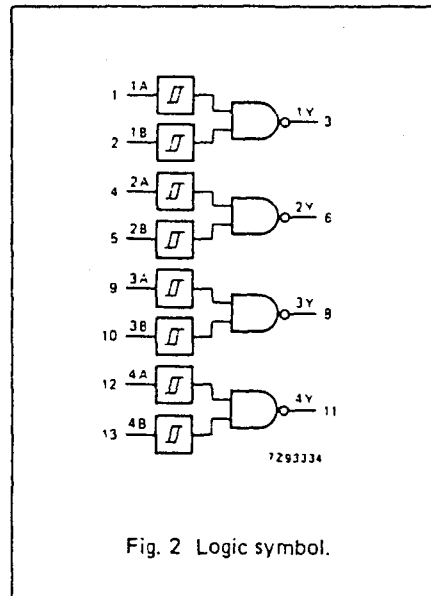


Fig. 2 Logic symbol.

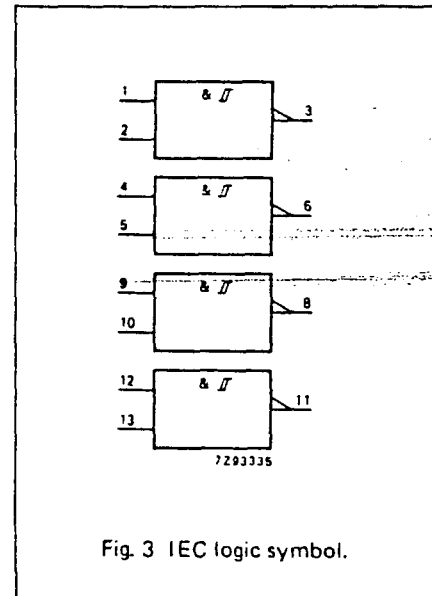
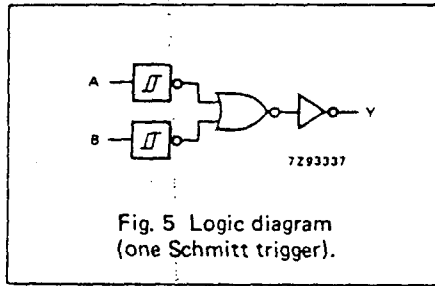
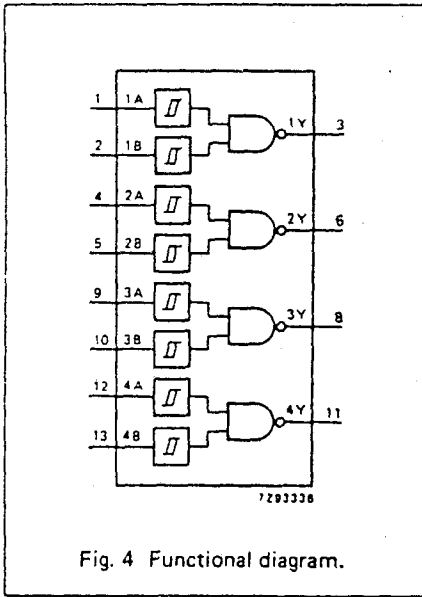


Fig. 3 IEC logic symbol.



APPLICATIONS

- Wave and pulse shapers
- Astable multivibrators
- Monostable multivibrators

FUNCTION TABLE

INPUTS		OUTPUT
nA	nB	nY
L	L	H
L	H	H
H	L	H
H	H	L

H = HIGH voltage level
L = LOW voltage level

74HC/HCT 132

DC CHARACTERISTICS FOR 74HC

For the DC characteristics see chapter "HCMOS family characteristics", section "Family specifications". Transfer characteristics are given below.

Output capability: standard
I_{CC} category: SSI

Transfer characteristics for 74HC

Voltages are referenced to GND (ground = 0 V)

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HC									V _{CC} V	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
V _{T+}	positive-going threshold	0.7 1.7 2.1		1.5 3.15 4.2	0.7 1.7 2.1	1.5 3.15 4.2	0.7 1.7 2.1	1.5 3.15 4.2	V	2.0 4.5 6.0	Figs 6 and 7	
V _{T-}	negative-going threshold	0.3 0.9 1.2		1.0 2.2 3.0	0.3 0.9 1.2	1.0 2.2 3.0	0.3 0.9 1.2	1.0 2.2 3.0	V	2.0 4.5 6.0	Figs 6 and 7	
V _H	hysteresis (V _{T+} - V _{T-})	0.2 0.4 0.6		1.0 1.4 1.6	0.2 0.4 0.6	1.0 1.4 1.6	0.2 0.4 0.6	1.0 1.4 1.6	V	2.0 4.5 6.0	Figs 6 and 7	

AC CHARACTERISTICS FOR 74HC

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HC									V _{CC} V	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
t _{PHL} / t _{PLH}	propagation delay nA, nB to nY		36 13 10	125 25 21		155 31 26		190 38 32	ns	2.0 4.5 6.0	Fig. 8	
t _{THL} / t _{TLH}	output transition time		19 7 6	75 15 13		95 19 16		110 22 19	ns	2.0 4.5 6.0	Fig. 8	

74 HC/HCT 132

DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see chapter "HCMOS family characteristics", section "Family specifications". Transfer characteristics are given below.

Output capability: standard
 I_{CC} category: SSI

Note to HCT types

The value of additional quiescent supply current (ΔI_{CC}) for a unit load of 1 is given in the family specifications. To determine ΔI_{CC} per input, multiply this value by the unit load coefficient shown in the table below.

input	unit load coefficient
nA, nB	0,3

Transfer characteristics for 74HCT

Voltages are referenced to GND (ground = 0 V)

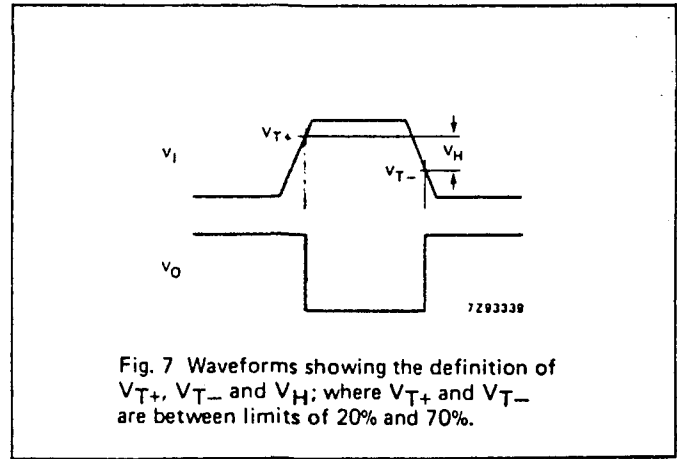
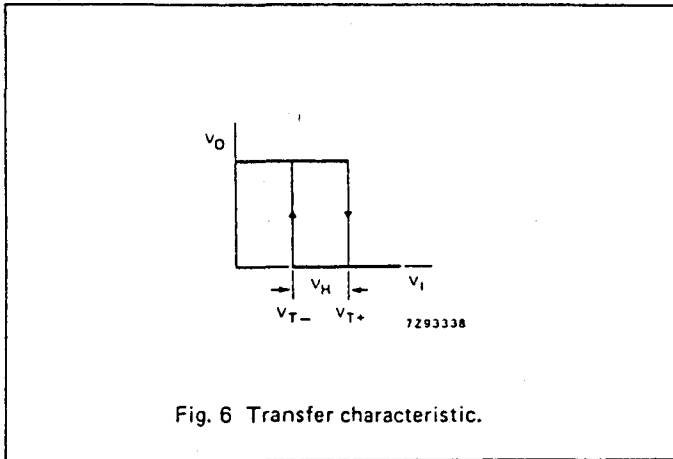
SYMBOL	PARAMETER	T_{amb} (°C)						UNIT	TEST CONDITIONS		
		74HCT							V_{CC} V	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.		max.		
V_{T+}	positive-going threshold	1.2 1.4		1.9 2.1	1.2 1.4	1.9 2.1	1.2 1.4	1.9 2.1	V	4.5 5.5	Figs 6 and 7
V_{T-}	negative-going threshold	0.5 0.6		1.2 1.4	0.5 0.6	1.2 1.4	0.5 0.6	1.2 1.4	V	4.5 5.5	Figs 6 and 7
V_H	hysteresis ($V_{T+} - V_{T-}$)	0.4 0.4		— —	0.4 0.4	— —	0.4 0.4	— —	V	4.5 5.5	Figs 6 and 7

AC CHARACTERISTICS FOR 74HCT

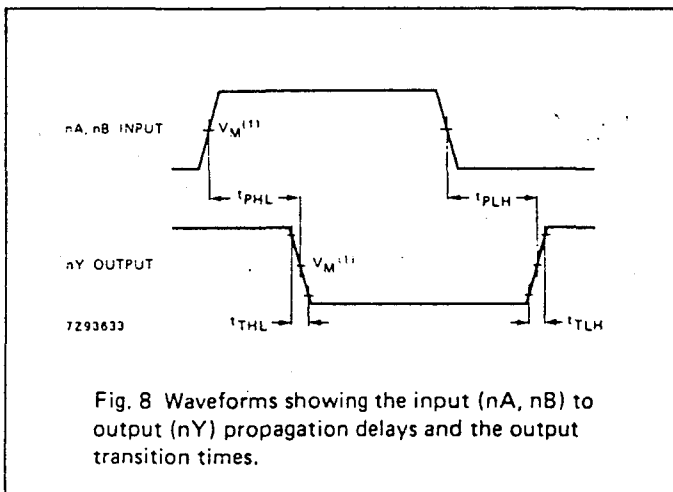
GND = 0 V; $t_r = t_f = 6$ ns; $C_L = 50$ pF

SYMBOL	PARAMETER	T_{amb} (°C)						UNIT	TEST CONDITIONS		
		74HCT							V_{CC} V	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.		max.		
t_{PHL}/t_{PLH}	propagation delay nA, nB to nY		20	33		41		50	ns	4.5	Fig. 8
t_{THL}/t_{TLH}	output transition time		7	15		19		22	ns	4.5	Fig. 8

TRANSFER CHARACTERISTIC WAVEFORMS



AC WAVEFORMS



Note to AC waveforms

- (1) HC : $V_M = 50\%$; $V_I = \text{GND to } V_{CC}$.
- HCT: $V_M = 1.3 \text{ V}$; $V_I = \text{GND to } 3 \text{ V}$.

74HC161 (HCT 161)

PRESETTABLE SYNCHRONOUS 4-BIT BINARY COUNTER; ASYNCHRONOUS RESET

FEATURES

- Synchronous counting and loading
- Two count enable inputs for n-bit cascading
- Positive-edge triggered clock
- Asynchronous reset
- Output capability: standard
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT161 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSSTTL). They are specified in compliance with JEDEC standard no. 7.

The 74HC/HCT161 are synchronous presettable binary counters which feature an internal look-ahead carry and can be used for high-speed counting.

Synchronous operation is provided by having all flip-flops clocked simultaneously on the positive-going edge of the clock (CP).

The outputs (Q₀ to Q₃) of the counters may be preset to a HIGH or LOW level. A LOW level at the parallel enable input (\overline{PE}) disables the counting action and causes the data at the data inputs (D₀ to D₃) to be loaded into the counter on the positive-going edge of the clock (providing that the set-up and hold time requirements for \overline{PE} are met). Preset takes place regardless of the levels at count enable inputs (CEP and CET).

(continued on next page)

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t _{PHL} / t _{PLH}	propagation delay CP to Q _n	C _L = 15 pF V _{CC} = 5 V	19	20	ns
	CP to TC		21	24	ns
	\overline{MR} to Q _n		20	25	ns
	\overline{MR} to TC		20	26	ns
	CET to TC		10	14	ns
f _{max}	maximum clock frequency		44	45	MHz
C _I	input capacitance		3.5	3.5	pF
CPD	power dissipation capacitance per package	notes 1 and 2	33	35	pF

GND = 0 V; T_{amb} = 25 °C; t_r = t_f = 6 ns

Notes

1. CPD is used to determine the dynamic power dissipation (P_D in μW):

$$P_D = CPD \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f_i = input frequency in MHz

f_o = output frequency in MHz

∑ (C_L × V_{CC}² × f_o) = sum of outputs

C_L = output load capacitance in pF

V_{CC} = supply voltage in V

2. For HC the condition is V_I = GND to V_{CC}
For HCT the condition is V_I = GND to V_{CC} - 1.5 V

ORDERING INFORMATION/PACKAGE OUTLINES

PC74HC/HCT161P: 16-lead DIL; plastic (SOT-38Z).

PC74HC/HCT161T: 16-lead mini-pack; plastic (SO-16; SOT-109A).

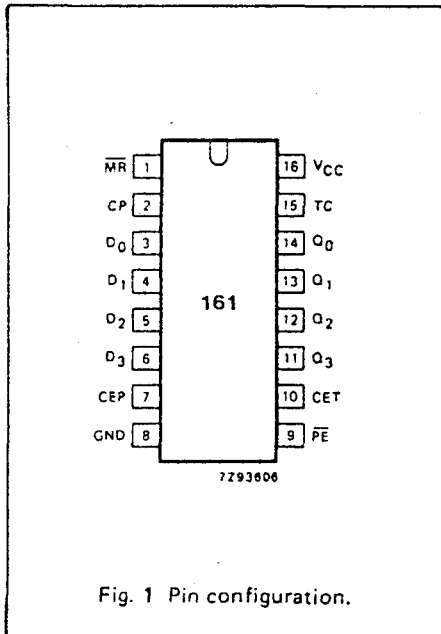


Fig. 1 Pin configuration.

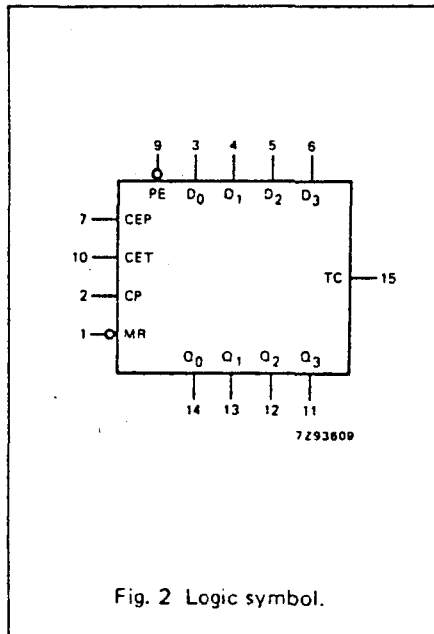


Fig. 2 Logic symbol.

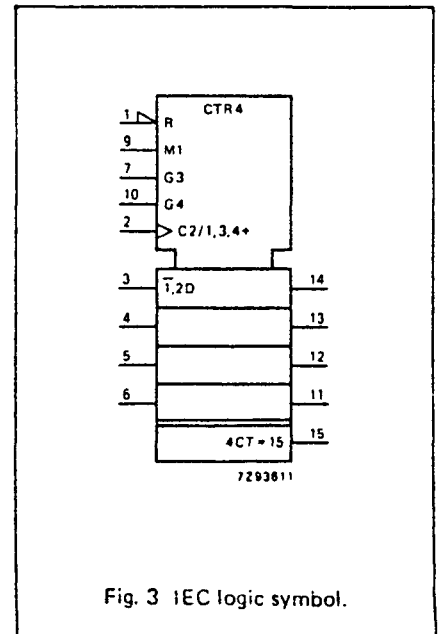


Fig. 3 IEC logic symbol.

74HC161 (HCT 161)

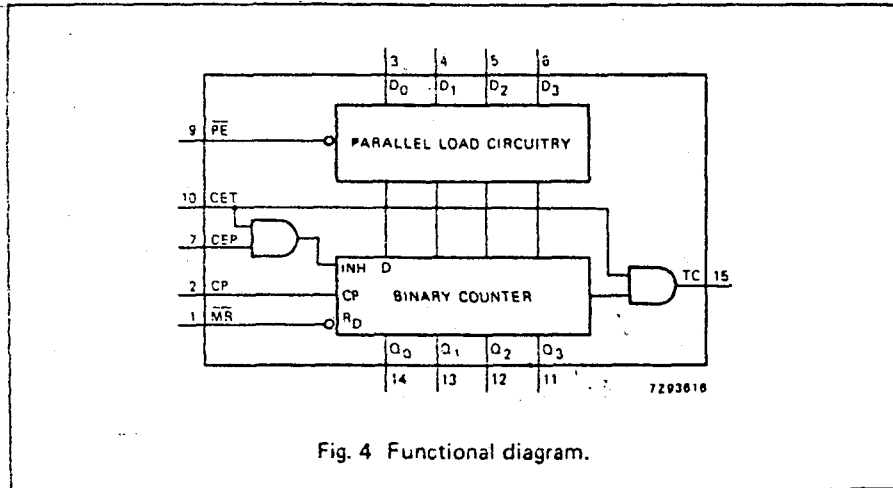


Fig. 4 Functional diagram.

GENERAL DESCRIPTION (Cont'd.)

A LOW level at the master reset input (\overline{MR}) sets all four outputs of the flip-flops (Q_0 to Q_3) to LOW level regardless of the levels at CP, \overline{PE} , CET and CEP inputs (thus providing an asynchronous clear function).

The look-ahead carry simplifies serial cascading of the counters. Both count enable inputs (CEP and CET) must be HIGH to count. The CET input is fed forward to enable the terminal count output (TC). The TC output thus enabled will produce a HIGH output pulse of a duration approximately equal to a HIGH level output of Q_0 . This pulse can be used to enable the next cascaded stage.

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1	\overline{MR}	asynchronous master reset (active LOW)
2	CP	clock input (LOW-to-HIGH, edge-triggered)
3, 4, 5, 6	D_0 to D_3	data inputs
7	CEP	count enable input
8	GND	ground (0 V)
9	\overline{PE}	parallel enable input (active LOW)
10	CET	count enable carry input
14, 13, 12, 11	Q_0 to Q_3	flip-flop outputs
15	TC	terminal count output
16	VCC	positive supply voltage

FUNCTION TABLE

OPERATING MODE	INPUTS						OUTPUTS	
	\overline{MR}	CP	CEP	CET	\overline{PE}	D_n	Q_n	TC
reset (clear)	L	X	X	X	X	X	L	L
parallel load	H	↑	X	X	l	l	L	L
	H	↑	X	X	l	h	H	*
count	H	↑	h	h	h	X	count	*
hold (do nothing)	H	X	l	X	h	X	q_n	*
	H	X	X	l	h	X	q_n	L

Note to function table

* The TC output is HIGH when CET is HIGH and the counter is at terminal count (HHHH).

H = HIGH voltage level

h = HIGH voltage level one set-up time prior to the LOW-to-HIGH CP transition

L = LOW voltage level

l = LOW voltage level one set-up time prior to the LOW-to-HIGH CP transition

q = lower case letters indicate the state of the referenced output one set-up time prior to the LOW-to-HIGH CP transition

X = don't care

↑ = LOW-to-HIGH CP transition

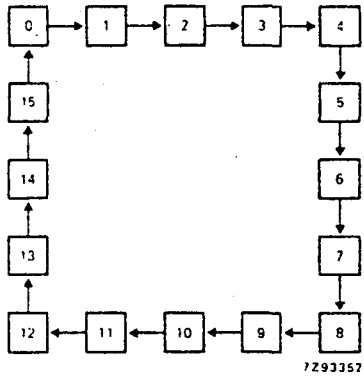


Fig. 5 State diagram.

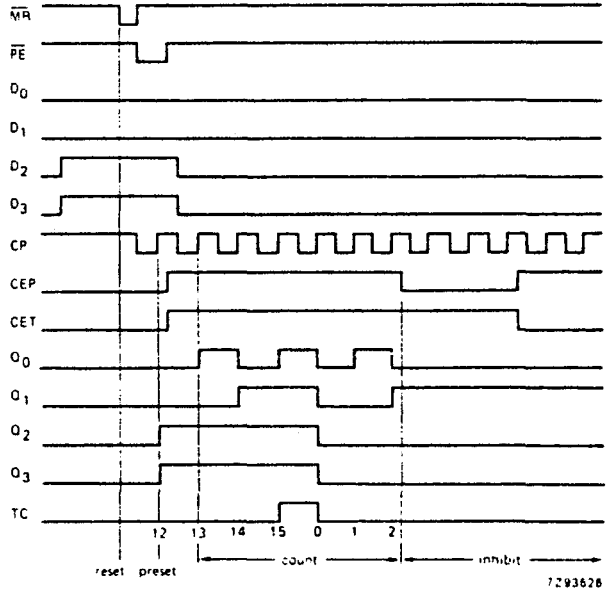


Fig. 6 Typical timing sequence: reset outputs to zero; preset to binary twelve; count to thirteen, fourteen, fifteen, zero, one and two; inhibit.

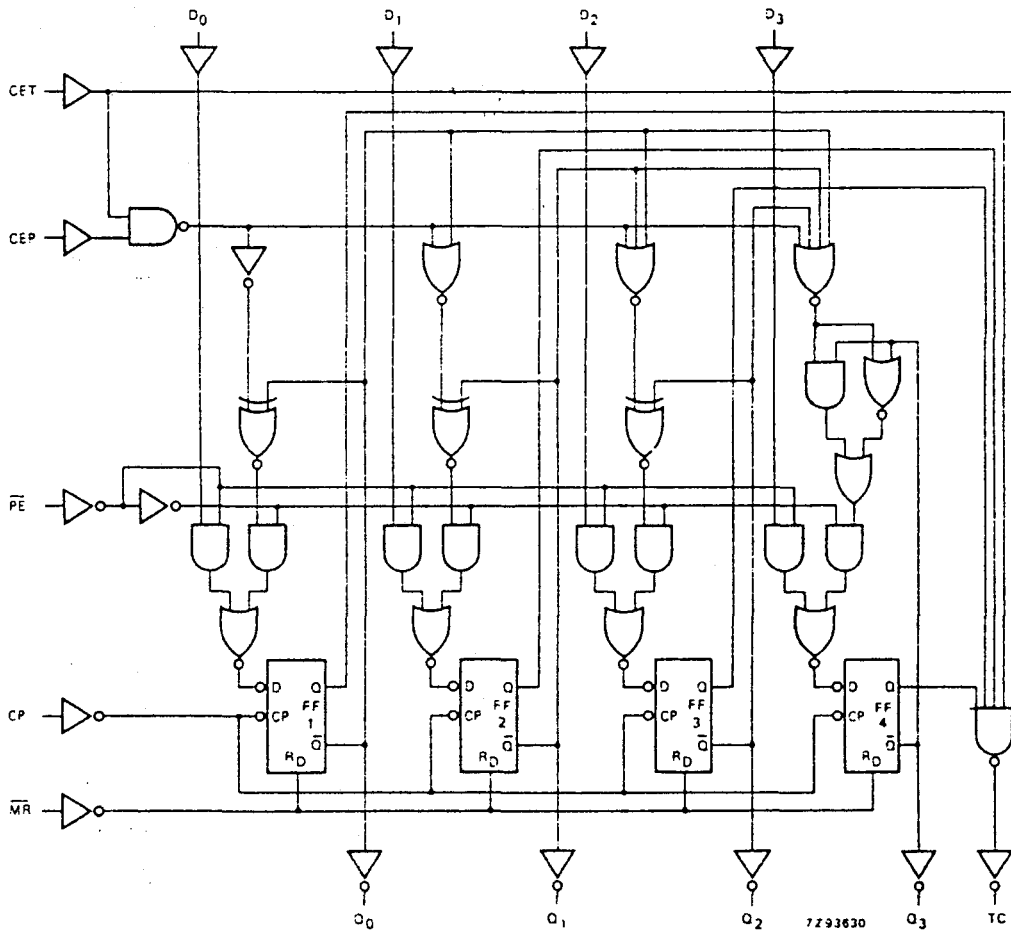


Fig. 7 Logic diagram.

74HC161 (HCT 161)

DC CHARACTERISTICS FOR 74HC

For the DC characteristics see chapter "HCMOS family characteristics", section "Family specifications".

Output capability: standard

I_{CC} category: MSI

AC CHARACTERISTICS FOR 74HC

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)						UNIT	TEST CONDITIONS		
		74HC							V _{CC} V	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.		max.		
t _{PHL} / t _{PLH}	propagation delay CP to Q _n		61	190		240		285	ns	2.0 4.5 6.0	Fig. 8
			22	38		48		57			
			18	32		41		48			
t _{PHL} / t _{PLH}	propagation delay CP to TC		69	215		270		325	ns	2.0 4.5 6.0	Fig. 8
			25	43		54		65			
			20	37		46		55			
t _{PHL}	propagation delay MR to Q _n		63	210		265		315	ns	2.0 4.5 6.0	Fig. 9
			23	42		53		63			
			18	36		45		54			
t _{PHL}	propagation delay MR to TC		63	220		275		330	ns	2.0 4.5 6.0	Fig. 9
			23	44		55		66			
			18	37		47		56			
t _{PHL} / t _{PLH}	propagation delay CET to TC		33	150		190		225	ns	2.0 4.5 6.0	Fig. 10
			12	30		38		45			
			10	26		33		38			
t _{THL} / t _{TLH}	output transition time		19	75		95		110	ns	2.0 4.5 6.0	Figs 8 and 10
			7	15		19		22			
			6	13		16		19			
t _w	clock pulse width HIGH or LOW	110	22		140		165		ns	2.0 4.5 6.0	Fig. 8
		22	8		28		33				
		19	6		24		28				
t _w	master reset pulse width; LOW	80	19		100		120		ns	2.0 4.5 6.0	Fig. 9
		16	7		20		24				
		14	6		17		20				
t _{rem}	removal time MR to CP	100	19		125		150		ns	2.0 4.5 6.0	Fig. 9
		20	7		25		30				
		17	6		21		26				
t _{su}	set-up time D _n to CP	80	25		100		120		ns	2.0 4.5 6.0	Fig. 11
		16	9		20		24				
		14	7		17		20				
t _{su}	set-up time PE to CP	100	30		125		150		ns	2.0 4.5 6.0	Fig. 11
		20	11		25		30				
		17	9		21		26				
t _{su}	set-up time CEP, CET to CP	170	47		215		255		ns	2.0 4.5 6.0	Fig. 12
		34	17		43		51				
		29	14		37		43				
t _h	hold time D _n , PE, CEP, CET to CP	0	-14		0		0		ns	2.0 4.5 6.0	Figs 11 and 12
		0	-5		0		0				
		0	-4		0		0				
f _{max}	maximum clock pulse frequency	4.6	13		3.6		3.0		MHz	2.0 4.5 6.0	Fig. 8
		23	40		18		15				
		27	48		21		18				

74 HC161(HCT 161)

DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see chapter "HCMOS family characteristics", section "Family specifications".

Output capability: standard

I_{CC} category: MSI

Note to HCT types

The value of additional quiescent supply current (ΔI_{CC}) for a unit load of 1 is given in the family specifications.

To determine ΔI_{CC} per input, multiply this value by the unit load coefficient shown in the table below.

INPUT	UNIT LOAD COEFFICIENT
\overline{MR}	0.95
CP	0.80
CEP	0.25

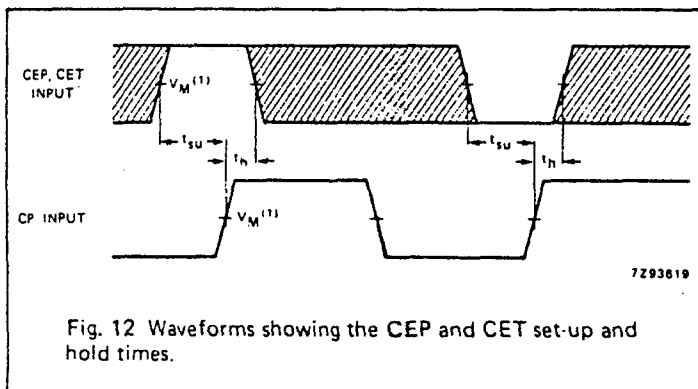
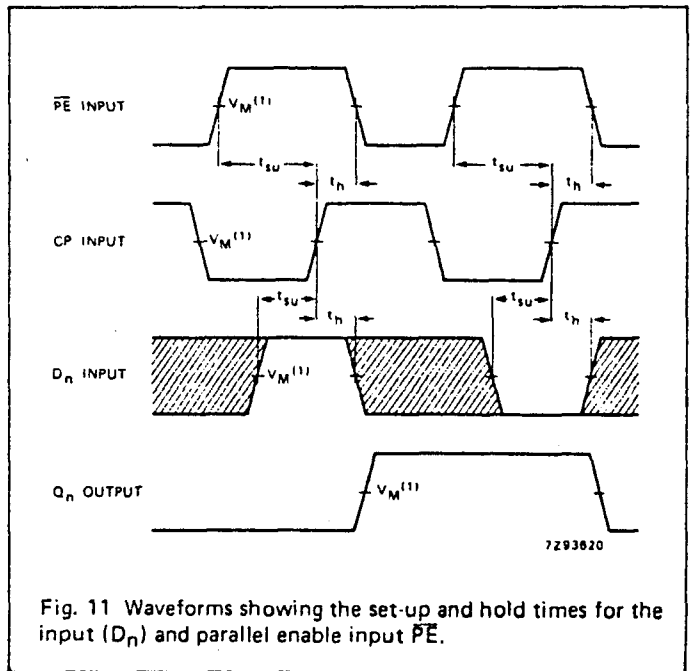
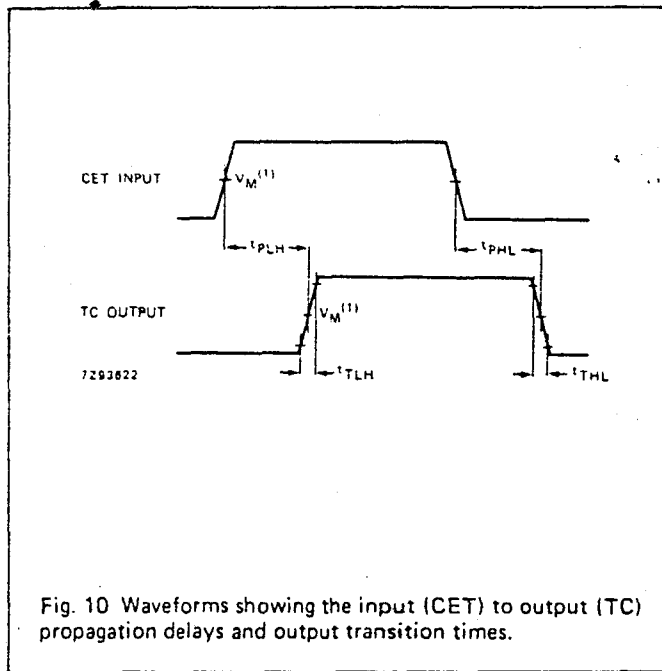
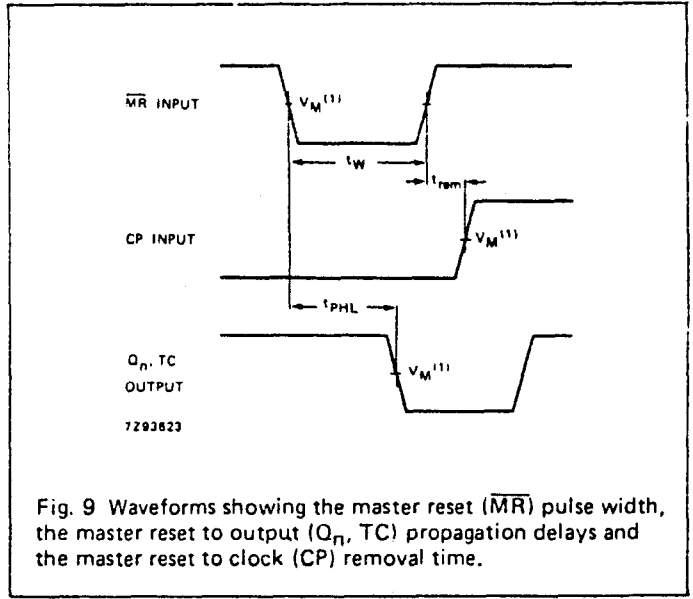
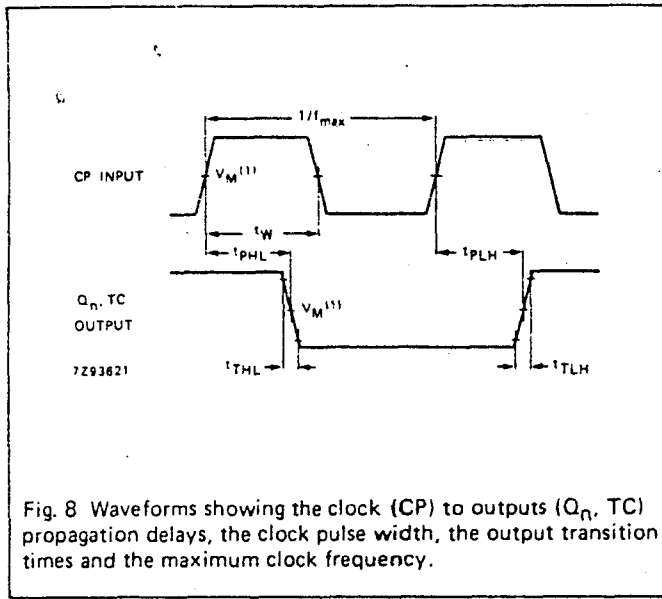
INPUT	UNIT LOAD COEFFICIENT
D_n	0.25
CET	0.75
\overline{PE}	0.30

AC CHARACTERISTICS FOR 74HCT

GND = 0 V; $t_r = t_f = 6$ ns; $C_L = 50$ pF

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HCT									V _{CC} V	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
t _{PHL} / t _{PLH}	propagation delay CP to Q _n		23	43		54		65	ns	4.5	Fig. 8	
t _{PHL} / t _{PLH}	propagation delay CP to TC		28	48		60		72	ns	4.5	Fig. 8	
t _{PHL}	propagation delay \overline{MR} to Q _n		29	46		58		69	ns	4.5	Fig. 9	
t _{PHL}	propagation delay \overline{MR} to TC		30	51		64		77	ns	4.5	Fig. 9	
t _{PHL} / t _{PLH}	propagation delay CET to TC		17	35		44		53	ns	4.5	Fig. 10	
t _{THL} / t _{TLH}	output transition time		7	15		19		22	ns	4.5	Figs 8 and 10	
t _w	clock pulse width HIGH or LOW	22	7		28		33		ns	4.5	Fig. 8	
t _w	master reset pulse width; LOW	20	10		25		30		ns	4.5	Fig. 9	
t _{rem}	removal time \overline{MR} to CP	20	6		25		30		ns	4.5	Fig. 9	
t _{su}	set-up time D _n to CP	18	8		23		27		ns	4.5	Fig. 11	
t _{su}	set-up time \overline{PE} to CP	30	17		38		45		ns	4.5	Fig. 11	
t _{su}	set-up time CEP, CET to CP	40	17		50		60		ns	4.5	Fig. 12	
t _h	hold time D _n , \overline{PE} , CEP, CET to CP	0	-7		0		0		ns	4.5	Figs 11 and 12	
f _{max}	maximum clock pulse frequency	23	41		18		15		MHz	4.5	Fig. 8	

AC WAVEFORMS



Note to Figs 11 and 12
 The shaded areas indicate when the input is permitted to change for predictable output performance.

Note to AC waveforms
 (1) HC : $V_M = 50\%$; $V_I = \text{GND to } V_{CC}$.
 HCT: $V_M = 1.3 \text{ V}$; $V_I = \text{GND to } 3 \text{ V}$.

74HC/HCT 244

OCTAL BUFFER/LINE DRIVER; 3-STATE

FEATURES

- Output capability: bus driver
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT244 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSSTTL). They are specified in compliance with JEDEC standard no. 7.

The 74HC/HCT244 are octal non-inverting buffer/line drivers with 3-state outputs. The 3-state outputs are controlled by the output enable inputs 1OE and 2OE. A HIGH on nOE causes the outputs to assume a high impedance OFF-state. The "244" is identical to the "240" but has non-inverting outputs.

FUNCTION TABLE

INPUTS		OUTPUT
nOE	nAn	nYn
L	L	L
L	H	H
H	X	Z

H = HIGH voltage level
L = LOW voltage level
X = don't care
Z = high impedance OFF-state

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t _{PHL} / t _{PLH}	propagation delay 1A _n to 1Y _n ; 2A _n to 2Y _n	C _L = 15 pF V _{CC} = 5 V	9	11	ns
C _I	input capacitance		3.5	3.5	pF
C _{PD}	power dissipation capacitance per buffer	notes 1 and 2	35	35	pF

GND = 0 V; T_{amb} = 25 °C; t_r = t_f = 6 ns

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in μW):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f_i = input frequency in MHz

C_L = output load capacitance in pF

f_o = output frequency in MHz

V_{CC} = supply voltage in V

∑ (C_L × V_{CC}² × f_o) = sum of outputs

2. For HC the condition is V_I = GND to V_{CC}
For HCT the condition is V_I = GND to V_{CC} - 1.5 V

ORDERING INFORMATION/PACKAGE OUTLINES

PC74HC/HCT244P: 20-lead DIL; plastic (SOT-146).

PC74HC/HCT244T: 20-lead mini-pack; plastic (SO-20; SOT-163A).

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1	1OE	output enable input (active LOW)
2, 4, 6, 8	1A ₀ to 1A ₃	data inputs
3, 5, 7, 9	2Y ₀ to 2Y ₃	bus outputs
10	GND	ground (0 V)
17, 15, 13, 11	2A ₀ to 2A ₃	data inputs
18, 16, 14, 12	1Y ₀ to 1Y ₃	bus outputs
19	2OE	output enable input (active LOW)
20	V _{CC}	positive supply voltage

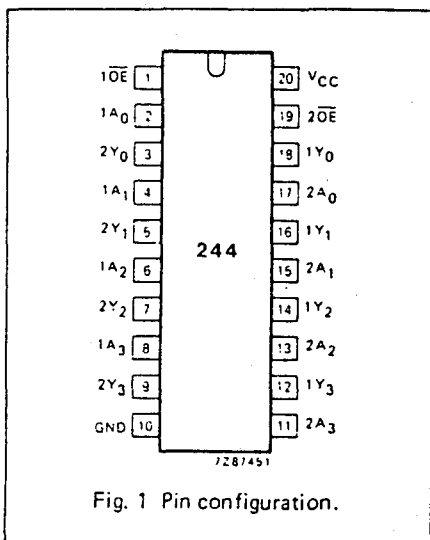


Fig. 1 Pin configuration.

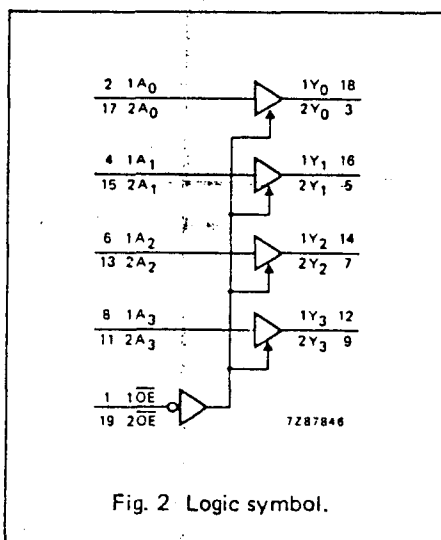


Fig. 2 Logic symbol.

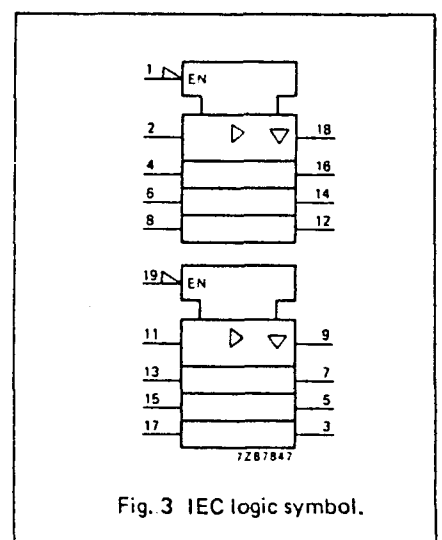


Fig. 3 IEC logic symbol.

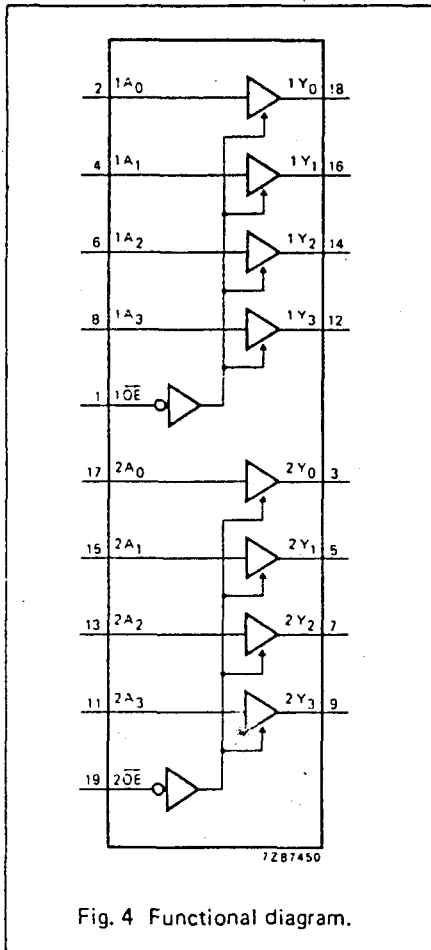


Fig. 4 Functional diagram.

74HC/HCT 244

DC CHARACTERISTICS FOR 74HC

For the DC characteristics see chapter "HCMOS family characteristics", section "Family specifications".

Output capability: bus driver

I_{CC} category: MSI

AC CHARACTERISTICS FOR 74HC

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HC									V _{CC} V	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
t _{PHL} / t _{PLH}	propagation delay 1A _n to 1Y _n ; 2A _n to 2Y _n		30	110		145		165	ns	2.0	Fig. 5	
			11	22		28		33		4.5		
			9	19		24		28		6.0		
t _{PZH} / t _{PZL}	3-state output enable time 1 $\overline{O}E$ to 1Y _n ; 2 $\overline{O}E$ to 2Y _n		36	150		190		225	ns	2.0	Fig. 6	
			13	30		38		45		4.5		
			10	26		33		38		6.0		
t _{PHZ} / t _{PLZ}	3-state output enable time 1 $\overline{O}E$ to 1Y _n ; 2 $\overline{O}E$ to 2Y _n		39	150		190		225	ns	2.0	Fig. 6	
			14	30		38		45		4.5		
			11	26		33		38		6.0		
t _{THL} / t _{TLH}	output transition time		14	60		75		90	ns	2.0	Fig. 5	
			5	12		15		18		4.5		
			4	10		13		15		6.0		

DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see chapter "HCMOS family characteristics", section "Family specifications".

Output capability: bus driver

I_{CC} category: MSI

Note to HCT types

The value of additional quiescent supply current (ΔI_{CC}) for a unit load of 1 is given in the family specifications. To determine ΔI_{CC} per input, multiply this value by the unit load coefficient shown in the table below.

input	unit load coefficient
1A _n	0.70
2A _n	0.70
1 $\overline{O}E$	0.70
2 $\overline{O}E$	0.70

AC CHARACTERISTICS FOR 74HCT

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)						UNIT	TEST CONDITIONS		
		74HCT							V _{CC} V	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
t _{PHL} / t _{PLH}	propagation delay 1A _n to 1Y _n ; 2A _n to 2Y _n		13	22		28		33	ns	4.5	Fig. 5
t _{PZH} / t _{PZL}	3-state output enable time 1 $\overline{O}E$ to 1Y _n ; 2 $\overline{O}E$ to 2Y _n		15	30		38		45	ns	4.5	Fig. 6
t _{PHZ} / t _{PLZ}	3-state output enable time 1 $\overline{O}E$ to 1Y _n ; 2 $\overline{O}E$ to 2Y _n		15	25		31		38	ns	4.5	Fig. 6
t _{THL} / t _{TLH}	output transition time		5	12		15		18	ns	4.5	Fig. 5

AC WAVEFORMS

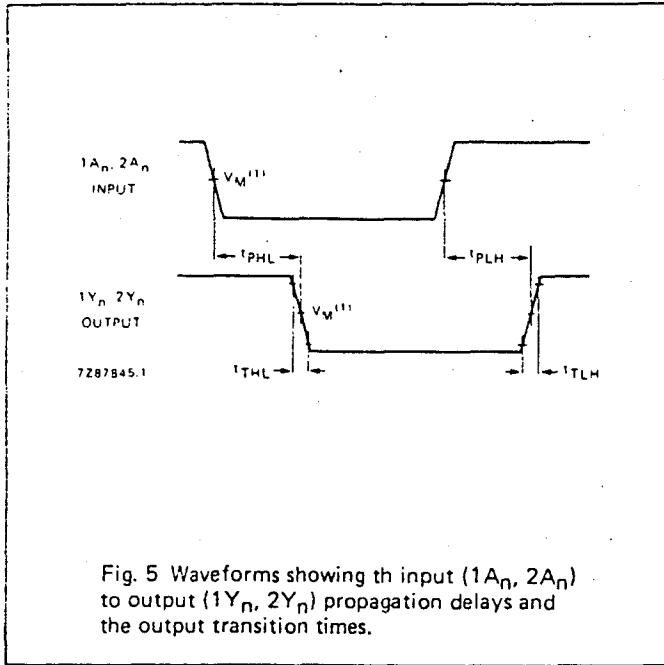


Fig. 5 Waveforms showing the input (1A_n, 2A_n) to output (1Y_n, 2Y_n) propagation delays and the output transition times.

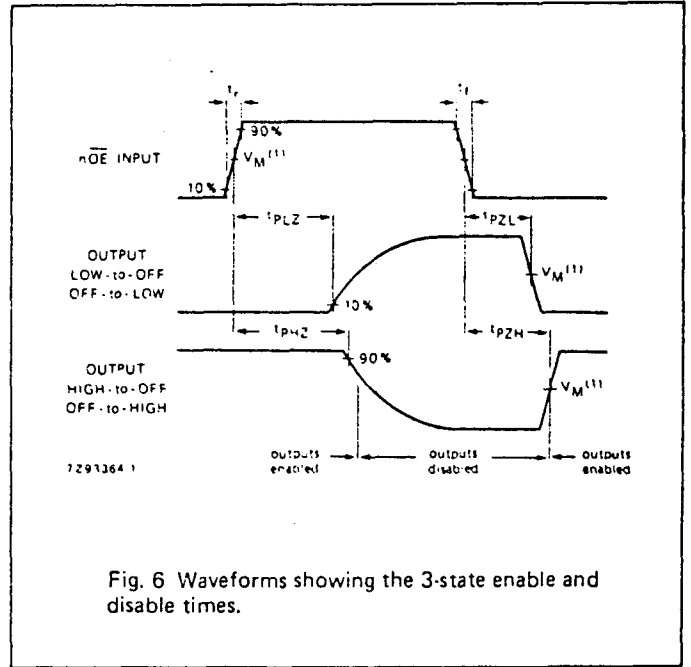


Fig. 6 Waveforms showing the 3-state enable and disable times.

Note to AC waveforms

- (1) HC : V_M = 50%; V_I = GND to V_{CC}.
- HCT: V_M = 1.3 V; V_I = GND to 3 V.

OCTAL D-TYPE TRANSPARENT LATCH; 3-STATE

FEATURES

- 3-state non-inverting outputs for bus oriented applications
- Common 3-state output enable input
- Functionally identical to the "563", "573" and "533"
- Output capability: bus driver
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT373 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LS TTL). They are specified in compliance with JEDEC standard no. 7.

The 74HC/HCT373 are octal D-type transparent latches featuring separate D-type inputs for each latch and 3-state outputs for bus oriented applications. A latch enable (LE) input and an output enable (\overline{OE}) input are common to all latches.

The "373" consists of eight D-type transparent latches with 3-state true outputs. When LE is HIGH, data at the D_n inputs enter the latches. In this condition the latches are transparent, i.e. a latch output will change state each time its corresponding D-input changes.

When LE is LOW the latches store the information that was present at the D-inputs a set-up time preceding the HIGH-to-LOW transition of LE.

When \overline{OE} is LOW, the contents of the 8 latches are available at the outputs.

When \overline{OE} is HIGH, the outputs go to the high impedance OFF-state.

Operation of the \overline{OE} input does not affect the state of the latches.

The "373" is functionally identical to the "533", "563" and "573", but the "563" and "533" have inverted outputs and the "563" and "573" have a different pin arrangement.

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t _{PHL} / t _{PLH}	propagation delay D _n to Q _n LE to Q _n	C _L = 15 pF V _{CC} = 5 V	12 15	14 13	ns ns
C _I	input capacitance		3.5	3.5	pF
CPD	power dissipation capacitance per latch	notes 1 and 2	45	41	pF

GND = 0 V; T_{amb} = 25 °C; t_r = t_f = 6 ns

Notes

1. CPD is used to determine the dynamic power dissipation (P_D in μW):

$$P_D = CPD \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f_i = input frequency in MHz

f_o = output frequency in MHz

∑ (C_L × V_{CC}² × f_o) = sum of outputs

C_L = output load capacitance in pF

V_{CC} = supply voltage in V

2. For HC the condition is V_I = GND to V_{CC}
For HCT the condition is V_I = GND to V_{CC} - 1.5 V

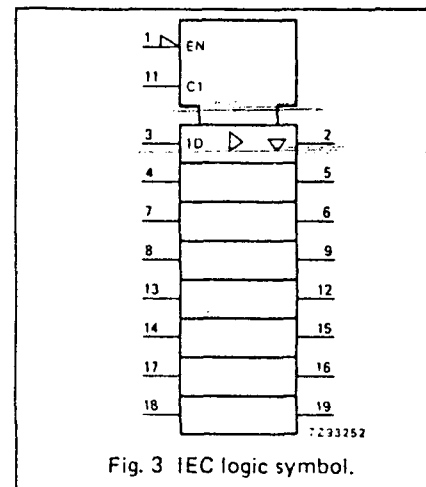
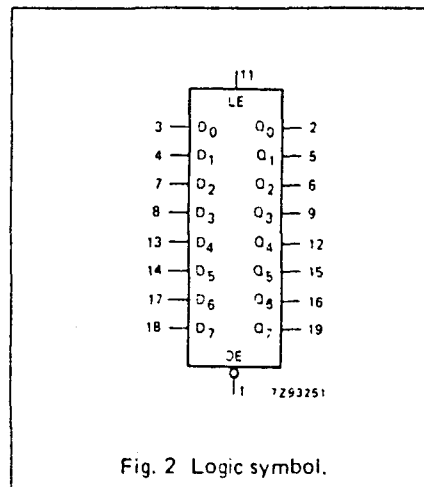
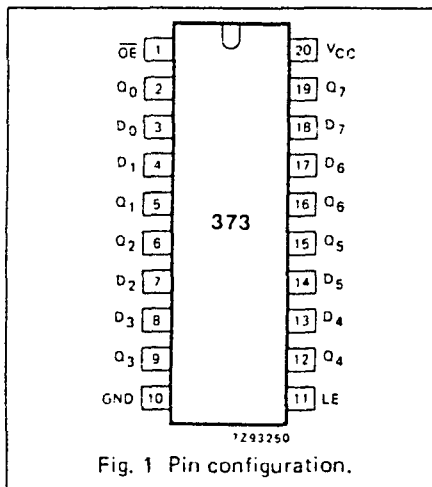
ORDERING INFORMATION/PACKAGE OUTLINES

PC74HC/HCT373P: 20-lead DIL; plastic (SOT-146).

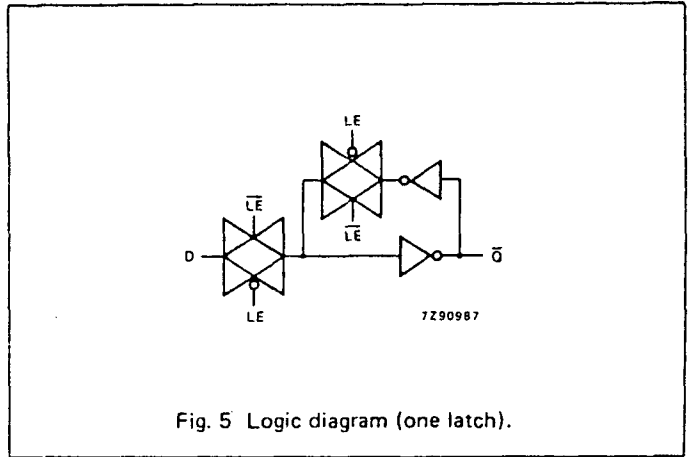
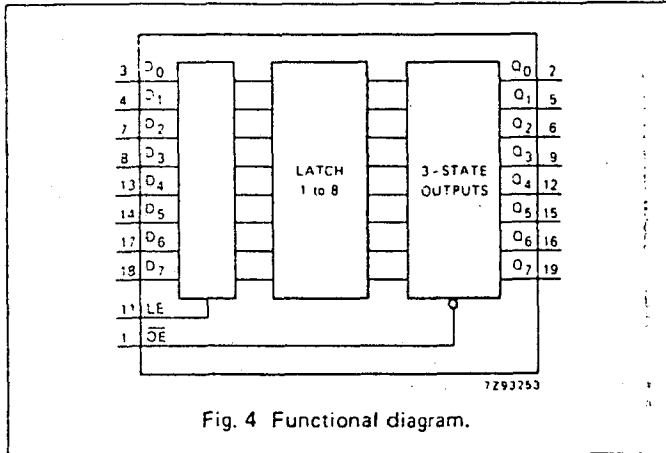
PC74HC/HCT373T: 20-lead mini-pack; plastic (SO-20; SOT-163A).

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1	\overline{OE}	3-state output enable input (active LOW)
2, 5, 6, 9, 12, 15, 16, 19	Q ₀ to Q ₇	3-state latch outputs
3, 4, 7, 8, 13, 14, 17, 18	D ₀ to D ₇	data inputs
10	GND	ground (0 V)
11	LE	latch enable input (active HIGH)
20	V _{CC}	positive supply voltage



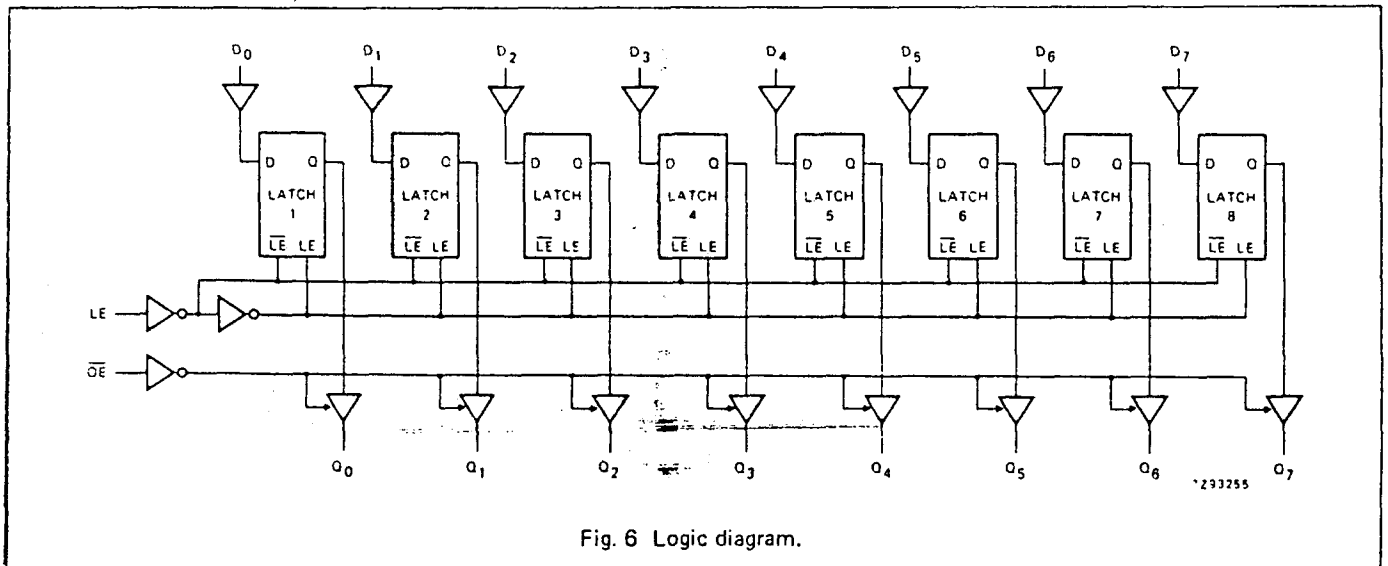
74 HC/HCT 373



FUNCTION TABLE

OPERATING MODES	INPUTS			INTERNAL LATCHES	OUTPUTS
	\overline{OE}	LE	D_n		Q_0 to Q_7
enable and read register (transparent mode)	L	H	L	L	L
	L	H	H	H	H
latch and read register	L	L	l	L	L
	L	L	h	H	H
latch register and disable outputs	H	L	l	L	Z
	H	L	h	H	Z

H = HIGH voltage level
h = HIGH voltage level one set-up time prior to the HIGH-to-LOW LE transition
L = LOW voltage level
l = LOW voltage level one set-up time prior to the HIGH-to-LOW LE transition
Z = high impedance OFF-state



74 HC/HCT373

DC CHARACTERISTICS FOR 74HC

For the DC characteristics see chapter "HCMOS family characteristics", section "Family specifications".

Output capability: bus driver

I_{CC} category: MSI

AC CHARACTERISTICS FOR 74HC

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HC									V _{CC} V	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
t _{PHL} / t _{PLH}	propagation delay D _n to Q _n		41 15 12	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig. 7	
t _{PHL} / t _{PLH}	propagation delay LE to Q _n		50 18 14	175 35 30		220 44 37		265 53 45	ns	2.0 4.5 6.0	Fig. 8	
t _{PZH} / t _{PZL}	3-state output enable time OE to Q _n		44 16 13	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig. 9	
t _{PHZ} / t _{PLZ}	3-state output disable time OE to Q _n		47 17 14	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig. 9	
t _{THL} / t _{TLH}	output transition time		14 5 4	60 12 10		75 15 13		90 18 15	ns	2.0 4.5 6.0	Fig. 7	
t _W	LE pulse width HIGH	80 16 14	17 6 5		100 20 17		120 24 20		ns	2.0 4.5 6.0	Fig. 8	
t _{su}	set-up time D _n to LE	50 10 9	14 5 4		65 13 11		75 15 13		ns	2.0 4.5 6.0	Fig. 10	
t _h	hold time D _n to LE	5 5 5	-8 -3 -2		5 5 5		5 5 5		ns	2.0 4.5 6.0	Fig. 10	

74 HC/HCT 373

DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see chapter "HCMOS family characteristics", section "Family specifications".

Output capability: bus driver

I_{CC} category: MSI

Note to HCT types

The value of additional quiescent supply current (ΔI_{CC}) for a unit load of 1 is given in the family specifications. To determine ΔI_{CC} per input, multiply this value by the unit load coefficient shown in the table below.

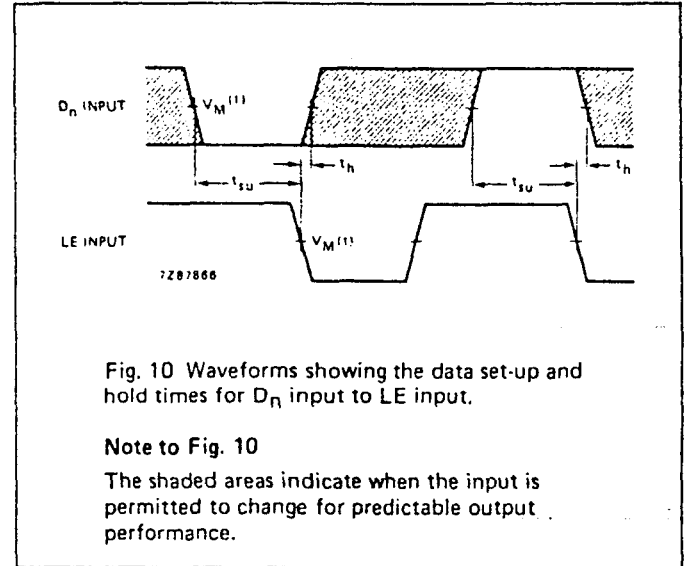
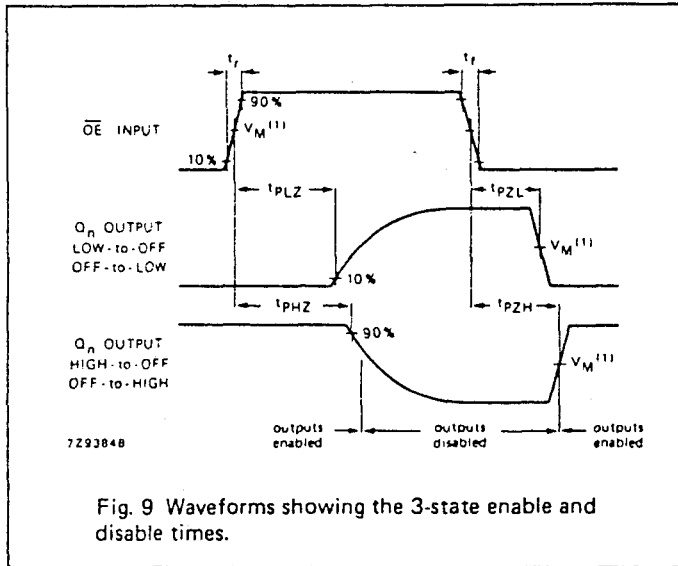
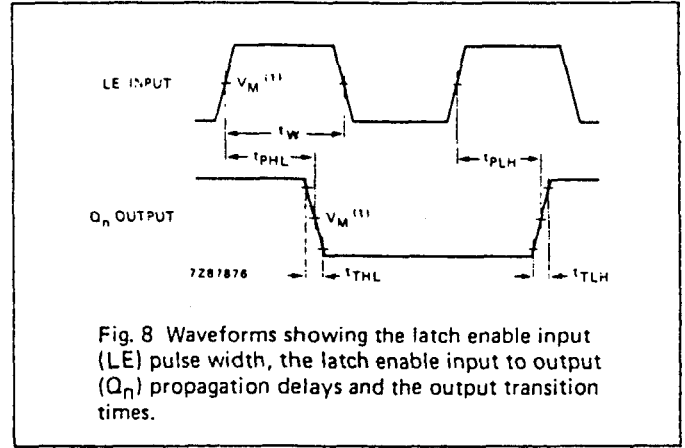
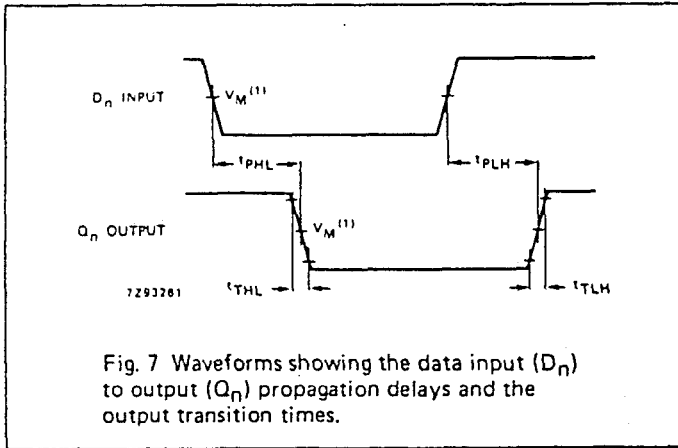
input	unit load coefficient
D _n	0.30
LE	1.50
\overline{OE}	1.00

AC CHARACTERISTICS FOR 74HCT

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)						UNIT	TEST CONDITIONS		
		74CHT							V _{CC} V	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
t _{PHL} / t _{PLH}	propagation delay D _n to Q _n		17	30		38		45	ns	4.5	Fig. 7
t _{PHL} / t _{PLH}	propagation delay LE to Q _n		16	32		40		48	ns	4.5	Fig. 8
t _{PZH} / t _{PZL}	3-state output enable time \overline{OE} to Q _n		19	32		40		48	ns	4.5	Fig. 9
t _{PHZ} / t _{PLZ}	3-state output disable time \overline{OE} to Q _n		18	30		38		45	ns	4.5	Fig. 9
t _{THL} / t _{TLH}	output transition time		5	12		15		18	ns	4.5	Fig. 7
t _W	LE pulse width HIGH	16	6		20		24		ns	4.5	Fig. 8
t _{SU}	set-up time D _n to LE	12	6		15		18		ns	4.5	Fig. 10
t _H	hold time D _n to LE	4	-1		4		4		ns	4.5	Fig. 10

AC WAVEFORMS



Note to AC waveforms

- (1) HC : $V_M = 50\%$; $V_I = \text{GND to } V_{CC}$.
- HCT: $V_M = 1.3 \text{ V}$; $V_I = \text{GND to } 3 \text{ V}$.

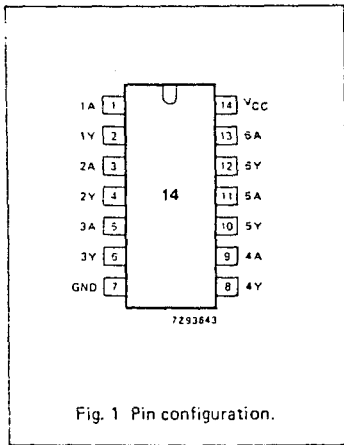


Fig. 1 Pin configuration.

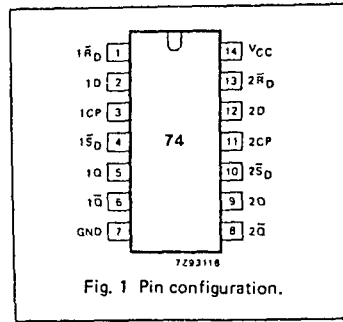


Fig. 1 Pin configuration.

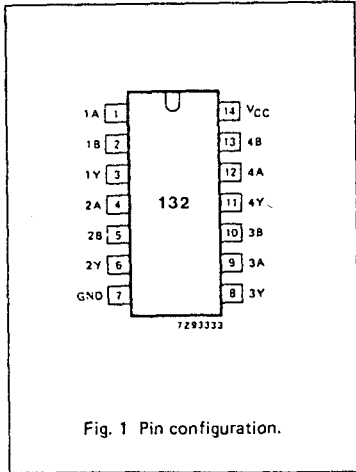


Fig. 1 Pin configuration.

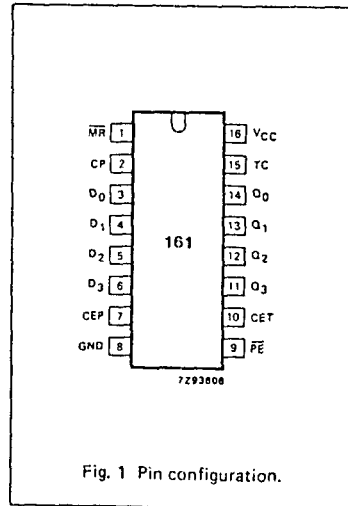


Fig. 1 Pin configuration.

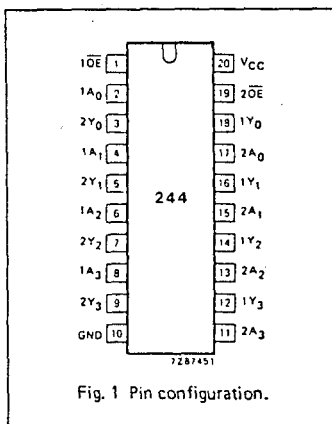


Fig. 1 Pin configuration.

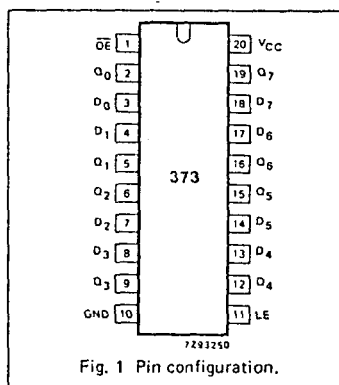


Fig. 1 Pin configuration.

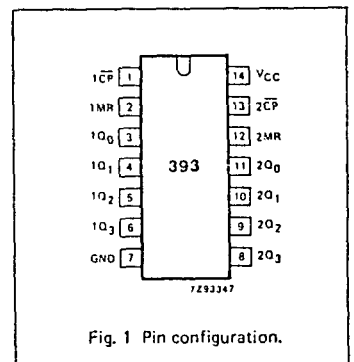
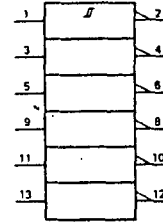
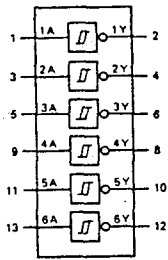


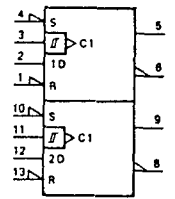
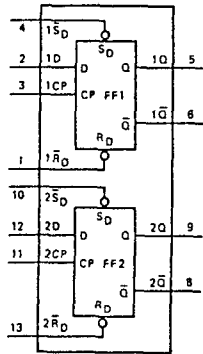
Fig. 1 Pin configuration.

HC/HCT14



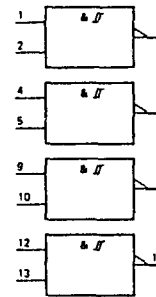
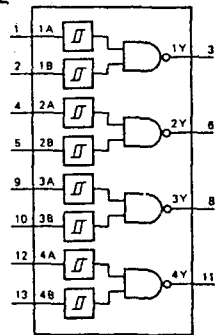
Hex inverting Schmitt trigger

HC/HCT74



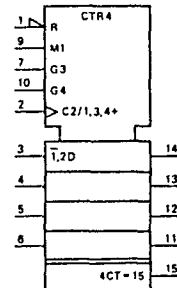
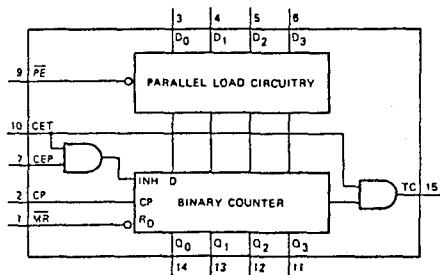
Dual D-type flip-flop with set and reset; positive-edge trigger

HC/HCT132



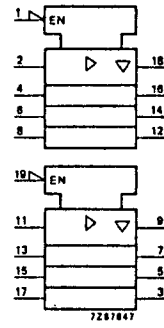
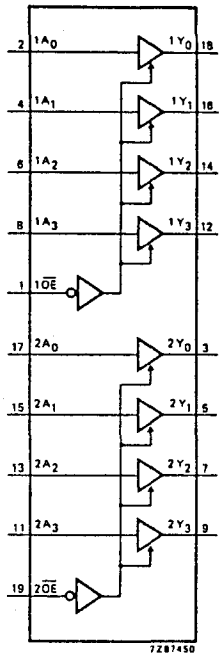
Quad 2-input NAND Schmitt trigger

HC/HCT161



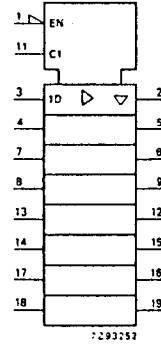
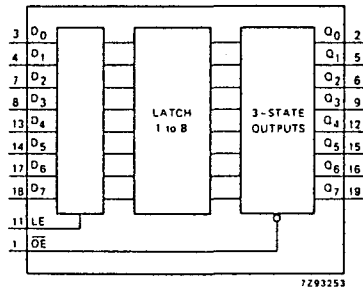
Presetable synchronous 4-bit binary counter; asynchronous reset

HC/HCT244



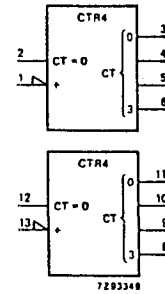
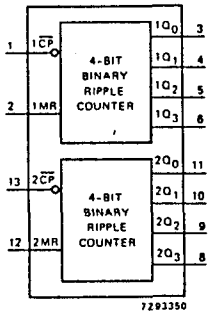
Octal buffer/line driver; 3-state

HC/HCT373



Octal D-type transparent latch; 3-state

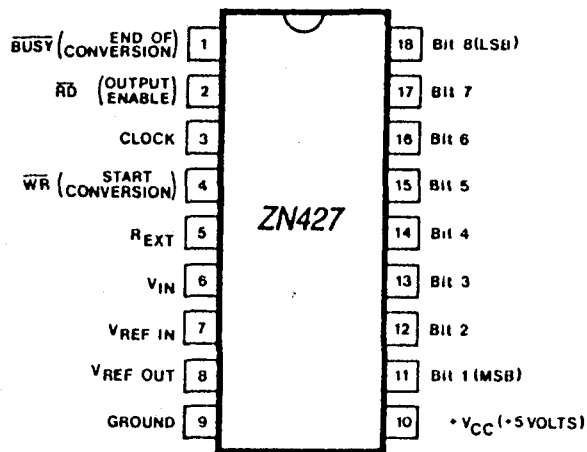
HC/HCT393



Dual 4-bit binary ripple counter

ZN427E 8-Bit A/D Converter

An 8-bit A/D converter with 3-state outputs to permit easy interfacing to a common data bus. The IC contains a voltage switching DAC, a fast comparator, successive approximation logic and a 2.56V precision voltage reference.



Characteristics (typical)

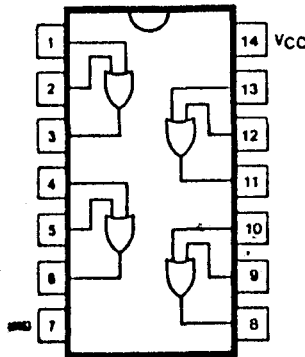
Supply voltage	4.5 to 5.5V
Max error	± 0.5 LSB
Conversion time	10 μ s
Clock frequency	1MHz
Supply current	25mA

OR GATES

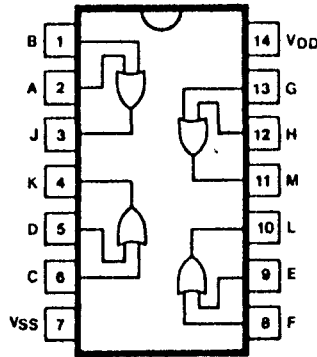
Quad 2-Input

Four 2-input OR gates in a single package available in standard TTL, LS, CMOS and HC types.

	7432	74LS32
High level output current (max)	-800 μ A	-400 μ A
Supply current avge per gate	4.75mA	1mA
Propagation delay low to high/ high to low 5V	10ns/14ns	14ns/14ns
	4071BE	74HC32
Propagation delay low to high/ high to low 5V	160ns	10ns
	10V	65ns
	15V	50ns

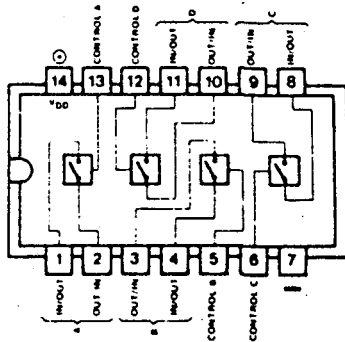


7432, 74LS32,
74HC32 Standard



4071BE Standard

CMOS



4016 CUADRUPLE CONMUTADOR BILATERAL

4066 CUADRUPLE CONMUTADOR BILATERAL

Los circuitos 4016 y 4066, están compuestos por 4 conmutadores para señales digitales y analógicas. Un nivel alto en la entrada de control, establece una conexión de pequeña impedancia bidireccional entre los terminales entrada/salida (condición ON). Con un nivel bajo en la entrada de control se desconecta el conmutador y la conexión entre los terminales de entrada/salida y salida/entrada pasa al estado de alta impedancia (condición OFF).

En algunas aplicaciones, la corriente enviada por el terminal VDD puede llegar a superponerse a la corriente conmutada y circulará a través de la resistencia de carga (RL). Para evitar esta corriente parásita cuando la corriente conmutadora entra por los terminales 1, 4, 8 ó 11, la diferencia de potencial entre la entrada y la salida de cada conmutador, no deberá exceder los 0,3 V. Ninguna corriente parásita circulará por la carga si la corriente conmutadora entra por los terminales 2, 3, 9 o 10.

Características eléctricas (Motorola) T = 25°C

(Typ.)	VDD	4016			4066			V	Conditions
		5	10	15	5	10	15		
'ON' Resistance		300	260	260	250	120	80	Ω	RL = 10 kΩ
Δ 'ON' Resistance between any two circuits in a common package		15			25	10	5	Ω	
Crosstalk between any two switches									RL = 1 kΩ Switch A ON Switch B OFF
20 log $\frac{V_{out}(B)}{V_{in}(A)}$ = -50 dB			1.25			8		MHz	
Frequency response (-3 dB)			40			65		MHz	RL = 10 kΩ Switch ON
Max. Control input pulse frequency		5	10	12	6	8	8.5	MHz	RL = 1 kΩ
Crosstalk Control input to signal output		30	50	100		300		mV	f = 1 kHz
Sine wave distortion			0.16			0.1		%	RL = 10 kΩ f = 1 kHz
Input current control			10			10		nA	

4013 BE

4013 DOBLE BIESTABLE D

El dato se tiene en cuenta cuando la entrada de reloj está a nivel bajo y se transfiere a la salida después del siguiente flanco de subida de la señal de reloj. Las entradas asincronas RESET (R) (puesta a cero) y SET (S) (puesta a uno), activas a nivel alto, son independientes y prioritarias sobre el dato (D) y sobre el reloj.

V _{DD}	5	10	15	V
Frecuencia máx. de reloj (MOTOROLA)	4	10	14	MHz

TABLA DE VERDAD (Motorola)

CLOCK	INPUTS			OUTPUTS		
	DATA	RESET	SET	Q	\bar{Q}	
	0	0	0	0	1	
	1	0	0	1	0	
	X	0	0	Q	\bar{Q}	No cambia
X	X	1	0	0	1	
X	X	0	1	1	0	
X	X	1	1	1	1	

X = Indiferente

