



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLÁN



FALLA DE ORIGEN

**“DISEÑO DE UN INTEGRADOR DE PICOS CROMATOGRAFICOS
BASADO EN EL MICROPROCESADOR Z80”**

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A:
JOSE AMANDO ORTIZ ISLAS

A S E S O R :
ING. JORGE BUENDIA GOMEZ



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FALLA DE ORIGEN

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
UNIDAD DE LA ADMINISTRACIÓN ESCOLAR
DEPARTAMENTO DE EXÁMENES PROFESIONALES

U. N. A. M.
FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLÁN

ASUNTO: VOTOS APROBATORIOS

DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLÁN
P R E S E N T E .

AT'N: Ing. Rafael Rodríguez Ceballos
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA:

"Diseño de un Integrador de picos cromatógraficos basado en el microprocesador 280"

que presenta el pasante: José Amanda Ortiz Islas
con número de cuenta: 8003627-2 para obtener el TITULO de:
Ingeniero Mecánico Electricista ; en colaboración con :

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .

"POR MI RAZA HABLARA EL ESPIRITU"

Cuatitlán Izcalli, Edo. de Méx., a 9 de Diciembre de 1991

PRESIDENTE Ing. José Luis Rivera López

VOCAL Ing. Ubaldo Ramírez Urizar

SECRETARIO Ing. Jorge Eusebio Gómez

PRIMER SUPLENTE Ing. Nicéida Calva Tania

SEGUNDO SUPLENTE Ing. Rogelio Ramos Carranza

AGRADECIMIENTOS:

AL ING. JORGE BUENDIA GOMEZ

Por su excelente asesoria.

**AL ING. CERECERO, JULIO, JORGE, CARLOS Y VICTOR.
COMPANEROS DEL I.M.P.**

Por la invaluable ayuda que me brindaron.

**A TODOS AQUELLOS QUE DE UNA U OTRA FORMA
CONTRIBUYERON EN MI FORMACION.**

A MIS PADRES:

**MARTIN ORTIZ ORTIZ
MARTHA ISLAS DE ORTIZ**

Por darme la oportunidad de ver la luz de este mundo.

A MI HIJA:

CITLALLI

Por ser la estrella que guía mi camino.

A MI COMPANERA:

LAURA

Por su cariño y paciencia.

A MIS TIOS:

VICTOR LOPEZ Y CARMEN ORTIZ

Por tenderme la mano cuando lo he necesitado.

I N D I C E

	Paq.
	CAPITULO 1
INTRODUCCION	1
	CAPITULO 2
EL INTEGRADOR	5
	CAPITULO 3
AMPLIFICADOR Y CONVERTIDOR	7
3.1 Primera Etapa De Amplificación	
3.2 Segunda Etapa De Amplificación	
3.3 Conversión Analógico/Digital	
	CAPITULO 4
CPU Y MEMORIA	20
4.1 El Microprocesador	
4.2 La Memoria	
	CAPITULO 5
ETAPA DE ENTRADA/SALIDA	29
5.1 Introducción	
5.2 El Puerto 8255	
5.3 Programación Del Puerto	
5.4 La Interface De Impresión	
5.5 Comunicación Con El Usuario	
	CAPITULO 6
LA PROGRAMACION	41
6.1 Introducción	
6.2 Inicialización del sistema	
6.3 La Adquisición De Datos	
6.3.1 Inicio y Desarrollo de La Corrida	

6.3.2	Conteo de Tiempo Real	
6.3.3	El Cromatograma	
6.3.4	Fin de la Corrida	
6.4	Los Cálculos	
6.4.1	Introducción	
6.4.2	Integración de los Picos Cromatográficos	
6.4.3	Area Total y % Area	
6.5	Impresión Del Reporte	
6.5.1	Introducción	
6.5.2	Conversión HEX-DEC-ASCII	
6.5.3	Impresión de Resultados	

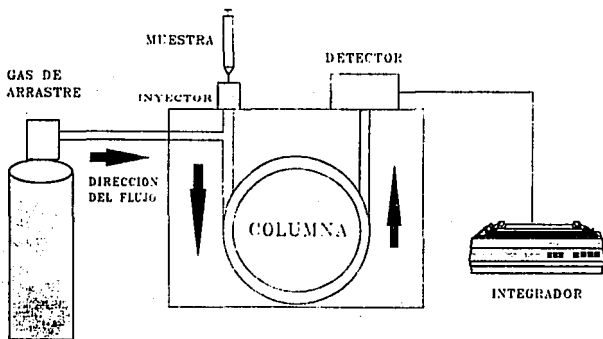
	CAPITULO 7	
EL PROGRAMA ENSAMBLADO		64
	CAPITULO 8	
OPERACION DEL INTEGRADOR		82
CONCLUSIONES		86
APENDICES		88
BIBLIOGRAFIA		98

FALLA DE ORIGEN

CAPITULO 1

INTRODUCCION

La cromatografía es uno de los métodos más ampliamente utilizados para la separación e identificación de los distintos componentes de una mezcla, y esta basada en la adsorción de esos componentes con respecto a un dispositivo llamado columna, la cual los separa y los hace llegar hasta la sección conocida como el Detector; hay diversas clases de detectores, pero en general todos tienen la característica de entregar una señal eléctrica proporcional a la cantidad de componentes que forman la mezcla.



ANALISIS CROMATOGRAFICO

FIGURA 1.1

En la figura 1.1 se muestra un sistema de análisis cromatográfico clásico. Existen varios tipos de cromatógrafos, pero en general todos constan de tres secciones principales: el puerto de inyección o INYECTOR, la COLUMNA y el DETECTOR. Por el inyector se introduce la muestra a analizar la cual es arrastrada

por un flujo de gas hasta la Columna donde son separados sus distintos componentes, estos son llevados por el mismo flujo de gas hasta el Detector, el cual entrega una señal eléctrica proporcional al perfil de los distintos componentes. Al flujo que arrastra la muestra se le denomina comunmente "carrier" o gas de arrastre. Las características del Detector y la Columna que se utilizan dependen del tipo de muestra y de los componentes que quieran ser identificados o cuantificados.

En la figura 1.2 se muestra un cromatograma típico, en el cual se observan las dos principales características de la cromatografía: se pueden hacer análisis cuantitativos y cualitativos, pues se basa en la velocidad con que los componentes, ya separados, llegan al Detector, obteniéndose básicamente dos tipos de información, las AREAS y los TIEMPOS DE RETENCION de cada pico.

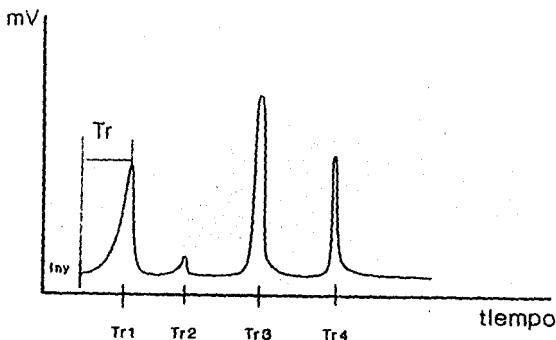


FIGURA 1.2

Un cromatograma es un registro gráfico de los picos correspondientes a los componentes ya separados con su respectivo Tiempo de Retención. Como se puede ver el cromatograma es una gráfica de tiempo vs. respuesta del detector, de este modo,

cuando los componentes son separados este nos muestra un perfil de la concentración de cada uno de ellos, y cuando por el detector solo esta pasando gas de arrastre se obtiene una línea idealmente horizontal y recta, la cual es llamada LINEA BASE. En la figura 1.2 se puede apreciar que el Tiempo de Retención T_r , es el tiempo transcurrido desde el momento en que la muestra es inyectada hasta que el pico alcanza su punto máximo; a cada pico le corresponde una vez establecidas las condiciones del análisis, solo un Tiempo de Retención característico de algún elemento ó componente.

Como se podrá ver toda la información anterior es muy importante, por lo cual normalmente se recurre a un equipo complementario capaz de procesar toda la información obtenida, el cual es conocido comunmente como INTEGRADOR, ya que se encarga de obtener el área de cada pico, aplicando métodos numéricos para hacer integración y así resolver el área de cada pico a partir de la señal eléctrica que entrega el cromatógrafo, el cual también se encarga de filtrarla y amplificarla adecuadamente.

En años atrás la integración se hacía utilizando métodos un tanto cuanto burdos, lentos y no muy precisos, tales como recortar cada pico del cromatograma y pesarlo, suponiendo que la densidad o peso específico del papel era uniforme; o semejando cada pico a un triángulo y midiendo sus dimensiones para así calcular su área. Poco después surgieron los integradores de disco, los cuales eran básicamente un registrador con un dispositivo adicional que graficaba por medio de cuentas o ciclos el área de cada pico.

En la actualidad existen una gran variedad de marcas y modelos de integradores individuales o de redes de integración, los cuales procesan toda la información más rápida y fielmente, pues utilizan todos los recursos electrónicos modernos tales como los convertidores analógico-digital y digital-analógico, memorias, microprocesadores, amplificadores operacionales, etc.

lo cual los hace más versátiles y poderosos, pero también de un costo elevado.

Muchos de los integradores actuales son caros y están sobrados para las necesidades de aquellos laboratorios que no necesitan algo tan sofisticado, por lo que en este trabajo de tesis se hace la propuesta de un integrador sencillo para cubrir esas necesidades.

EL INTEGRADOR

El integrador que aquí se pone a consideración, esta basado en el microprocesador de Zilog, el Z80, el cual tiene bastante tiempo en el mercado, pero aun sigue siendo adecuado para este tipo de trabajos, tanto que en la actualidad las marcas comerciales lo siguen utilizando en sus modelos más recientes.

La función básica de un integrador es que a partir de una señal eléctrica de millivoltaje, pueda aplicar métodos numéricos para obtener el área de cada uno de los llamados picos, además de manipular esos datos para brindar otra información también valiosa como los porcentajes de área, el área total y los tiempos de retención. Para poder realizar esas tareas, el integrador necesita ser básicamente un sistema de adquisición de datos, por lo cual debe estar provisto de una gran capacidad de memoria RAM. El integrador aquí propuesto consta de un sistema mínimo (CPU, memoria RAM, ROM y puertos), un convertidor analógico-digital y una interface para la impresión de resultados en una impresora con entrada paralelo tipo Centronics. De esta forma se aprovecharía que en la actualidad todos los laboratorios cuentan con una computadora así como con su respectiva impresora, la cual podría utilizarse para reproducir los resultados, ahorrándose de esta forma, el costo de un integrador con su impresora, ya que estos cuentan con inconvenientes costosos tales como papel y cartuchos de tinta especiales y únicos.

En la figura 2.1 se muestra un diagrama a bloques de lo que sería el integrador. La señal proveniente del cromatógrafo es filtrada y adecuada a un nivel conveniente en la etapa de amplificación para que después, el convertidor analógico-digital la digitalize; la señal digitalizada entra al sistema del microprocesador a través del puerto de entrada/salida para ser

procesada por el CPU y/o almacenada en la memoria RAM en base a la programación almacenada en la memoria ROM; la interface de impresión será utilizada para que a través de ella el GPU se comunique con la impresora para reproducir el reporte de cada corrida. Los controles e indicadores son el medio por el cual el usuario se comunica con el integrador, para indicarle la acción que debe realizar a través de dos teclas (INICIO/FIN y ABORTAR) ó para saber en que estado se encuentra (LISTO ó ADQUIRIENDO DATOS) dependiendo de cual indicador este encendido.

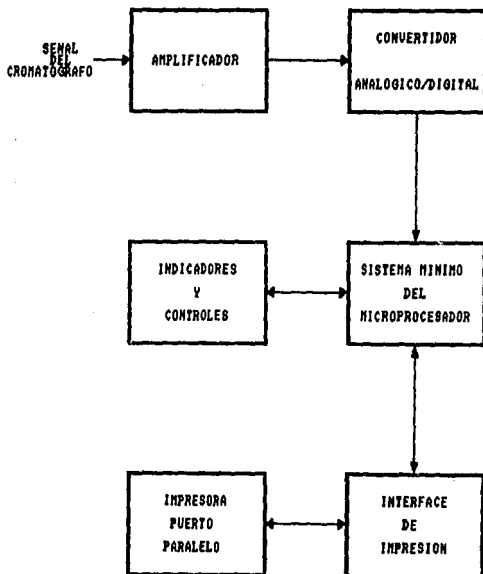


Figura 2.1

AMPLIFICADOR Y CONVERTIDOR A/D

3.1 PRIMERA ETAPA DE AMPLIFICACION

La etapa de amplificación esta formada de dos partes, implementadas con un amplificador operacional cada una, para la primera etapa se utiliza un LF356N en una configuración de ganancia unitaria, este amplificador operacional es utilizado básicamente por su característica de entrada FET de alta impedancia, para evitar interacciones e incompatibilidad con los distintos detectores y cromatógrafos que existen. En la figura 3.1 se puede apreciar la configuración con la cual se utiliza el LF356, para obtener una ganancia unitaria y de polaridad inversa.

Para estas condiciones el fabricante recomienda:

$$R1 = \frac{R2}{4} ,$$

si $R2 = 4.7K\Omega$, entonces $R1 = 1.2k\Omega$;

como de datos del fabricante se tiene que:

$$R1C \geq \frac{1}{2\pi(5MHz)}$$

entonces:

$$C \geq \frac{1}{2\pi R1(5MHz)}$$

Si $R1 = 1.2k\Omega$, se obtiene que:

$$C \geq \frac{1}{2\pi R1(1.2k)(5M)} = 26.5pF$$

$C \geq 26.5pF$, acercandolo a un valor comercial :

$C = 27pF$ y por lo tanto se obtendrá, $AV = -1$

La ganancia de esta etapa es unitaria ya que su función

principal es servir como acopladora de impedancias. El potenciómetro R6 sirve para ajustar el balance del amplificador operacional, es decir, con un voltaje de entrada igual a cero Volts, se debe obtener un voltaje a la salida del AMP OP, también de cero Volts, si no es así, el potenciómetro se ajusta hasta obtenerlo. R1 y C forman un filtro pasa-bajas, con el fin de eliminar ruido de frecuencias superiores a 5 Mhz.

Por otro lado, la señal proveniente del cromatógrafo es conectada directamente a la entrada de esta etapa, es decir, el voltaje V_{en} , es el voltaje que el cromatógrafo entrega. A la salida de esta etapa se obtiene el mismo voltaje de entrada en magnitud, pero con polaridad contraria.

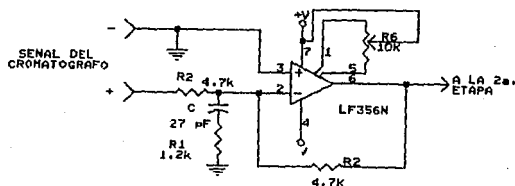


Figura 3.1 PRIMERA ETAPA

3.2 SEGUNDA ETAPA DE AMPLIFICACION

Para esta etapa, como se aprecia en la figura 3.2, se utiliza un dispositivo con un nivel bajo de offset de entrada, debido a que los niveles de señal que se manejarán serán bajos (0-2.5 Volts). El dispositivo que cumple perfectamente estas condiciones es el LM607 de NATIONAL ó el OP07 de PRECISION MONOLITHICS, ya que por especificaciones de fabrica, son AMP OP de precisión, confiables y con un voltaje de offset de entrada de $25\mu V$.

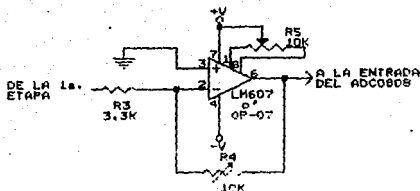


Figura 3.2 SEGUNDA ETAPA

Todos los cromatógrafos tienen una salida para integrador con una amplitud máxima de 1 Volt, por lo tanto, el voltaje máximo de entrada a esta etapa será ese mismo, solo que con polaridad contraria, debido a que en la primera etapa se obtiene una ganancia de -1 . Por otro lado, el voltaje máximo que el convertidor A/D puede resolver, es su voltaje de referencia, el cual, como veremos más adelante es de 2.5 Volts, entonces la configuración de este segundo operacional deberá tener una ganancia de -2.5 , para que cuando a la entrada se tenga el voltaje máximo posible (1 Volt), a la salida se obtenga el voltaje máximo que puede resolver el convertidor A/D, de esta forma:

$$(V_{en_{max}}) \times (AV_{TOT}) = V_{ref};$$

entonces si:

$$AV_{TOT} = AV_1 AV_2;$$

en este caso:

$$AV_1 = -1, \quad AV_2 = -2.5, \quad \text{por lo tanto:}$$

$$AV_{TOT} = -1(-2.5) = 2.5;$$

de esta manera:

$$V_{en_{max}} = \frac{V_{ref}}{AV_{TOT}} = \frac{2.5V}{2.5} = 1 \text{ Volt}$$

El voltaje mínimo que el convertidor puede resolver es el voltaje de referencia dividido por el número de pasos del convertidor, los cuales en este caso son de 2.5 V, y 256 respectivamente, por lo que el voltaje mínimo que el convertidor resuelve será:

$$\frac{V_{ref}}{256} = \frac{2.5V}{256} = 9.76 \text{ mV}$$

Si se divide este voltaje entre la ganancia total se obtendrá el voltaje mínimo que el integrador detectará:

$$\frac{9.76\text{mV}}{2.5} = 3.9 \text{ mV} \approx 4 \text{ mV}$$

Para esta etapa, si $R3 = 3.3k\Omega$, entonces $R4 \approx 10k\Omega$ y por especificaciones del fabricante $R5 = 10k\Omega$.

El ajuste $R5$ sirve para balancear el circuito, es decir, para que cuando la señal de entrada al amplificador sea cero, la salida de este también sea nula. Por otro lado el potenciómetro $R4$ sirve para ajustar la ganancia de la segunda etapa y en general de todo el amplificador. Los ajustes deberán hacerse, primero el de balanceo y después el de ganancia.

3.3 CONVERSION ANALOGICA/DIGITAL

La función de un convertidor A/D, es producir una palabra digital, la cual represente la magnitud de alguna señal analógica, ya sea voltaje ó corriente. Para esta etapa es necesario utilizar un convertidor A/D, con el fin de transformar la señal analógica proveniente del cromatógrafo, en este caso voltaje, en palabras digitales, para que el sistema del microprocesador pueda procesarla, ya que de otra forma no sería posible, pues el lenguaje del microprocesador es puramente digital.

Existe un amplio número de circuitos para implementar convertidores A/D, pero los más populares son:

- Tipo Paralelo
- De Integración
- De Aproximaciones Sucesivas

cada uno con características que los hacen ser más adecuados que otros para una aplicación específica, por su velocidad, exactitud, costo, tamaño y versatilidad.

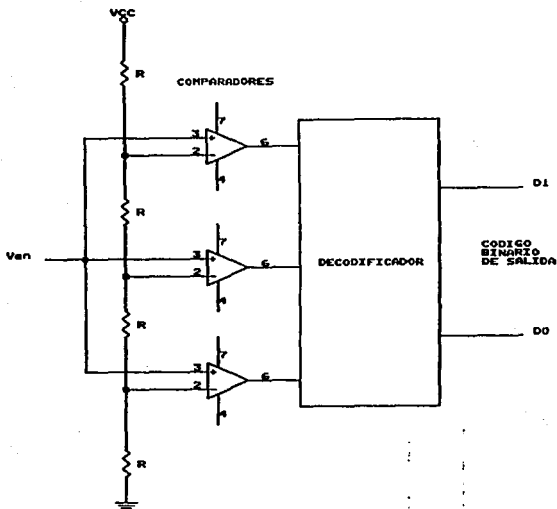


Figura 3.3 Conversión A/D Paralelo

En la conversión A/D del tipo paralelo, como se muestra en la figura 3.3, se utilizan comparadores en paralelo. Un divisor de voltaje sirve para poner los voltajes de referencia en las entradas inversoras de cada comparador. El voltaje con que el divisor es alimentado, representa el valor de escala total del

convertidor. El voltaje que va a ser convertido es aplicado a las entradas no inversoras de todos los comparadores a la vez, es decir, en paralelo. Si el voltaje de entrada en un comparador es más grande que su voltaje de referencia de su entrada inversora, la salida del comparador tomará un estado lógico alto. De esta forma, las salidas de los comparadores proporcionan una representación digital del nivel de voltaje de la señal de entrada.

La mayor ventaja de un convertidor A/D del tipo paralelo, es su velocidad de conversión, la cual es simplemente el tiempo de propagación de los comparadores, por lo que también se le conoce como conversión A/D tipo *Flash*. Por otra parte el código de salida de los comparadores no es un código binario estándar, pero este puede ser convertido con alguna lógica simple. La mayor desventaja de un convertidor A/D tipo flash es el número de comparadores necesarios para producir un resultado con una resolución razonable. Un convertidor A/D de dos bits, como el de la figura 3.3 necesita de tres comparadores. Para implementar un comparador de n bits de resolución se necesita de $(2^n - 1)$ comparadores. Convertidores de este tipo están disponibles a un costo relativamente alto.

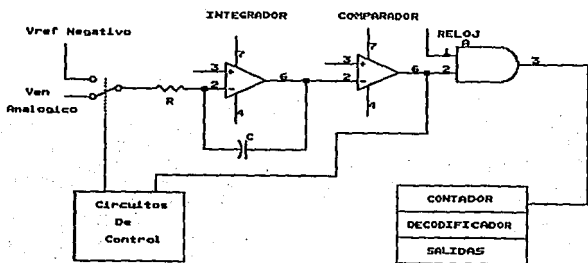


Figura 3.4 Convertido A/D tipo INTEGRACION

Otra forma de conversión A/D ampliamente difundida es la del tipo de *INTEGRACION*, de la cual, la más utilizada es la de doble pendiente (dual-slope). Este tipo de convertidor es frecuentemente utilizado como el corazón de voltímetros digitales, debido a que pueden dar un número grande de bits de resolución a bajo costo. En la figura 3.4 se muestra un circuito clásico de un convertidor de este tipo. Al iniciar la conversión, el voltaje de entrada que va a ser convertido, es aplicado por medio de un switch de control a un circuito integrador cuyo periodo esta dado por RC. Este periodo normalmente es de pocos milisegundos. Si se asume que el voltaje de entrada es positivo, entonces este provocará en la salida del integrador una rampa negativa, tal como se muestra en la figura 3.5. En cuanto la salida del integrador alcanza un nivel de unos cuantos microvolts abajo del nivel de tierra, la salida del comparador toma un nivel lógico alto, habilitando a la compuerta AND y permitiendo que la señal de reloj entre en el arreglo del contador. Después de un número fijo de pulsos de reloj, la circuiteria de control conmuta la entrada del integrador a un voltaje de referencia negativo, esto provocará ahora una rampa positiva en la salida del integrador, la cual al cruzar por cero volts causará que la salida del comparador tome un nivel lógico bajo, deshabilitando la compuerta AND y cortando la señal de reloj al contador. El número de pulsos transcurridos hasta que la salida del integrador llega a cero es directamente proporcional al voltaje de entrada. Este tipo de convertidores son exactos, baratos y proporcionan protección contra ruido, pero su principal desventaja es su baja velocidad de conversión, pues un convertidor de 4 1/2 dígitos se toma 300mS para hacer una conversión.

Otros convertidores A/D de Integración, utilizan simple, triple y cuádruple pendiente. La primera mejora la velocidad de conversión, pero reduce la exactitud; mientras que los de triple y cuádruple pendiente mejoran la exactitud pero reducen la velocidad de conversión.

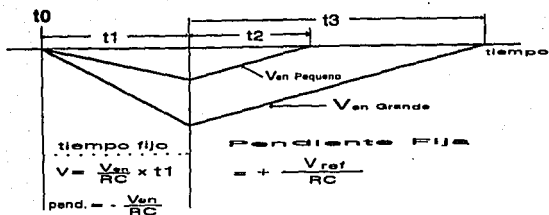


Figura 3.5 Rampa Dual

El otro tipo de convertidores es el de **APROXIMACIONES SUCESIVAS**, el cual involucra n operaciones de comparación para un convertidor de n bits; cada comparación resulta en la determinación de un valor binario (1 ó 0) de un bit. El primer bit comparado es el más significativo y el último es el menos significativo.

En la figura 3.6 se muestra un diagrama que sirve para ilustrar el funcionamiento de este tipo de convertidores. La parte principal del convertidor es un Registro de Aproximaciones Sucesivas (SAR), el cual, después de que se le indica, por medio de la señal SC (Start Conversion), que inicie la conversión, al primer pulso de reloj pone un 1 en el bit más significativo, lo que provoca que el Convertidor Digital/Analógico (DAC) ponga en la entrada del comparador la mitad del voltaje de referencia. Si el voltaje de entrada es más bajo que el entregado por el DAC, el comparador pondrá un 0 lógico en su salida, indicándole al SAR que ponga a 0 el bit, porque el voltaje de entrada es menor a la mitad del voltaje de referencia. Si por el contrario, el voltaje de entrada es más grande, entonces el comparador pondrá un 1 en su salida indicándole al SAR que mantenga el bit con un 1. Al siguiente pulso de reloj, el SAR pone un 1 también en el siguiente bit en orden descendente y el DAC entregará una parte

proporcional del voltaje de referencia de $3/4$ ó $1/4$ dependiendo del estado del primer bit convertido, realizándose, de esta manera la misma rutina que para el bit anterior hasta completar los n bits, en este caso 8. De esta forma se pone en claro que para una conversión A/D de 8 bits se necesitan 8 pulsos de reloj.

Al completarse la conversión el SAR manda una señal que indica el fin de esta (EOC), la cual dispara el dato a la salida del Latch. En algunos casos en lugar del DAC se utiliza un arreglo de resistencias y switches para poner el voltaje de referencia respectivo en la entrada del comparador.

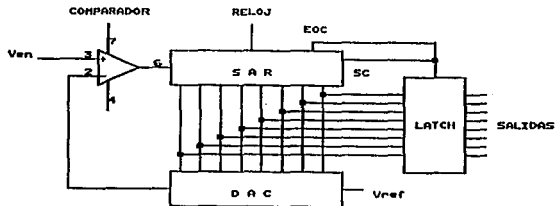


Figura 3.6 Convertidor A/D por Aproximaciones Sucesivas

En esta etapa, para la conversión analógica-digital, se utiliza el convertidor ADC0808 de National, este dispositivo tiene una resolución de 8 bits, utiliza solo una fuente de 5 Volts, su tiempo de conversión típico es de $100 \mu\text{s}$, tiene capacidad para manejar 8 entradas analógicas multiplexadas y es fabricado con tecnología CMOS.

El ADC0808 utiliza el método de aproximaciones sucesivas, el cual es ampliamente utilizado, especialmente para interfasar con computadoras dado que tiene ambas características,

alta resolución (hasta 16 bits) y alta velocidad. El tiempo de conversión es fijo e independiente de la magnitud del voltaje de entrada. Cada conversión es única e independiente de los resultados de la conversión previa, porque la lógica interna es limpiada en el inicio de la conversión. Los convertidores actuales incluyen salidas de datos de tres estados y señales de control para interfazarlos fácilmente con microprocesadores.

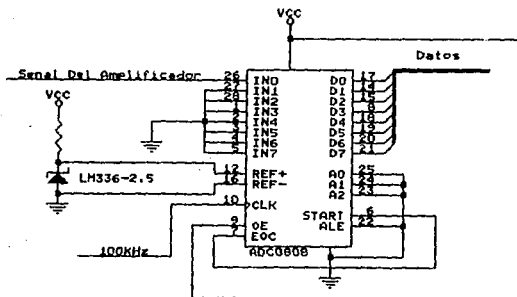


Figura 3.7 El ADC0808

En el ADC0808 no es necesario ajustar el cero ni la escala total de el convertidor, tal como sucede en muchos otros casos. En la figura 3.7 se muestra la configuración con la cual se utilizará el convertidor. Como se puede observar, solo se utiliza una de las 8 entradas analógicas, en este caso, la entrada 0 correspondiente al pin 26, pero puede ser utilizada cualquiera de las 8 disponibles para alimentar la señal analógica proveniente del amplificador de entrada. Como las entradas analógicas del convertidor son multiplexadas o direccionadas por las señales ADD A, ADD B y ADD C en los pines 23, 24 y 25 respectivamente, en

este caso la entrada 0 es direccionada conectando las tres directamente a tierra, junto con las 7 entradas analógicas sobrantes. Los pines 6 y 7 (START y EOG) están conectados entre sí para una conversión continua, pues la señal START (Inicio de conversión) es activada por la señal EOG (Fin de conversión); en este modo de funcionamiento del convertidor, es necesario aplicar un pulso bajo externo para iniciar la conversión después de que el sistema ha sido energizado, esto se logra conectando el switch INICIO/FIN al pin de control de un 74LS125 para que mande un nivel bajo durante el tiempo que el switch dure oprimido, siendo de utilidad solo la primera vez que se oprime después de haberse energizado y no interfiere en la conversión en las siguientes ocasiones.

El voltaje de referencia a través de los pines 12 y 16, se obtiene conectando entre ellos un diodo Zener de referencia, el LM336-2.5, el cual se caracteriza por su voltaje de 2.5 Volts, gran estabilidad por variaciones de temperatura y un $\pm 1\%$ de tolerancia de 0 a 2.5 Volts; este rango del voltaje de referencia, servirá para poder obtener una digitalización correcta. De este modo los voltajes inferiores a 0 Volts y superiores a 2.5 Volts, provocarán un dato en las salidas D0-D7, pero este dato no podrá ser tomado como correcto, sin embargo los voltajes entre 0 y 2.5 Volts provocarán una salida digital proporcional a la entrada analógica.

Por otro lado la señal ALE (Address Line Enable) es decir, la habilitación de las líneas de direcciones, se conecta a tierra para tenerla habilitada todo el tiempo, lo mismo que la señal OE (Output Enable) o habilitación de salidas, solo que esta es conectada directamente a 5 Volts.

La señal de reloj entra por el pin 10, en este caso es de 100 khz y proviene de un oscilador implementado con un temporizador 555 en su modo astable. El convertidor ocupa 64 pulsos de reloj para realizar una conversión, de esta forma cada 0.64 mS se llevará a cabo una conversión.

Finalmente como se mencionó con anterioridad, la salida digital se obtiene entre D0-D7 y es proporcional al voltaje analógico en la entrada del convertidor, la relación matemática que explica esto es:

$$\frac{V_{EN}}{V_{ET} - V_Z} = \frac{D_X}{D_{MAX} - D_{MIN}}$$

donde para este caso:

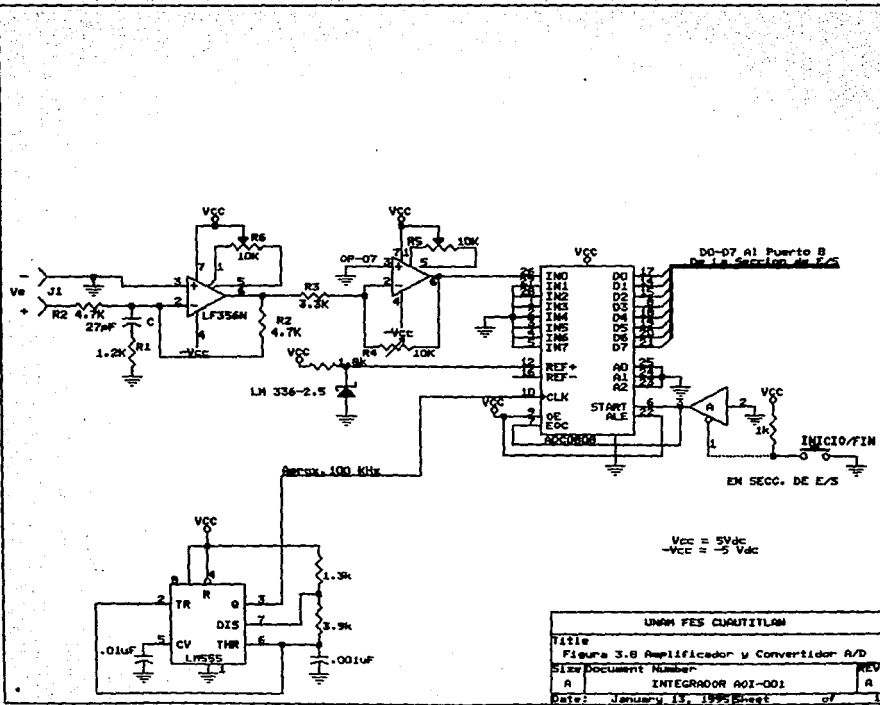
- V_{EN} = Voltaje de entrada al ADG
- V_{ET} = Voltaje a escala total
- V_Z = Voltaje de cero
- D_X = Dato que esta siendo leído
- D_{MAX} = Dato máximo obtenible = 255
- D_{MIN} = Dato mínimo obtenible = 0

por lo tanto:

$$V_{EN} = V_{ET} \frac{D_X}{D_{MAX}}$$

es decir, el ADQ0808 utiliza la llamada conversión proporcional (Ratiometric Conversion), la cuál consiste en que el convertidor tiene un pin para conectarle un voltaje de referencia externo, logrando de esta forma que la salida digital represente una proporción de la entrada analógica, al voltaje de referencia, tal como si fuera un divisor analógico con salida digital.

En el diagrama de la figura 3.8 se muestra la interconexión de todas las etapas mencionadas en este capítulo.



CPU Y MEMORIA

4.1 EL MICROPROCESADOR

El microprocesador que se utiliza es el ya conocido Z80, un dispositivo de 8 bits. El Z80 solo necesita una fuente de 5 Volts como alimentación. Tiene un bus de direcciones de 16 Bits (A0-A15), con salidas de tercer estado y activadas con un nivel lógico alto. Este Bus proporciona el direccionamiento para la memoria (hasta 64 kbytes), y para los puertos de entrada/salida también. Para el direccionamiento de puertos solo se utilizan los 8 Bits menos significativos del Bus, lo que permite al usuario direccionar solamente 256 puertos en forma directa.

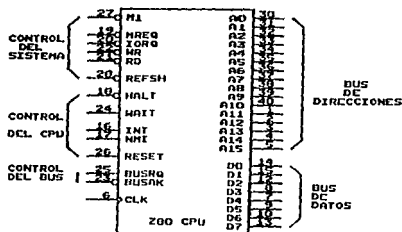


Figura 4.1 Configuración del Z80

En la figura 4.1 se muestra la configuración del Z80 y como se puede ver, los datos que maneja son de 8 Bits (D0-D7), los cuales fluyen a través del Bus de datos. Este Bus tiene salidas de tercer estado y también se activan con un nivel lógico alto.

El Bus es utilizado para el intercambio de datos con la memoria y los puertos.

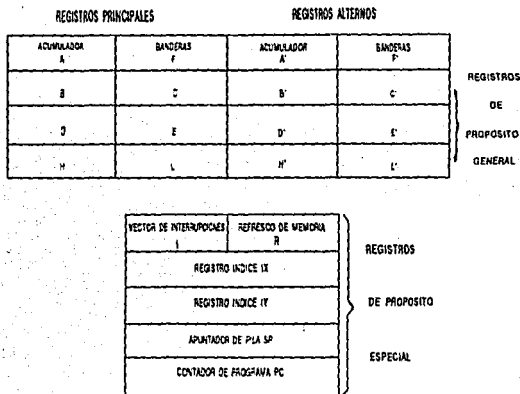


Figura 4.2 Registros del Z80

Como se puede observar en la figura 4.2, este microprocesador cuenta con 208 bits de memoria de LECTURA/ESCRITURA implementados con memoria RAM estática accesibles al usuario. Esta memoria esta configurada en 18 registros de 8 bits cada uno y 4 de 16 bits. Estos se agrupan en 2 bloques de 6 registros de propósito general, los cuales pueden ser utilizados como registros individuales de 8 bits o en registros pares de 16 bits; tambien hay dos bloques de los registros acumulador y de banderas. Los registros de propósito especial son:

- Contador de programa (PC)
- Registros indices (IX, IY)

Apuntador de pila (SP)
Registro de interrupciones (I)
Refresco de memoria (R)

Los registros de propósito general pueden ser usados en pares o en forma individual y son:

PRINCIPALES (BC, DE, HL)
ALTERNOS (BC', DE', HL')

Por último están el Acumulador y el Registro de Banderas:
PRINCIPALES (A, F)
ALTERNOS (A', F')

El Z80 puede ejecutar 158 diferentes tipos de instrucciones, desde tipo aritmético y lógico, hasta saltos condicionales, entre otras.

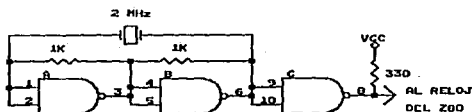


Figura 4.3 Reloj

En el diagrama de la figura 4.3 se muestra el circuito de reloj que utiliza el microprocesador; en este caso trabaja a una velocidad de 2Mhz y, aunque tiene la capacidad para trabajar a velocidades mayores, los 2Mhz son suficientes para satisfacer las necesidades del integrador; esta señal de reloj es obtenida de un arreglo con un cristal tal como se puede observar.

El integrador cuenta con un circuito clásico para resetear al microprocesador ya sea manual, ó automáticamente, al ser energizado. El diagrama de la figura 4.4 muestra el circuito de

RESET. Esta señal de RESET tambien se utiliza en otros dispositivos del integrador que necesitan de ella.

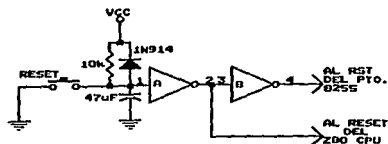


Figura 4.4 RESET

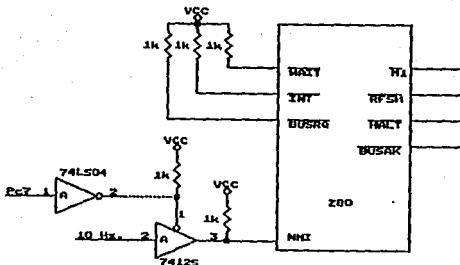


Figura 4.5 Señales de Control

En la figura 4.5 se muestra como las señales de control WAIT, INT y BUSRQ, son conectadas a Vcc a través de unas resistencias de pull-up, para que no puedan tomar otro nivel, ya que no serán utilizadas. Las salidas del Z80, MI, RFSH, HALT y BUSAK, tampoco serán utilizadas, por lo que

quedan libres. Las señales \overline{RD} y \overline{WR} , además de ser útiles en el direccionamiento de la memoria, serán utilizadas también en la sección de ENTRADA/SALIDA para controlar el puerto, lo mismo que la señal \overline{IORQ} .

La señal de entrada al Z80 llamada NMI (No Maskable Interruption), es utilizada como señal de conteo de tiempo real, dado que, si por este pin se alimenta una señal periódica y regular, se puede realizar ese conteo, utilizando la programación adecuada. Debido a que cuando se le solicita una interrupción no mascarable al Z80, el contador de programa es cargado automáticamente con la dirección 0066H, localidad donde se marca un salto a cierta subrutina, la cual se encarga de incrementar un contador para contabilizar el número de interrupciones realizadas e indirectamente el tiempo transcurrido.

4.2 LA MEMORIA

El tamaño de la memoria que se utiliza es de 12 kbytes dividida en 4k de ROM, obtenida de una memoria EPROM 2732, y 4k de RAM implementada con 4 memorias 4116 con capacidad de 2k x 8 cada una. La capacidad de la memoria RAM es grande debido a que el integrador es en realidad un sistema de adquisición de datos, por lo que requiere de una gran capacidad de almacenamiento.

La ROM del sistema de integración se encuentra a partir de la localidad 0000 y hasta la 0FFF y en ella se almacena toda la programación necesaria para poder operar el circuito como un integrador y, como se verá después, incluye el programa principal, las subrutinas de utilidad y las tablas.

La RAM se encuentra desde la localidad 1000 hasta la 2FFF, es decir, 8 kbytes necesarios para almacenar los datos adquiridos durante una corrida, además de guardar los resultados de las operaciones realizadas.

ENTRADAS				SALIDAS					
MRQ	A13	A12	A11	0	1	2	3	4	5
0	0	0	0	0	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1
0	0	1	0	1	1	0	1	1	1
0	0	1	1	1	1	1	0	1	1
0	1	0	0	1	1	1	1	0	1
0	1	0	1	1	1	1	1	1	0

Tabla 4.1

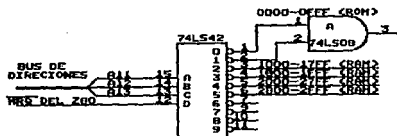


Figura 4.6 Codificación de la Memoria

La memoria es accesada por medio de un circuito integrado decodificador 74LS42, tal como se muestra en la figura 4.6, el cual utiliza las direcciones A11, A12 y A13 además de la señal MREQ para decodificar las señales de control \overline{CE} (Chip Enable) de cada memoria. Para acceder desde la dirección 0000 hasta la 0FFF se requiere utilizar las salidas 0 y 1 del decodificador como entradas de una compuerta AND la cual activara la señal \overline{CE} de la ROM cuando alguna de esas salidas sea decodificada. Las señales \overline{CE} de cada RAM son controladas por las salidas 2 (1000-17FF), 3 (1800-1FFF), 4 (2000-27FF) y 5 (2800-2FFF) del decodificador dependiendo de la dirección que se quiera acceder, como se muestra en la tabla 4.1.

En la figura 4.7 se muestra como, para controlar la señal \overline{OE} (Output Enable) de cada bloque de memoria, se utilizan las

señales \overline{MREQ} y \overline{RD} provenientes del Z80 de tal forma que cuando se requiera leer alguna localidad de memoria, las señales \overline{MREQ} y \overline{RD} son aplicadas a las entradas de una compuerta OR, dando como resultado una señal llamada \overline{MREAD} la cual es conectada directamente al pin de control OE de cada integrado.

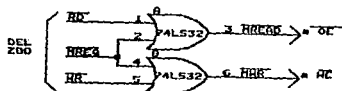
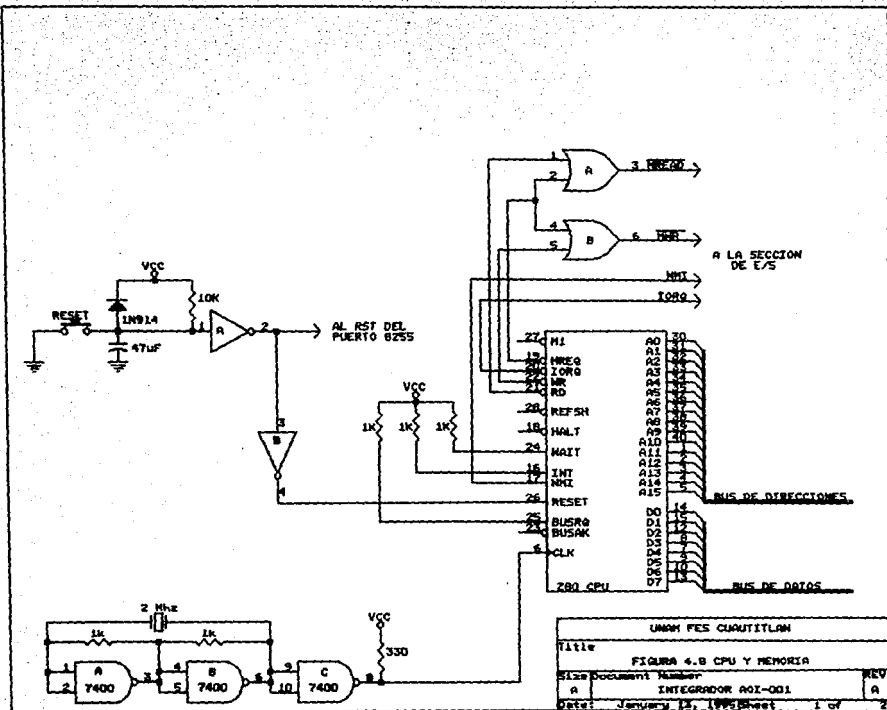
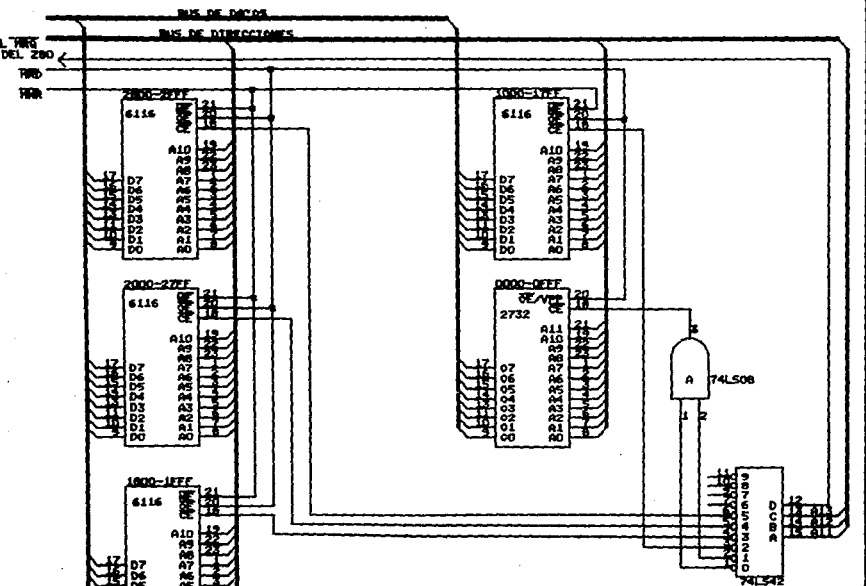


Figura 4.7 Señales \overline{OE} y \overline{WE}

Por otro lado, en forma análoga, cuando se requiere escribir en alguna localidad de la memoria RAM se utilizan las señales \overline{MREQ} y \overline{WR} , provenientes del microprocesador, en las entradas de una compuerta OR, la cual entregará a la salida una señal llamada \overline{MWR} , la cual es conectada directamente al pin \overline{WE} de los integrados RAM.

En la figura 4.8 se muestra el diagrama completo de la interconexión del microprocesador con el arreglo de memoria y sus respectivas señales de control.





UNIR PES CIRCUITLAN		
Title	FIGURA 4.8 CPU Y MEMORIA	
Size Document Number	A	
REV	A	
Date:	January 13, 1995	Sheet 2 of 2

ETAPA DE ENTRADA / SALIDA

5.1 INTRODUCCION

Esta sección trata sobre la comunicación entre el microprocesador y los dispositivos externos, con los que intercambia información; el dispositivo alrededor del cual gira todo ese intercambio, un PPI (Programmable Peripheral Interface), es el Circuito Integrado 8255, el cual tiene tres puertos de 8 bits cada uno (Puerto A, Puerto B y Puerto C) y pueden ser habilitados para funcionar como entrada ó salida de datos (A y B), ó como ambas (C). Los tres puertos son independientes entre sí y el puerto C tiene la peculiaridad de estar dividido en dos bloques de 4 bits cada uno, PC0-PC3 y PC4-PC7 respectivamente, también independientes entre sí, teniéndose de esta manera la posibilidad de programar a ambos como salidas, entradas, o como una combinación de ellas, tal como se muestra en la figura 5.1.

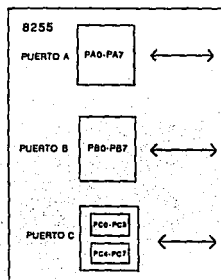


Figura 5.1 Puertos Del 8255

Existe también un grupo de señales de control que servirán para la comunicación entre el microprocesador y el puerto, es decir, con estas señales se pueden ejecutar operaciones de entrada/salida entre el Z80 y el 8255.

5.2 EL PUERTO 8255

En la figura 5.2 se puede ver la configuración del 8255, el cual tiene dos entradas llamadas A0 y A1 las cuales sirven para codificar únicamente cuatro direcciones, las de los puertos A, B y C, y la dirección del puerto por donde se introduce la palabra de control. Esas entradas están conectadas directamente a las líneas A0 y A1 del bus de direcciones.

La entrada \overline{CS} (Chip Select) habilita al 8255 para que el microprocesador pueda leer o escribir en él. En este caso, como es el único dispositivo de entrada/salida que hay, la señal \overline{IORQ} proveniente del microprocesador se conecta directamente a esa entrada.

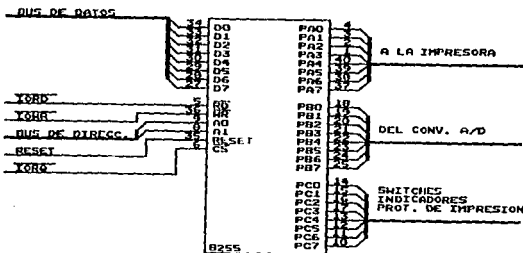


Figura 5.2 Configuración Del 8255

Por último están las entradas \overline{WR} y \overline{RD} las cuales le indican

al 8255, si se va a escribir un dato en él, desde el bus, por medio de la señal \overline{WR} , o si el PPI tiene que poner un dato al bus, con la señal \overline{RD} . A estas entradas se conectan las señales \overline{IOWR} y \overline{IORD} respectivamente, las cuales se obtienen combinando las señales \overline{RD} , \overline{WR} e \overline{IORQ} , provenientes del microprocesador, a través de unas compuertas OR, tal como se muestra en la figura 5.3.

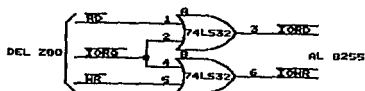


Figura 5.3 Señales IORD e IOWR

La señal de RESET, en este caso, es activada con un nivel lógico alto, de modo contrario que en el Z80, por lo que en el circuito de RESET del microprocesador, solo se invierte utilizando una compuerta.

RD	WR	A0	A1	Registro
1	0	0	0	Escribir dato a pto. A
0	1	0	0	Leer dato del pto. A
1	0	0	1	Escribir dato a pto. B
0	1	0	1	Leer dato del pto. B
1	0	1	0	Escribir dato a pto. C
0	1	1	0	Leer dato del pto. C
1	0	1	1	Escribir palabra de control
0	1	1	1	No usado

Tabla 5.1

Las líneas D0-D7 se conectan directamente al Bus de Datos para que por ellas fluya la información entre el Z80 y los

dispositivos externos. La tabla 5.1 muestra la forma en que los registros pueden ser accedidos.

El modo de funcionamiento de los registros 0-2 (A, B, C) es definido por la palabra de control puesta en el bus de datos. El significado de cada uno de los bits de la palabra de control es:

- D0 Puerto C (Bits bajos) 1=entrada, 0=salida
- D1 Puerto B 1=entrada, 0=salida
- D2 Define el Modo de operación del Pto.B; 0=Modo 0, 1=Modo 1
- D3 Puerto C (Bits altos) 1=entrada, 0=salida
- D4 Puerto A 1=entrada, 0=salida
- D5, D6 Definen el Modo de operación del Pto. A ; 00=Modo0, 01=Modo1, 1X=Modo2
- D7 Define el dato como palabra de control; 1=activo

El 8255 tiene tres modos de operación, los cuales son seleccionados por medio de la programación:

- Modo 0 Entrada/Salida Básico
- Modo 1 Entrada/Salida Disparado
- Modo 2 Bus Bidireccional

Cuando este dispositivo es reseteado, automáticamente sus tres puertos son puestos como entradas en Modo 0 y permanecerán así, hasta que se les programe, por medio del software, en otro modo de operación.

El puerto A, junto con los 4 bits más significativos del puerto C forman el llamado grupo A, mientras que el puerto B y los cuatro bits menos significativos del puerto C forman el grupo B. Los grupos pueden ser programados para trabajar en diferentes modos a la vez, por ejemplo el grupo A en modo 0 y el grupo B en Modo 1.

En Modo 1 el 8255 proporciona un medio para la

entrada/salida de datos hacia o desde un puerto específico, en conjunción con señales de disparo o protocolo. En este modo los puertos A y B usan las líneas del puerto C para generar o aceptar esas señales de protocolo, tal como se aprecia en la figura 5.4. En este modo de operación los cuatro Bits más significativos del Puerto C, son usados como líneas de protocolo para el Puerto A y los cuatro Bits menos significativos como líneas de protocolo para el Puerto B.

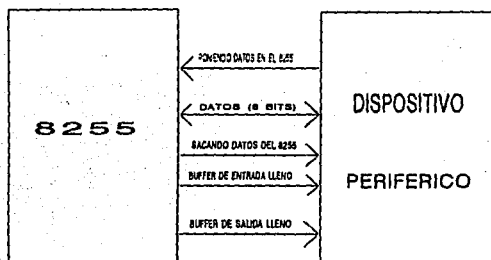


Figura 5.4 8255 En Modo 1

La idea principal de este modo de funcionamiento, es que al realizarse la transferencia de datos, el dispositivo periférico le informe al 8255 que le esta enviando un dato ó que esta listo para recibir uno. Las líneas de protocolo se usan para proporcionar esa información. Esta tecnica es muy común en aplicaciones donde el dispositivo externo es más lento que el microprocesador.

En Modo 2 el el 8255 se configura para comunicarse con algun dispositivo periférico, por medio de un Bus bidireccional de 8 bits para que ambos transmitan ó reciban datos. Las señales de protocolo se utilizan para mantener en forma apropiada el

flujo de datos. En este modo de operación, el Puerto A se utiliza como un Puerto Bidireccional, es decir, las 8 líneas del Puerto A pueden transmitir ó recibir datos desde un dispositivo periférico. Las señales de protocolo son implementadas en el Puerto C.

Por último esta el Modo 0, por medio del cual, se tienen operaciones de entrada/salida de datos en su modo más simple, para cada uno de los tres puertos. No se necesitan señales de protocolo, pues el dato es simplemente escrito en un Puerto específico ó leído de él.

5.3 PROGRAMACION DEL PUERTO

Aquí se utiliza al 8255 en Modo 0, y es programado por medio del software. Para esta aplicación, la palabra de control con la que se programa queda como sigue.

```
LD A,83H ;Cargar Acumulador con PALABRA DE CONTROL  
OUT (03),A;Sacar PALABRA DE CONTROL por puerto de control
```

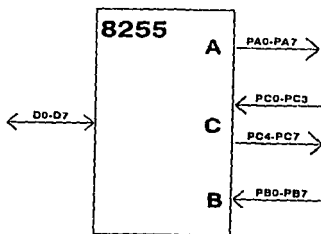


Figura 5.5 Programación del 8255

Con esto se le indica al 8255 que funcione en Modo 0, con el puerto A como salida, el puerto B como entrada, el bloque bajo

del puerto C como entrada y el bloque alto comosaida; el puerto A es el 00, el puerto B 01, el puerto C 02 y el puerto de control 03, tal como se muestra en la figura 5.5.

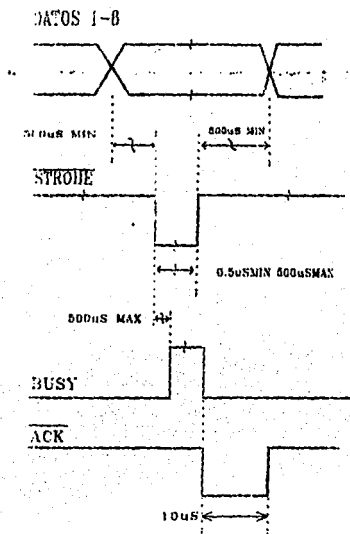


DIAGRAMA DE TIEMPOS DEL PROTOCOLO DE IMPRESION PARALELO (CENTRONICS)

Figura 5.6 Protocolo de Impresión

Por el puerto A se envían los datos de 8 bits en código ASCII al puerto paralelo de la impresora a través de unos buffers (74LS244); estos datos son la tabla de resultados y mensajes entre otros. El comando para enviar esos datos es :

LD A,DATO ;Cargar Acumulador con DATO
OUT (00),A ;Sacar DATO por puerto A

Por el puerto B se introducen al Bus los datos enviados ó proporcionados por el convertidor analógico-digital; es decir, el nivel de la señal de entrada ya digitalizado, se incorpora al bus de datos a través del puerto B para que pueda ser procesado ó almacenado. El comando para leer del convertidor sería :

IN A,(01) ;Leer dato del puerto B

Las señales de control del sistema fluyen a través del puerto C que, como se mencionó antes, tiene programados sus 4 bits menos significativos como entradas y sus 4 bits más significativos como salidas.

5.4 LA INTERFACE DE IMPRESION

La impresión de los resultados obtenidos por el INTEGRADOR, se llevará a cabo en una impresora con entrada paralelo tipo Centronics, esto quiere decir que, el INTEGRADOR contará con una interface, en la cual se podrá conectar cualquier impresora con entrada de datos en paralelo, para las cuales existe ya un protocolo de impresión, tal como se puede apreciar en la figura 5.6.

Las señales de control entre el sistema y la impresora son tres: ACK, BUSY y STROBE. La primera es enviada por la impresora para indicarle al sistema que ha tomado el dato que le envió a través del puerto A, esta señal se utiliza como reloj para disparar un flip-flop tipo D, el cual es alimentado con la señal llamada BUSY; con esta señal la impresora le indica al sistema que esta ocupada y que éste debe esperar a que ella pueda tomar el dato. Como se puede apreciar en la figura 5.7, las señales BUSY y ACK sirven para controlar el flujo de información del sistema hacia la impresora, es decir, para que el sistema envíe

un dato a la impresora, la señal BUSY debe estar en un nivel lógico alto, en el caso contrario no lo hará. La señal BUSY es alimentada por el puerto Pc0 y para sensar su estado se realiza la siguiente rutina:

```

IN A,(02) ;Leer Puerto C
BIT 0,A ;Probar Bit 0 Del Acumulador
  
```

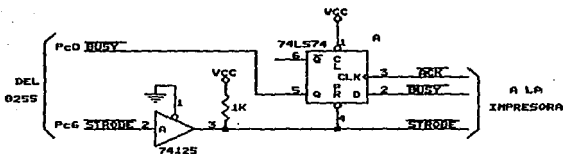


Figura 5.7 Señales De Control IMPRESORA-SISTEMA

La otra señal de control entre el sistema y la impresora, llamada STROBE es normalmente alta y cuando se convierte en un pulso bajo de por lo menos 10 μ S de duración, el sistema le indica a la impresora que le esta enviando un dato, el cual es leído durante el nivel bajo de la señal. El STROBE sale por el puerto Pc6 y, tambien sirve para iniciar el ciclo de envio de un dato activando el pin de Preset del flip-flop anteriormente mencionado, con lo cual la salida Q tomará un nivel alto. El STROBE es generado por programa.

5.5 COMUNICACION CON EL USUARIO

La comunicación entre el usuario y el sistema se dá a través de Pc1, Pc2, Pc4 y Pc5, tal como se observa en la figura 5.8. Por el puerto Pc1 entra una señal que le indica al sistema que debe empezar o finalizar la adquisición de datos, esto se logra mediante un switch tipo push button normalmente abierto, el cual

al ser oprimido por el usuario, conecta el Pci directamente a tierra, indicando que debe empezar o finalizar la adquisición de datos; la señal a través de Pci se debe mantener normalmente alta por medio de una resistencia de pull-up conectada a Vcc.

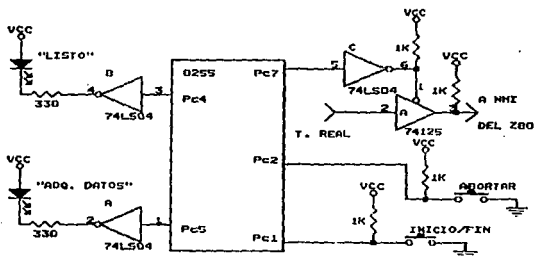


Figura 5.8 Comunicación SISTEMA/USUARIO

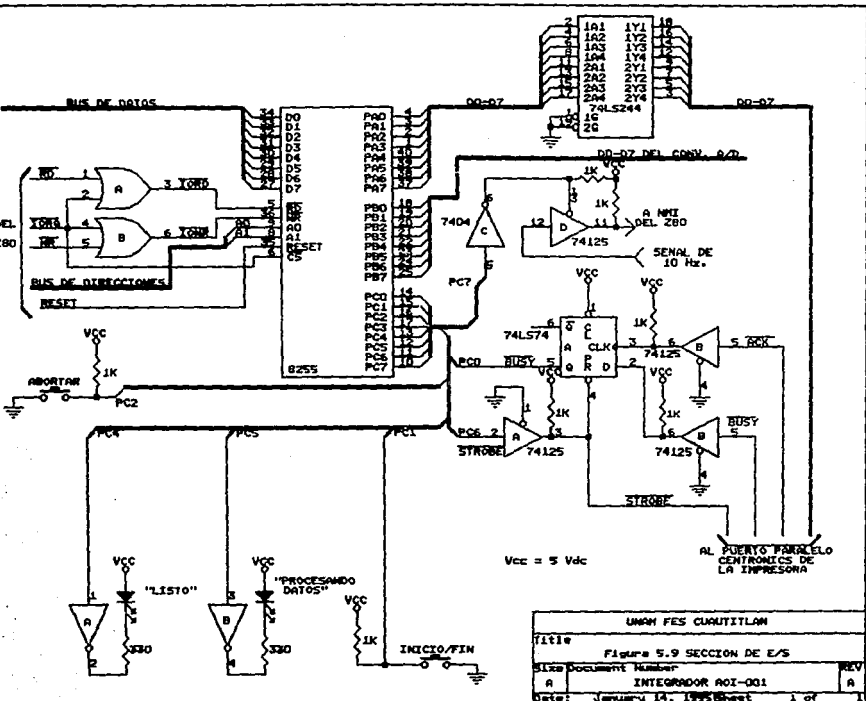
Por el puerto Pc2 el sistema sensa cuando el usuario quiere abortar una corrida; se tiene conectado un switch en forma análoga al anterior, y cuando este switch es oprimido, el sistema lo interpreta como que el usuario quiere abortar una corrida; de esta forma el INTEGRADOR dejará de adquirir datos y no realizará ningún cálculo, solo regresará al inicio del programa para reinicializar todo y esperar la indicación de empezar una nueva corrida.

Por los puertos Pc4 y Pc5 salen dos señales para encender dos Leds, los cuales sirven para indicarle al usuario que el sistema esta "LISTO" para comenzar la adquisición de datos (Pc4) ó que esta "ADQUIRIENDO DATOS" (Pc5).

Por último, por Pc7 sale la señal que activa a un buffer de tercer estado (74125), para que permita el paso de los pulsos de

reloj que contarán el tiempo real por medio de la señal NMI del microprocesador durante el tiempo que dure la corrida.

La configuración completa de toda la sección de este capítulo se muestra en el diagrama de la figura 5.9 .



UNAH FES CUAUTITLAN		
Title: Figura 5.9 SECCION DE E/S		
Schematic Document Number:		
A	INTEGRADOR AOI-Q01	REV
Date: January 14, 1995	Sheet	1 of 1

L A P R O G R A M A C I O N

6.1 INTRODUCCION

Una parte importante del integrador, si no es que la más, son los programas que el microprocesador debe ejecutar; estos programas servirán para llevar a cabo, desde el muestreo de la señal ya digitalizada, hasta la realización de los cálculos necesarios para la integración de los picos y la impresión de los resultados o mensajes.

En la figura 6.1 se muestra el diagrama de flujo general, para la integración de picos cromatográficos. En ella se puede apreciar que el paso inicial es básico, pues el microprocesador y sus puertos son programados para que el sistema pueda iniciar su función como integrador; después mandará a imprimir el mensaje "LISTO", para indicarle al usuario que puede iniciar la corrida, por lo que a partir de aquí el microprocesador solo esperará la indicación del usuario para iniciar la adquisición de los datos.

En cuanto se le indica al sistema que inicie la adquisición, este sensorá la señal ya digitalizada y preguntará periódicamente si termina la adquisición o aborta la corrida, a la vez que paralelamente imprimirá el cromatograma respectivo. Mientras no se le indique lo contrario, el integrador seguirá adquiriendo datos cada que detecte un pico, pero en cuanto el usuario active la tecla de FIN ó la tecla de ABORTAR, el integrador finalizará la adquisición. Si se le pide abortar la corrida, no realizará ningún cálculo y regresará a reinicializar el sistema. Si por el contrario, solo se le indica que termine la corrida, el siguiente paso será la realización de los diferentes cálculos para integrar el área de cada pico y a partir de aquí obtener la tabla de resultados, la cual se imprimirá en cuanto los cálculos sean terminados. Después de imprimir los resultados

de la corrida el programa regresará al punto de inicio para volver a hacer la misma rutina de inicialización y esperar la indicación para iniciar una nueva corrida.

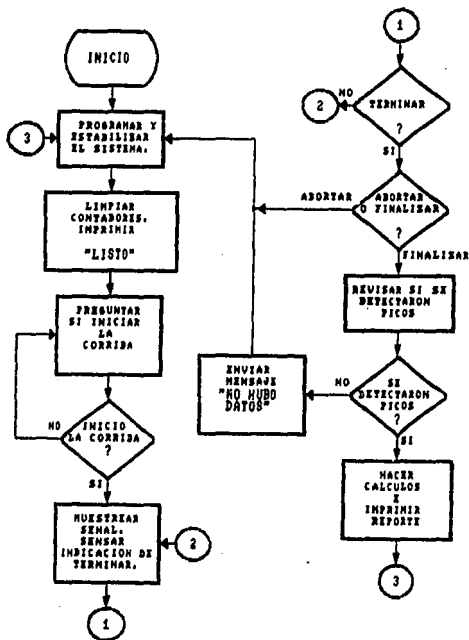


Figura 6.1 Diagrama de flujo general del Integrador

6.2 INICIALIZACION DEL SISTEMA

La rutina de inicialización se llama START, esta ilustrada

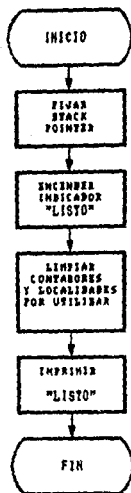


Figura 6.2 Inicialización del sistema

en el diagrama de flujo de la figura 6.2, y sirve para fijar el Stack Pointer a partir de la localidad \$108B, como primera tarea, después de la cual llama a un retardo de 2 mS para esperar a que el sistema se estabilice; en cuanto ese retardo se realiza, se continúa con la programación de los puertos, lo cual en este caso, se realiza a través del puerto de control (03) con la palabra de control 83H, la que programará, como se mencionó en sección de ENTRADA/SALIDA, al puerto A (00) como salida, al puerto B (01) como entrada y al puerto C (02) en su parte baja como entrada y en su parte alta como salida. Una vez programados los puertos, a través del puerto C se enciende el indicador luminoso que indicará que el sistema está listo, al mismo tiempo que se carga el registro índice IX con la localidad donde inicia

la pila de datos (IPDAT) e IY con la localidad donde inicia la pila de Tiempos de Retención, en este caso \$1320 y \$12CD respectivamente, de esta manera cabe la posibilidad de salvar 40 Tiempos de Retención correspondientes a igual número de picos. También los registros alternos D' y BC' son limpiados para que en el primero se lleve a cabo el conteo del número de picos resultantes de la corrida, y en el segundo el conteo de Tiempo Real. Las localidades DATO (\$12C9), DATOANT (\$12GA) y MENOR (\$12CB), también son limpiadas, pues serán utilizadas en la rutina que sirve para imprimir el cromatograma. Por último, el integrador imprimirá el mensaje "LISTO" indicándole al usuario que podrá iniciar la corrida, con solo oprimir la tecla de INICIO/FIN.

6.3 LA ADQUISICION DE DATOS

6.3.1 Inicio y Desarrollo de la Corrida

Después de que el sistema es inicializado, solo esperará a que el usuario le indique el inicio de la corrida al oprimir la tecla INICIO/FIN; esta tecla es monitoreada por medio de la subrutina SWITCH, la cual sensa continuamente el estado lógico del puerto PC1. Mientras la tecla no sea oprimida, el programa permanecerá en un loop esperando a que ésta sea oprimida, para apagar el indicador luminoso de "LISTO" y encender el indicador luminoso "ADQ. DATOS", además de activar el control del contador de Tiempo Real y mantener la señal de STROBE en alto a través de PC5, PC7 y PC6 respectivamente. A partir de aquí, el integrador comenzará a muestrear la señal proveniente del cromatógrafo, cada 100 mSegundos, y por medio de la subrutina INICIO, cuyo diagrama de flujo se muestra en la figura 6.3, se compara el dato capturado actualmente con el dato anterior, de esta forma, si el dato actual es mayor al anterior existe la posibilidad de un pico y, si esto se repite por 5 ocasiones, el integrador lo asume como tal, almacenando sus datos representativos, si no, desechará los datos. Se necesita que sean 5 ocasiones para filtrar los picos que pudieran deberse solo ruido y no a un componente de la muestra a analizar.

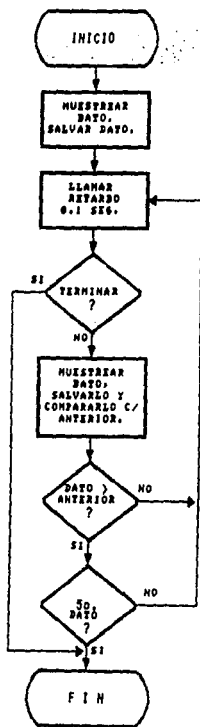


Figura 6.3 Subrutina INICIO

Si el sistema detecta un pico, el programa continuará con la subrutina llamada CIMA, cuyo diagrama de flujo se ilustra en la figura 6.4, la cual es la encargada de seguir almacenando

datos y de compararlos para detectar el punto máximo o cima del pico, momento en le que se llama a una subrutina llamada TIERET,

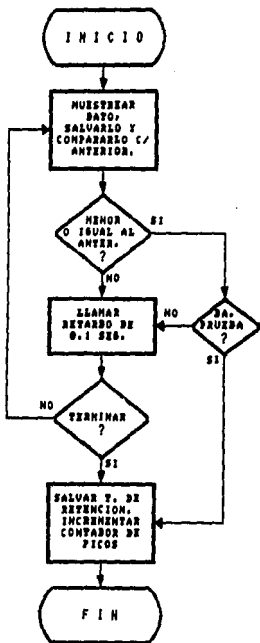


Figura 6.4 Subrutina CIMA

la cual salvará el Tiempo de Retención correspondiente a ese pico; ese Tiempo de Retención será el dato que contenga en ese

momento el registro par BC'; tambien en este momento el registro D', que se utiliza como contador de picos, es incrementado. El punto máximo del pico es detectado cuando en tres ocasiones consecutivas, el dato muestreado es igual o menor que el anterior, de otra forma no se asume como la cima del pico y se continuará muestreando y guardando datos hasta que se cumpla esa condición. La siguiente parte del programa es la subrutina FINPICO, cuyo diagrama de flujo se muestra en la figura 6.5, y la encargada de guardar los datos restantes hasta que el fin de pico es detectado, momento en que la localidad apuntada por IX, es cargada con el dato \$FF para marcar el fin del pico. Se considera que el pico ha finalizado cuando el dato actual es igual ó mayor al dato anterior, mientras tanto, se seguirán salvando los datos del pico.

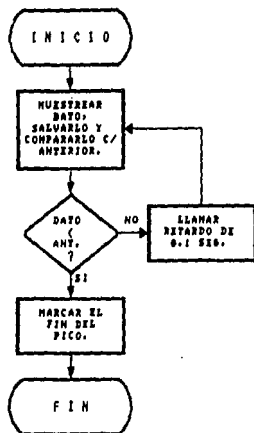


Figura 6.5 Subrutina FINPICO

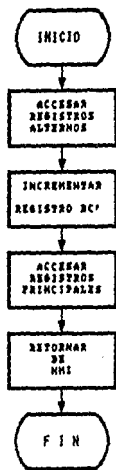


Figura 6.6 Subrutina COTIRE

6.3.2 Conteo de Tiempo Real

El conteo de Tiempo Real es muy importante, debido a que como ya se explicó anteriormente, es una de las informaciones básicas de un análisis cromatográfico. En este caso el conteo de tiempo real está basado en la interrupción no mascarable (NMI); esta interrupción, cuando es solicitada, hace que el microprocesador cargue al contador de programa con \$0066; localidad en la cual se encuentra programado un salto a una subrutina llamada COTIRE, cuyo diagrama de flujo se muestra en la figura 6.6, la cual incrementará el registro par BC', previamente limpiado, cada que una NMI se produzca. En este caso, la señal de reloj tendrá una frecuencia de 0.1 segundos, con lo que los tiempos de retención estarán dados en décimas de segundo.

Como la capacidad del registro par BC' es \$FFFF se tendrá una posibilidad de 65535 décimas de segundo, lo que equivale a un total de 109.225 minutos; por lo tanto las corridas que se podrán realizar con el integrador serán de un máximo de 1.82 horas. La NMI es habilitada a través de PG7 cuando el usuario inicia la corrida y es deshabilitada cuando la termina, para que solo actue durante la adquisición de datos.

6.3.3 El Cromatograma

En cuanto la corrida es iniciada y el sistema comienza a sensar la señal que el cromatografo entrega, tambien entra en funcionamiento la subrutina GRAFICA; esta subrutina se encarga de imprimir el cromatograma de la señal de milivoltaje entregada por el cromatógrafo.

En primer lugar, solo uno de cada diez datos muestreados es impreso, es decir, como el muestreo es cada 100 mS. el cromatograma se compondrá de datos tomados cada 1 S.; esto se hace así por dos causas, primero, si se graficará cada punto que es muestreado el consumo de papel sería excesivo, aun en el espaciamiento entre líneas más pequeño posible; en segundo lugar, con estas condiciones se logra obtener un cromatograma bastante representativo.

La rutina de impresión del cromatograma utiliza una subrutina llamada IMPRI, cuyo digrama de flujo se muestra en la figura 6.7, la cual hace de enlace entre el sistema y la impresora, valiendose de la interface de impresión ya tratada en un capítulo anterior. La subrutina IMPRI envia a la impresora el dato con que esta cargado el acumulador sacándolo por el puerto A; despues genera la señal STROBE y sensa las señales BUSY y ACK provenientes de la impresora; la primera indica si la impresora esta lista o no para recibir un dato, la segunda sirve para que la impresora indique si ha tomado el dato que le es enviado por el sistema. La combinación de ambas señales en la interface de impresión permite saber cuando la impresora ha tomado el dato y

esta lista para recibir otro. La subrutina IMPRI es utilizada en todos los casos en que el sistema imprima algo, ya sea mensajes o resultados.

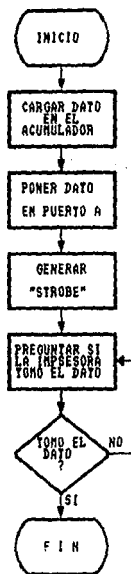


Figura 6.7 Subrutina IMPRI

Regresando a la impresión del cromatograma, la subrutina GRAFICA primero chequea que el punto a graficar corresponda a la décima muestra, si no lo es, regresa al llamado sin imprimir el dato, si por el contrario, se trata de la décima muestra, la subrutina procederá a poner a la impresora, por medio del comando

ESC J 3, con un espaciamento entre líneas de 3/216 ó 3/144 pulgadas segun la impresora de que se trate, por lo que la velocidad de graficación será de 0.83 ó 1.25 pig/min. Despues se continua programando a la impresora con el modo de impresión comprimido, por medio del comando SI, con el cual se podrán imprimir hasta 132 caracteres por línea, de esta forma existe la posibilidad de imprimir todos los puntos que puede resolver el convertidor A/D, ya que el dato antes de imprimirse es dividido entre 2 dando como resultado un dato máximo de 127. Ya dividido entre 2, al datos se le suma 4 como offset, de otra forma la impresión de puntos menores a cuatro no puede llevarse a cabo correctamente. Por último es comparado con el dato anterior, resultando 3 posibilidades, que sea menor, igual o mayor que el dato anterior y dependiendo del resultado, se utilizará una subrutina correspondiente a cada caso para imprimir el cromatograma.

Finalmente con el comando ESC @ la impresora eliminará el espaciamento y el modo comprimido regresando a sus condiciones originales, hasta que esta subrutina sea llamada nuevamente para imprimir un punto del cromatograma.

6.3.4 Fin de la Corrida

Durante el tiempo que dura la corrida el programa pregunta constantemente, por medio de la subrutina FIN, si la finaliza o la aborta. En la figura 6.8 se muestra el diagrama de flujo de esta subrutina. Si el usuario le indica abortar, el programa detendrá el muestreo e irá al principio para inicializar otra vez el sistema y esperar que se le indique que inicia otra corrida y no realizará ningún cálculo. Si por el contrario se le indica que la corrida termina, el programa finalizará el muestreo y el conteo del tiempo real para inmediatamente revisar, por medio de la subrutina REVISION, el contenido del registro D'; si el número de picos es cero se imprimirá el mensaje "NO HUBO DATOS" , en caso de que si se hubiesen detectado picos, el contenido del registro D' se salvará en la localidad NUMPIC(\$12CC) y se

continuará a la parte de cálculos. En caso de no haberse detectado picos el programa regresará a inicializar el sistema y esperará la indicación de iniciar otra corrida.

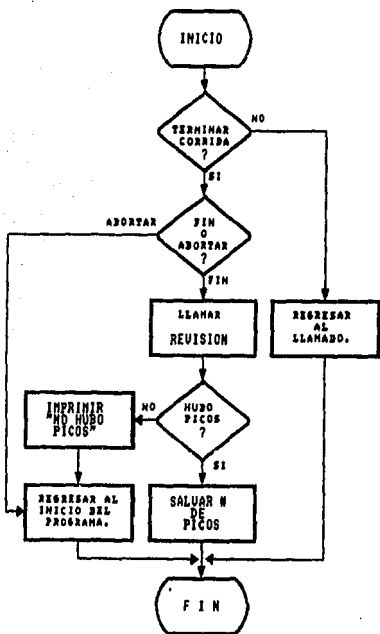


Figura 6.8 Subrutina FIN

6.4 LOS CALCULOS

6.4.1 Introducción.

Esta parte del programa es donde, ya almacenados los datos, se procede a su procesamiento, en el que como primer paso se realiza la integración de los datos de cada pico para calcular su área correspondiente.

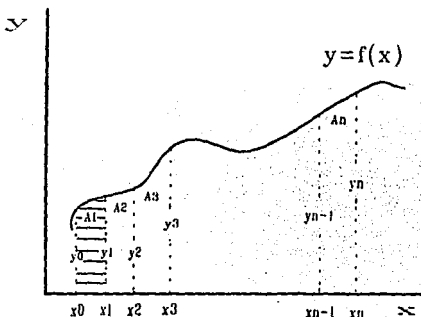


Figura 6.9 Area bajo la curva

Como anteriormente se mencionó, al integrador se le llama así debido a que aplica métodos numéricos para obtener el área bajo la curva de los datos adquiridos durante una corrida cromatográfica. También se mencionó que son dos tipos de información los importantes en la cromatografía: los tiempos de retención de cada pico y sus correspondientes áreas.

En el integrador aquí presentado la señal proveniente del cromatógrafo en forma de milivoltaje, es amplificada e introducida a un convertidor analógico-digital, el cual la digitaliza. La señal ya digitalizada es muestreada periódicamente (cada 100 mS) y cuando un pico es detectado, los

datos muestreados durante la duración del pico son almacenados en la memoria RAM; al término de la corrida estos forman un paquete de datos que representan el perfil de cada pico detectado. A partir de aquí solo restaría hacer la integración de cada uno de ellos.

Existen diversas fórmulas para realizar la integración numérica, unas más exactas que otras pero también con ciertos inconvenientes.

Como se puede ver en la figura 6.9, el área bajo la curva puede ser dividida en áreas pequeñas a cada intervalo de muestreo, realizando la sumatoria se obtendrá el área total.

La fórmula de Simpson 1/3 es :

$$A = h/3 \left[Y_0 + Y_n + 2 \sum_{\text{pares}} \text{ordenadas} + 4 \sum_{\text{impares}} \text{ordenadas} \right]$$

la cual tiene el inconveniente de que se requiere que el número de áreas pequeñas sea par.

También está la fórmula de Simpson 3/8:

$$A = (3/8)h \left[Y_0 + Y_n + 2 \sum_{\text{multiplo de 3}} \text{ordenadas} + 3 \sum_{\text{resto de ordenadas}} \right]$$

la cual es aplicable únicamente si se tiene un número múltiplo de 3 de áreas pequeñas, es decir, n debe ser múltiplo de 3.

Como h es el intervalo de muestreo y no puede ser cambiado durante la corrida, el número de áreas pequeñas en que un pico es dividido es aleatorio, debido a que depende del tiempo en que el pico tarde en salir. Dadas estas circunstancias, se recurre a la fórmula más elemental de integración numérica, la fórmula trapezoidal:

$$A_{1/2} = h/2 \left[Y_0 + Y_n + 2 \sum \text{Resto de ordenadas} \right]$$

en la que, como se puede ver no existe alguna limitante, y es aplicable en todos los casos, por lo que esta fórmula es la que se aplicará para integrar los picos cromatográficos.

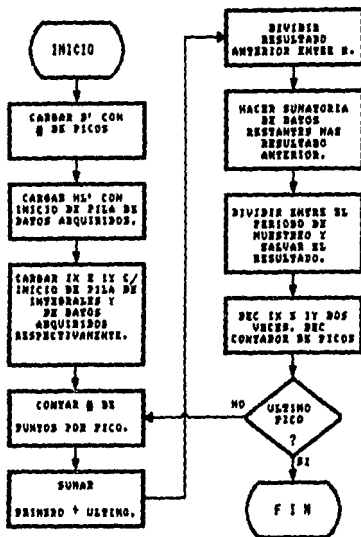


Figura 6.10 Subrutina INTEGRAL

6.4.2 La Integración de los Picos Cromatográficos.

En la figura 6.10 se muestra el diagrama de flujo de la subrutina INTEGRAL, la cual se utiliza para realizar la integración de los picos resultantes de la corrida. En primera instancia, se carga el registro D' con el número de picos salvado en la localidad NUMPIC(\$I2CC) y el registro HL' con la localidad

donde inicia la pila de datos adquiridos IPDAT(\$1320); tambien el registro IY es cargado con la localidad IPDAT y las integrales son salvadas a partir de la localidad IPINT(\$1279) con la cual se carga el registro IX. Despues se procede a utilizar una subrutina llamada CONTADOR, la cual se encarga de contar el número de datos de que se compone cada pico. Para resolver la fórmula, ésta se divide en dos partes, en primera instancia se realiza la suma del primero más el último datos y al resultado se le divide entre dos, el dato obtenido es salvado al intercambiar los registros AF con AF'. Despues se procede a realizar la sumatoria del resto de las ordenadas y el resultado es sumado al se que salvó anteriormente. Por último el total es dividido entre 100, lo cual equivale a multiplicarlo por 0.1 segundos, debido a que este es el período de muestreo, en resumen:

$$\begin{aligned}
 A_{1/2} &= h \left[(Y_0 + Y_n)/2 + \sum \text{Resto de datos} \right] \\
 &= 0.1 \left[(Y_0 + Y_n)/2 + \sum \text{Resto de datos} \right] \\
 &= \frac{(Y_0 + Y_n)/2 + \sum \text{Resto de datos}}{100}
 \end{aligned}$$

Las unidades de área que se obtienen son Volts x Segundo. Cada que se integra un pico el registro D' es decrementado una unidad hasta que todos los picos han sido integrados. El área de cada pico es entonces almacenada una a una a partir de la localidad IPINT(\$1279), y en un máximo de 40 áreas, para ser utilizada en cálculos posteriores.

6.4.3 Area Total y % de Area.

La siguiente subrutina se llama DIVAIAT y sirve para obtener los porcentajes de área, lo cual se logra obteniendo el área total como primer paso para despues dividir el área de cada pico entre ella. El área total se obtiene sumando las áreas de todos los picos, por medio de la subrutina llamada SUMARE. Si el área total fuese mayor a la capacidad del registro HL (65535), el

programa llamará a la subrutina ARTOGRA, la cual imprimirá el mensaje "AREA TOTAL MUY GRANDE" y regresará a reinicializar el sistema, sin seguir los cálculos, para esperar la indicación de iniciar una nueva corrida. Esto es interpretado normalmente como que el volumen de la muestra inyectada debe ser disminuido para obtener una área total más pequeña, pues como se entenderá, el área total es directamente proporcional con el volumen de la muestra que se inyectó.

Si el área total no satura al registro HL, su contenido es salvado en la localidad TOTARE(\$1277), para utilizarla en cálculos posteriores. Después se procede a obtener los porcentajes de área, dividiendo primeramente, el área total entre 10 y salvando el resultado en la localidad TOAREDI(\$1275); después se procede a obtener el porcentaje para cada área bajo el siguiente procedimiento, primero se divide el área correspondiente entre el resultado salvado en TOAREDI, el resultado de la división es multiplicado por 10 y este corresponde a las decenas ó centenas de la parte entera. Enseguida el remanente de la división anterior es multiplicado también por 10 y se divide entre TOAREDI, obteniéndose así las unidades de la parte entera; unidades y decenas son sumadas y salvadas. Con el remanente de la división anterior se sigue el mismo procedimiento para obtener las décimas y centésimas de la parte fraccionaria siendo también salvadas en una localidad contigua a la anterior, es decir:

$$\%Ai = \frac{Ai}{At} \times 100 = \frac{Ai}{At/10} \times 10 = \left[\frac{Ai}{At/10} \right] \times 10$$

Los porcentajes de área de cada pico son salvados en una pila a partir de la localidad IPAIAT(\$1225) apuntada por el registro IX, para después ser convertidos con el fin de imprimir el reporte.

En esta rutina se utilizan las subrutinas MULTIP y DIVISION, las cuales sirven para multiplicar dos datos de 8 bits cada uno en el primer caso, y para dividir dos datos de 16 bits en el segundo caso.

Ya obtenidos y salvados los porcentajes de área individuales, se procede a sumarlos para obtener el total, el cual en teoría debe ser el 100%, pero que en la práctica puede ser unas centésimas mayor o menor, debido al redondeo cuando se calculan los porcentajes de área. Esta suma se realiza por medio de la subrutina SUPOAR y el resultado es salvado en la localidad TOTENT(\$11D0) en su parte entera y en la localidad TOTFRA(\$11D1) en su parte fraccionaria, para su posterior conversión e impresión en el reporte.

6.5 IMPRESION DEL REPORTE

6.5.1 Introducción

Esta parte del integrador es quizá una de las más importantes, debido a que es aquí cuando los resultados de la corrida son puestos a disposición del usuario por medio de una impresora. Anteriormente se mencionó que para la impresión de los resultados, se puede utilizar cualquier impresora, con la única condición de que esta sea con interface paralelo tipo Centronics. Este tipo de interface se caracteriza por tener un método de transmisión de datos en paralelo de 8 bits, entre la impresora y el sistema que le envía datos. La sincronización se hace por medio de la señal STROBE, generada por el sistema, mientras que la respuesta de la impresora es por medio de las señales ACKNLG y BUSY. En el apéndice se muestra el número de pin y su correspondiente señal para este tipo de interface.

Para la impresión de los resultados del integrador, se utilizan solo las señales ACKNLG, BUSY y STROBE, con las cuales es suficiente para establecer la comunicación con la impresora y poder imprimir los resultados. La señal STROBE es normalmente de

un nivel lógico alto y cuando cambia a bajo por un periodo mayor a 10 μ S, la impresora lo interpreta como que se le ha enviado un dato para que lo lea. La impresora leerá el dato durante el intervalo bajo de la señal y, si por alguna causa no lo pudiera leer activa la señal BUSY con un alto para indicar que no puede recibir el dato.

Como se mencionó en un capítulo anterior existe una subrutina llamada IMRPI la cual pone primero el dato en el puerto A, después genera la señal STROBE para poner la salida Q del flip-flop en alto y a la vez indicarle a la impresora que se le ha mandado un dato; como respuesta la impresora genera el BUSY mientras el dato no sea leído completamente; esa señal es utilizada como la entrada D en el flip-flop. En cuanto la impresora ha tomado el dato, la señal BUSY toma un nivel lógico bajo y simultáneamente se genera el pulso bajo del ACK el cual es utilizado como la señal de reloj del flip-flop. El programa sensa después de generar el STROBE, la señal BUSY, la cual será baja en cuanto la impresora genere el ACK. Mientras BUSY sea alto el integrador no enviará otro dato, de esta forma el envío de datos a la impresora es controlado por esas tres señales utilizando solo un flip-flop tipo D para sincronizarlas.

6.5.2 Conversión HEX-DEC-ASCII

Por otro lado, para que la impresora pueda interpretar la información que le envía el integrador, los datos deben ser enviados en código ASCII (American Standard Character for Information Interchange) debido a que prácticamente todas las impresoras existentes en el mercado utilizan este código.

Los caracteres que enviará el integrador son únicamente letras, números y algunos especiales como % y #, los cuales tienen su correspondiente en el código mencionado, como se podrá apreciar en el apéndice.

Para la impresión de los resultados numéricos se tiene que

hacer una conversión Hexadecimal-Decimal-ASCII, es decir, los resultados ya obtenidos y almacenados en la memoria RAM, tienen primero que ser convertidos a código ASCII para poder ser impresos. La conversión no puede realizarse directamente, por lo que tiene que utilizarse una conversión intermedia.

Como primer paso, tiene que realizarse la conversión HEX-DEC, es decir, los datos almacenados en Hexadecimal son convertidos a unidades, decenas, centenas, etc., en Decimal según sea el caso. La conversión HEX-DEC se realiza dividiendo el dato en Hexadecimal entre 10 000, 1 000, 100 y 10 consecutivamente para obtener decenas de millar, millares, centenas, decenas y unidades según la magnitud del dato y salvando cada uno. Como paso siguiente, se realiza la conversión DEC-ASCII, para que los datos puedan ser finalmente interpretados por la impresora. La conversión DEC-ASCII es muy sencilla, y se lleva a cabo solo sumándole un \$30 a cada uno de los datos ya convertidos anteriormente a Decimal, pues no debe olvidarse que el microprocesador solo maneja datos en Hexadecimal aunque estos representen información en Decimal. La conversión HEX-DEC la realizan las subrutinas ONE y TWO, mientras que la conversión DEC-ASCII la realiza la subrutina DEGASC.

6.5.3 Impresión de los Resultados

Antes de realizar la conversión e impresión de los resultados, una subrutina llamada tabla se ocupa de colocar los datos de área, tiempo de retención y porcentaje de área en orden, formando una tabla a partir de las localidades IPDACOAR(\$10DF), IPDAGOTR(\$10E1) e IPDACOPA(\$10E3) respectivamente, esto con el propósito de formar la tabla de resultados para que puedan ser convertidos e impresos.

Los reportes están dispuestos de tal forma que puedan ser impresos en papel tamaño carta y con el pitch de los caracteres

en modo PICA, es decir, 10 caracteres por pulgada, con lo cual el número de caracteres por línea será de 80.

"LISTO"

.....
.....
.....

INTEGRADOR AOI-001

REPORTE DEL ANALISIS

PICO#	AREA	TIEMPO DE RETENCION	%AREA
01	000123	000045	51.25
02	000121	000116	50.41
TOTALES	000244		101.66%

Figura 6.11

En la figura 6.11 se muestra un ejemplo de una corrida y su correspondiente reporte, en ella se puede apreciar que en primera instancia se encuentra el cromatograma donde se pueden observar algunos picos, despues esta escrito un encabezado en el cual se imprimen los datos del integrador, enseguida de algunas lineas en blanco, esta la leyenda que identifica a la impresion como un reporte de la corrida, por último, tambien despues de unos espacios en blanco, se encuentran 4 columnas debajo de las cuales se imprimiran los resultados; esas columnas son PICO#, AREA, TIEMPO DE RETENCION Y %AREA. En la primera se asigna un número consecutivo a cada pico detectado, en la siguiente se imprime el

área correspondiente a ese pico, en la tercera su tiempo de retención y en la última su porcentaje de área respectivo.

El centrado y espaciamientos correspondientes a cada línea se hace "imprimiendo" espacios en blanco en igual número a cada lado de los textos. En la realidad, a cada carácter del código ASCII le corresponde un número en Hexadecimal. Las tablas de los textos, como las de el encabezado y otros, son almacenadas en la memoria ROM y son llamadas cada que son utilizadas.

La impresión del encabezado es relativamente sencilla ya que el texto siempre será el mismo y solo es llamada cada que el integrador imprima un reporte; la impresión de los resultados es un poco más complicada, debido a que sus caracteres representativos (en este caso números) son aleatorios y están fuera del alcance del usuario.

Para la impresión del encabezado se utiliza el programa llamado ENCABEZA, el cual se encarga de imprimir, carácter por carácter, cada una de las líneas de que consta el encabezado.

Después de la impresión del encabezado se utiliza el programa REPORTE para imprimir los resultados numéricos, lo cual será realizado línea por línea, es decir, los datos relacionados con un pico en particular, serán impresos en una sola línea correspondientes a ese pico. Como primer paso se realiza la conversión HEX-DEC-ASCII del número de pico, área, tiempo de retención y porcentaje de área correspondientes al primer pico y los datos ya convertidos son colocados a partir de la localidad DATIMP(\$10CB). Enseguida se llama a la subrutina RESULTAD, la cual imprimirá la línea de datos correspondientes al pico en cuestión abajo de su respectiva columna. Por último, la subrutina REPORTE revisa si hay más datos correspondientes a otros picos, y en caso de haberlos se repite el procedimiento anterior.

En cuanto ya no existen más datos por convertir e imprimir,

el programa continua con una subrutina llamada IMPTOT, la que se encargará de imprimir la línea de los totales, tanto de área como de porcentajes; para esto se ayuda de las subrutinas TOTALES, AREATOT, INTER y PORAR las cuales se encargan de darle forma a esta línea. Antes, la conversión HEX-DEC-ASCII y su ordenamiento son realizados por las subrutinas CONARTOT y CONPORAR.

La última tarea que le queda al programa será reinicializar el sistema, ejecutando para esto un salto a la localidad \$0000 con el fin de reprogramarse y estar listo para iniciar una nueva corrida cuando el usuario se lo indique.

EL PROGRAMA ENSAMBLADO

```

1. 0000: ;PROGRAMA PRINCIPAL DEL INTEGRADOR
2. 0000: LOSTPO: EQU $108B ;STACK POINTER
3. 0000: IPDAT: EQU $1320 ;PILA DE DATOS
4. 0000: IPTRET: EQU $12CD ;PILA DE T. RET
5. 0000: NUMPIC: EQU $12CC ;LOC P/SALVAR #PICOS
6. 0000: IPINT: EQU $1279 ;PILA INTEGRALES
7. 0000: IPAIAT: EQU $1225 ;PILA DE AI/AT
8. 0000: TOTARE: EQU $1277 ;LOC P/SALVAR AT
9. 0000: TOAREDI: EQU $1275 ;LOC P/SALVAR AT/10
10. 0000: PARENT: EQU $11D3 ; " ENTEROS
11. 0000: PARFRA: EQU $11D4 ; " FRACC
12. 0000: SPARENT: EQU $11D2 ; " SUMA ENT
13. 0000: TOTFRA: EQU $11D1 ; " TOTAL FRAC
14. 0000: TOTENT: EQU $11D0 ; " ENT
15. 0000: IPDACO: EQU $10DF ;PILA DATOS A CONV
16. 0000: DATIMP: EQU $10CB ;DATO x IMP
17. 0000: PTOA: EQU $00 ;PUERTO A
18. 0000: PTOB: EQU $01 ;PUERTO B
19. 0000: PTOC: EQU $02 ;PUERTO C
20. 0000: PTOCONT: EQU $03 ;PUERTO CONTROL
21. 0000: DATO: EQU $12C9 ;DATO ACTUAL
22. 0000: DATOANT: EQU $12CA ;DATO ANTERIOR
23. 0000: MENOR: EQU $12C8 ;DATO MENOR
24. 0000: LISTOPI: EQU $0EFC ;MENSAJE "LISTO"
25. 0000: TAB1: EQU $0F00 ;ENCABEZADO
26. 0000: LNHDAT: EQU $0ED7 ;"NO HUBO DATOS"
27. 0000: LATGRA: EQU $0EBE ;"AREA TOTAL MUY GRANDE"
28. 0000: IPDACOPA: EQU $10E3 ;XA P/CONVERTIR
29. 0000: IPDACOTR: EQU $10E1 ;T. RET P/CONVERTIR
30. 0000: IPDACOAR: EQU $10DF ;AREA P/CONVERTIR
31. 0000: ARPOR: EQU $10C4 ;XAREA
32. 0000: ARTOT: EQU $10BE ;AREA TOTAL
33. 0000: TOTALS: EQU $0FB5 ;SUBROUTINA
34. 0000: GRAFICA: EQU $0500 ;
35. 0000: FIN: EQU $057D ;
36. 0000: LISTO: EQU $05D1 ;
37. 0000: SWITCH: EQU $05E5 ;
38. 0000: RETARDO: EQU $05F5 ;
39. 0000: RETISEG: EQU $0800 ;
40. 0000: EN CAB: EQU $0853 ;
41. 0000: CONTADOR: EQU $087C ;
42. 0000: MULT10: EQU $068A ;
43. 0000: DIVID: EQU $0893 ;
44. 0000: SUMARE: EQU $08A0 ;
45. 0000: DIVISION: EQU $0700 ;
46. 0000: ALINLI: EQU $0721 ;
47. 0000: IMPRI: EQU $072C ;

```

```

48. 0000:      DALE:      EQU $073F      ;      "
49. 0000:      ONE:      EQU $0783      ;      "
50. 0000:      TWO:      EQU $0770      ;      "
51. 0000:      THREE:     EQU $07A4      ;      "
52. 0000:      RESULTAD:  EQU $07D1      ;      "
53. 0000:      TOTALES:   EQU $0858      ;      "
54. 0000:      AREATOT:   EQU $086A      ;      "
55. 0000:      INTER:     EQU $087B      ;      "
56. 0000:      FORAR:     EQU $0881      ;      "
57. 0000:      CONARTOT:  EQU $081A      ;      "
58. 0000:      CONFORAR:  EQU $082B      ;      "
59. 0000:
60. 0000:      31 8B 10      COMIENZO LD SP,LOSTPO      ;PROG STACK POINTER
61. 0003:      CD 3A 00      CALL ESPERA      ;ESTABILIZACION
62. 0008:      3E 83      PROGPTO LD A,$83      ;PROG PUERTOS
63. 000B:      D3 03      OUT (PTOCONT),A
64. 000A:      18 3D      JR CORRIDA
111. 003A:      21 FF 02      ESPERA LD HL,$02FF
112. 003D:      2D      ESPERA1 DEC L
113. 003E:      20 FD      JR NZ,ESPERA1
114. 0040:      25      DEC H
115. 0041:      20 FA      JR NZ,ESPERA1
116. 0043:      C8      RET
117. 0044:      D9      COTIRE EXX
118. 0045:      03      INC BC
119. 0046:      D9      EXX
120. 0047:      ED 45      RETN
121. 0049:      ;PROGRAMA PARA DETECTAR LOS PICOS,
122. 0049:      CD 81 00      CORRIDA CALL START      ;RUTINA DE ADQUISICION
123. 004C:      CD 88 00      MORE CALL INICIO      ;DE DATOS Y GRAFICACION.
124. 004F:      CD 30 01      CALL CINA
125. 0052:      CD 9E 01      CALL FINFICO
126. 0055:      18 F5      JR MORE
142. 0086:      C3 44 00      NMI JP COTIRE      ;CONTADOR DE TIEMPO
143. 0089:      ;PROGRAMA QUE REALIZA LOS CALCULOS DESPUES
144. 0089:      ;DE TERMINAR UNA CORRIDA.
145. 0089:      CD C2 01      CALCULOS CALL INTEGRAL      ;OBTENER ARRA
146. 008C:      CD 13 02      CALL DIVAIAT      ;      " XAREA
147. 008F:      CD 7A 02      CALL SUPOAR
148. 0072:      CD C6 02      CALL TABLA      ;ACOHODAR RESULTADOS
149. 0075:      CD E8 02      CALL REPORTE      ;IMPRIMIR EL
150. 0078:      CD 25 03      CALL CONTOT      ;REPORTE DE LA
151. 007B:      CD 2C 03      CALL IMPTOT      ;CORRIDA
152. 007E:      C3 00 00      JP COMIENZO      ;INICIALIZAR
153. 0081:      ;PROGRAMA PARA LIMPIAR CONTADORES
154. 0081:      ;Y ESPERAR LA INDICACION DE INICIAR.
155. 0081:      3E 50      START LD A,$50      ;PROGRAMAR INDICADORES
156. 0083:      D3 02      OUT (PTOC),A      ;Y HABILITADORES.
157. 0085:      DD 21 20 13      LD IX,IPDAT
158. 0089:      FD 21 CD 12      LD IY,IPTRT
159. 008D:      21 C9 12      LD HL,DATO
160. 0090:      38 00      LD (HL),$00
161. 0092:      21 CA 12      LD HL,DATOANT

```


182.	0085:	38 04		LD (HL),#04	:OFFSET
183.	0087:	21 C8 12		LD HL,MENOR	
184.	009A:	38 00		LD (HL),#00	
185.	009C:	0E 0A		LD C,#140A	
186.	009E:	D8		EXX	
187.	009F:	01 00 00		LD BC,\$0000	:LIMPIAR C. TIEMPO
188.	00A2:	18 00		LD D,\$00	: " C. PICOS
189.	00A4:	D8		EXX	
170.	00A5:	DD E5		PUSH IX	
171.	00A7:	CD D1 05		CALL LISTO	:IMPRIMIR "LISTO"
172.	00AA:	DD E1		POP IX	
173.	00AC:	CD E5 05	STARTER	CALL SWITCH	:INICIO CORRIDA?
174.	00AF:	C5		PUSH BC	
175.	00B0:	CD 00 08		CALL RETISEG	:ESPERAR 1SEG
178.	00B3:	C1		POP BC	
177.	00B4:	3E 0E		LD A,\$0E	
178.	00B8:	D3 02		OUT (PTOC),A	
179.	00B8:	C9		RET	
180.	00B9:	DB 01	INICIO	IN A,(PTOB)	:TOMAR 1er DATO
181.	00BB:	DD 77 00	CARGA	LD (IX+00),A	
182.	00BE:	CD 00 05		CALL GRAFICA	:GRAFICAR C/20 PUNTOS
183.	00C1:	CD F5 05		CALL RETARDO	:ESPERAR 100ms
184.	00C4:	CD 7D 05		CALL FIN	:FIN O ABORTAR?
185.	00C7:	DB 01		IN A,(PTOB)	:TOMAR 2o DATO
188.	00C9:	DD 77 01		LD (IX+01),A	
187.	00CC:	CD 00 05		CALL GRAFICA	
188.	00CF:	DD 7E 01		LD A,(IX+01)	
189.	00D2:	DD BE 00		CP (IX+00)	:2o-1ro
190.	00D5:	28 E4		JR Z,CARGA	: = REGRESA
191.	00D7:	38 E2		JR C,CARGA	: < REGRESA
192.	00D9:	CD F5 05		CALL RETARDO	
193.	00DC:	CD 7D 05		CALL FIN	
194.	00DF:	DB 01		IN A,(PTOB)	:TOMAR 3er DATO
195.	00E1:	DD 77 02		LD (IX+02),A	
196.	00E4:	CD 00 05		CALL GRAFICA	
197.	00E7:	DD 7E 02		LD A,(IX+02)	
198.	00EA:	DD BE 01		CP (IX+01)	:3o-2o
199.	00ED:	28 CC		JR Z,CARGA	: = REGRESA
200.	00EF:	38 CA		JR C,CARGA	: < REGRESA
201.	00F1:	CD F5 05		CALL RETARDO	
202.	00F4:	CD 7D 05		CALL FIN	
203.	00F7:	DB 01		IN A,(PTOB)	:TOMAR 4o DATO
204.	00F9:	DD 77 03		LD (IX+03),A	
205.	00FC:	CD 00 05		CALL GRAFICA	
206.	00FF:	DD 7E 03		LD A,(IX+03)	
207.	0102:	DD BE 02		CP (IX+02)	:4o-3o
208.	0105:	28 B4		JR Z,CARGA	: = REGRESAR
209.	0107:	38 B2		JR C,CARGA	: < REGRESAR
210.	0108:	CD F5 05		CALL RETARDO	
211.	010C:	CD 7D 05		CALL FIN	
212.	010F:	DB 01		IN A,(PTOB)	:TOMAR 5o DATO
213.	0111:	DD 77 04		LD (IX+04),A	
214.	0114:	CD 00 05		CALL GRAFICA	

215.	0117:	DD	7E	04	LD A,(IX+04)	
216.	011A:	DD	BE	03	CP (IX+03)	;50-40
217.	011D:	28	9C		JR Z,CARGA	;= REGRESAR
218.	011F:	38	9A		JR C,CARGA	
219.	0121:	CD	F5	05	CALL RETARDO	
220.	0124:	CD	7D	05	CALL FIN	
221.	0127:	DD	23		INC IX	
222.	0129:	DD	23		INC IX	
223.	012B:	DD	23		INC IX	
224.	012D:	DD	23		INC IX	
225.	012F:	C8			RET	
226.	0130:				;PROGRAMA PARA DETECTAR LA CIMA	
227.	0130:				;DE UN PICO.	
228.	0130:	DB	01		CIMA IN A,(PTOB)	;TOMAR DATO
229.	0132:	DD	77	01	LD (IX+01),A	
230.	0135:	CD	00	05	CALL GRAFICA	
231.	0138:	DD	7E	01	LD A,(IX+01)	
232.	013B:	DD	BE	00	CP (IX+00)	;DATO-DATOANT
233.	013E:	DD	23		INC IX	
234.	0140:	28	0A		JR Z,UNO	;= POSIBLE CIMAPICO
235.	0142:	38	08		JR C,UNO	
236.	0144:	CD	F5	05	CALL RETARDO	
237.	0147:	CD	7D	05	CALL FIN	
238.	014A:	18	E4		JR CIMA	;REGRESA
239.	014C:	CD	F5	05	CALL RETARDO	
240.	014F:	CD	7D	05	CALL FIN	
241.	0152:	DB	01		IN A,(PTOB)	;TOMA DATO
242.	0154:	DD	77	01	LD (IX+01),A	
243.	0157:	CD	00	05	CALL GRAFICA	
244.	015A:	DD	7E	01	LD A,(IX+01)	
245.	015D:	DD	BE	00	CP (IX+00)	;DATO-DATOANT
246.	0160:	DD	23		INC IX	
247.	0162:	28	0A		JR Z,DOS	;=POSIBLE CIMAPICO
248.	0164:	38	08		JR C,DOS	
249.	0166:	CD	F5	05	CALL RETARDO	
250.	0169:	CD	7D	05	CALL FIN	
251.	016C:	18	C2		JR CIMA	;REGRESAR
252.	016E:	CD	F5	05	CALL RETARDO	
253.	0171:	CD	7D	05	CALL FIN	
254.	0174:	DB	01		IN A,(PTOB)	;TOMAR DATO
255.	0176:	DD	77	01	LD (IX+01),A	
256.	0178:	CD	00	05	CALL GRAFICA	
257.	017C:	DD	7E	01	LD A,(IX+01)	
258.	017F:	DD	BE	00	CP (IX+00)	;DATO-DATOANT
259.	0182:	DD	23		INC IX	
260.	0184:	28	0A		JR Z,TIERET	;CIMA DE PICO
261.	0186:	38	08		JR C,TIERET	; " "
262.	0188:	CD	F5	05	CALL RETARDO	
263.	018B:	CD	7D	05	CALL FIN	
264.	018E:	18	A0		JR CIMA	;REGRESAR
265.	0190:	D8			EXX	
266.	0191:	FD	70	00	LD (IY+00),B	;SALVAR T. DE
267.	0194:	FD	71	01	LD (IY+01),C	;RETENCION E

```

268. 0187: FD 23          INC IY          ;INCREMENTAR CONTADOR
268. 0189: FD 23          INC IY          ;DE PICOS.
270. 018B: 14            INC D
271. 018C: D9            EXX
272. 018D: C8            RET
273. 018E:                ;PROGRAMA PARA FINALIZAR LA
274. 018E:                ;DETECCION DE UN PICO.
275. 018E: CD F5 05      FINPICO        CALL RETARDO
276. 01A1: CD 7D 05      CALL FIN
277. 01A4: D8 01        IN A,(PTOB)    ;TOMAR DATO
278. 01A8: DD 77 01      LD (IX+01),A
279. 01A9: CD 00 05      CALL GRAFICA
280. 01AC: DD 7E 01      LD A,(IX+01)
281. 01AF: DD BE 00      CP (IX+00)     ;DATO-DATOANT
282. 01B2: DD 23          INC IX
283. 01B4: 38 E8        JR C,FINPICO   ;< SEGUIR MUESTREANDO
284. 01B6: DD 23          INC IX
285. 01B8: DD 36 00 FF   LD (IX+00),$FF ;MARCAR EL FIN
286. 01BC: DD 23          INC IX          ;DEL PICO
287. 01BE: CD 7D 05      CALL FIN
288. 01C1: C9            RET
289. 01C2:                ;INTEGRACION DE PICOS
290. 01C2: D9            INTEGRAL       EXX
291. 01C3: 21 CC 12      LD HL,NUMPIC   ;RECUPERAR # DE PICOS
292. 01C8: 58            LD D,(HL)
293. 01C7: 21 20 13      LD HL,IPDAT
294. 01CA: D9            EXX
295. 01CB: DD 21 79 12   LD IX,IPINT    ;PILA DE INTEGRALES
296. 01CF: FD 21 20 13   LD IY,IPDAT    ;PILA DE DATOS
297. 01D3: CD 7C 06      PRINCIPPIO    CALL CONTADOR  ;DATOS x PICO
298. 01D8: FD 7E 00      LD A,(IY+00)   ;CARGAR 1er DATO
299. 01D9: 0B            DEC BC
300. 01DA: FD E5          PUSH IY
301. 01DC: E1            POP HL
302. 01DD: 09            ADD HL,BC
303. 01DE: 88            ADD A,(HL)     ;1ro + ULTIMO
304. 01DF: CB 3F          SRL A          ;DIVIDIR/2
305. 01E1: 08            EX AF,AF       ;SALVAR
306. 01E2: FD 23          INC IY
307. 01E4: 0B            DEC BC
308. 01E5: 21 00 00      LD HL,$0000
309. 01E8: 18 00          LD D,$00
310. 01EA: FD 5E 00      OTRO2          LD E,(IY+00)   ;
311. 01ED: 19            ADD HL,DE      ;SUMAR RESTO DE DATOS
312. 01EE: FD 23          INC IY
313. 01F0: 0B            DEC BC
314. 01F1: 78            LD A,B
315. 01F2: B1            OR C
316. 01F3: 20 F5          JR NZ,OTRO2
317. 01F5: 08            EX AF,AF       ;RECUPERAR
318. 01F6: 5F            LD R,A
319. 01F7: 19            ADD HL,DE      ;RESULTADO FINAL
320. 01F8: 44            LD B,H

```

```

321. 01F9: 4D          LD C,L
322. 01FA: 1E 0A      LD E,#0A
323. 01FC: CD 00 07   CALL DIVISION ;DIVIDIR/10
324. 01FF: DD 70 00   LD (IX+00),B ;SALVAR LA
325. 0202: DD 71 01   LD (IX+01),C ;INTEGRAL.
326. 0205: FD 23      INC IY
327. 0207: FD 23      INC IY
328. 0208: DD 23      INC IX
329. 020B: DD 23      INC IX
330. 020D: D9         RXX
331. 020E: 15         DEC D ;DEC #PICOS
332. 020F: D9         RXX
333. 0210: 20 C1      JR NZ,PRINCIPIO;INTEGRAR MAS PICOS
334. 0212: C9         RET
335. 0213:
336. 0213: ;PROG. PARA DIVIDIR A1 ENTRE EL AT Y
;OBTENER LOS PORCENTAJES.
337. 0213: CD A0 06   DIVAIAT CALL SUMARE ;OBTENER A. TOTAL
338. 0218: FD 21 79 12 LD IY,IPINT
339. 021A: DD 21 25 12 LD IX,IPAIAT ;PILA A1/AT
340. 021E: D9         RXX
341. 021F: 21 CC 12   LD HL,NUMPIC
342. 0222: 46         LD B,(HL)
343. 0223: D9         RXX
344. 0224: ED 4B 77 12 LD BC,(TOTARE) ;RESCATAR AT
345. 0228: 11 0A 00   LD DE,#000A
346. 022B: CD 00 07   CALL DIVISION ;DIVIDIR AT/10
347. 022E: ED 43 75 12 LD (TOAREDI),BC;SALVAR AT/10
348. 0232: ED 5B 75 12 UNHAS LD DE,(TOAREDI)
349. 0236: FD 48 00   LD B,(IY+00)
350. 0239: FD 4E 01   LD C,(IY+01)
351. 023C: CD 00 07   CALL DIVISION ;A1/(AT/10)
352. 023F: E5         PUSH HL ;SALVAR REMANENTE
353. 0240: 50         LD D,B
354. 0241: 58         LD E,C
355. 0242: CD 8A 08   CALL MULT10 ;RESULTADO x 10
356. 0245: DD 75 00   LD (IX+00),L ;SALVAR (CENT Y DEC)
357. 0248: D1         POP DE ;RECUPERAR REM
358. 0249: CD 93 06   CALL DIVID ;OBTENER UNIDADES
359. 024C: 78         LD A,C
360. 024D: DD 88 00   ADD A,(IX+00) ;OBTENER P. ENTERA
361. 0250: DD 77 00   LD (IX+00),A ;Y SALVARLA.
362. 0253: 54         LD D,H
363. 0254: 5D         LD E,L
364. 0255: CD 93 06   CALL DIVID
365. 0258: E5         PUSH HL
366. 0259: 50         LD D,B
367. 025A: 58         LD E,C
368. 025B: CD 8A 08   CALL MULT10 ;OBTENER (Dec Y Cent)
369. 025E: DD 75 01   LD (IX+01),L
370. 0261: D1         POP DE
371. 0262: CD 93 08   CALL DIVID ;OBTENER Miliesimas
372. 0265: 78         LD A,C
373. 0268: DD 88 01   ADD A,(IX+01) ;OBTENER P. FRACC
374. 0268: DD 77 01   LD (IX+01),A ;Y SALVARLA

```

375.	026C:	DD	23			INC IX
378.	026E:	DD	23			INC IX
377.	0270:	FD	23			INC IY
378.	0272:	FD	23			INC IY
379.	0274:	D8				EXX
380.	0275:	05				DEC B
381.	0276:	D8				EXX
382.	0277:	20	B8			JR NZ,UNHAS
383.	0278:	C9				RET
384.	027A:					;PROG. PARA SUHAR LOS XAREA Y
385.	027A:					;OBTENER SU TOTAL.
386.	027A:	DD	21	D3	11	SUPOAR LD IX,PARENT ;PILA DE P. ENT
387.	027E:	FD	21	D4	11	LD IY,PARFRA ;PILA DE P. FRACC
388.	0282:	D8				EXX
389.	0283:	21	CC	12		LD HL,NUMPIC
390.	0288:	46				LD B,(HL)
391.	0287:	D8				EXX
392.	0288:	3E	00			LD A,\$00 ;LIMPIAR
393.	028A:	DD	4E	00		HASE LD C,(IX+00)
394.	028D:	81				ADD A,C ;SUMATORIA ENTEROS
395.	028E:	DD	23			INC IX
396.	0290:	DD	23			INC IX
397.	0292:	D8				EXX
398.	0293:	05				DEC B
399.	0294:	D8				EXX
400.	0295:	20	F8			JR NZ,HASE ;SUHAR HAS ENT
401.	0297:	32	D2	11		LD (SPARENT),A ;SALVAR ENTEROS
402.	029A:	D8				EXX
403.	029B:	21	CC	12		LD HL,NUMPIC
404.	029E:	46				LD B,(HL)
405.	029F:	D8				EXX
408.	02A0:	21	00	00		LD HL,\$0000 ;LIMPIAR
407.	02A3:	06	00			LD B,\$00
408.	02A5:	FD	4E	00		MASF LD C,(IY+00)
409.	02A8:	08				ADD HL,BC ;SUHAR FRACCIONES
410.	02A9:	FD	23			INC IY
411.	02AB:	FD	23			INC IY
412.	02AD:	D8				EXX
413.	02AE:	05				DEC B
414.	02AF:	D8				EXX
415.	02B0:	20	F3			JR NZ,MASF
418.	02B2:	44				LD B,H ;SUHAR HAS FRACC
417.	02B3:	4D				LD C,L
418.	02B4:	11	84	00		LD DE,\$0084
418.	02B7:	CD	00	07		CALL DIVISION ;FRACC/100
420.	02BA:	7D				LD A,L
421.	02BB:	32	D1	11		LD (TOTFRA),A ;SALVAR TOTAL FRACC
422.	02BE:	3A	D2	11		LD A,(SPARENT) ;OBTENER
423.	02C1:	81				ADD A,C ;TOTAL ENTEROS
424.	02C2:	32	D0	11		LD (TOTENT),A
425.	02C5:	C9				RET

```

426. 02C8: ;PROG. PARA ACOMODAR LOS RESULTADOS
427. 02C8: ;EN UNA TABLA.
428. 02C8: DD 21 79 12 TABLA LD IX,IPINT
429. 02CA: FD 21 DF 10 LD IY,IPDACOAR
430. 02CE: CD 3F 07 CALL DALE
431. 02D1: DD 21 CD 12 LD IX,IPTRRET
432. 02D5: FD 21 E1 10 LD IY,IPDACOTR
433. 02D8: CD 3F 07 CALL DALE
434. 02DC: DD 21 25 12 LD IX,IPAIAT
435. 02E0: FD 21 E3 10 LD IY,IPDACOPA
436. 02E4: CD 3F 07 CALL DALE
437. 02E7: C9 RET
438. 02E8: ;PROG. PARA IMPRIMIR LA TABLA
439. 02E8: ;DE RESULTADOS.
440. 02E8: CD 53 08 REPORTE CALL EN CAB ;IMPRIMIR ENCABEZADO
441. 02EB: FD 21 DF 10 LD IY,IPDACO ;PILA DE DATOS A CONV
442. 02EF: D9 EXX
443. 02F0: 21 CC 12 LD HL,NUMPIC
444. 02F3: 48 LD B,(HL)
445. 02F4: 1E 01 LD E,#01
446. 02F8: 7B LD A,E
447. 02F7: D9 OTRALI EXX
448. 02F8: DD 21 CB 10 LD IX,DATIMP
449. 02FC: 08 00 LD B,#00 ;#PICOS EN BC
450. 02FE: 4F LD C,A
451. 02FF: CD 63 07 CALL ONE ;#PICOS DEC-ASCII
452. 0302: CD 70 07 CALL TWO ;AREA
453. 0305: CD 70 07 CALL TWO ;T.RET
454. 0308: CD A4 07 CALL THREE
455. 030B: CD 83 07 CALL ONE ;XA ENT
456. 030E: 3E 2E LD A,#2E ;PUNTO DECIMAL
457. 0310: DD 77 00 LD (IX+00),A
458. 0313: DD 23 INC IX
459. 0315: CD A4 07 CALL THREE
460. 0318: CD 83 07 CALL ONE ;XA FRACC DEC-ASCII
461. 031B: CD D1 07 CALL RESULTAD ;IMPRIMIR UNA LINEA
462. 031E: D9 EXX
463. 031F: 1C INC E
464. 0320: 7B LD A,E
465. 0321: 10 D4 DJNZ OTRALI
466. 0323: D9 EXX
467. 0324: C9 RET
468. 0325: ;PROG. QUE CONVIERTE Y COLOCA
469. 0325: ;LOS TOTALES PARA IMPRIMIRLOS.
470. 0325: CD 1A 08 CONTOT CALL CONARTOT ;DEC-ASCII AT
471. 0328: CD 2B 08 CALL CONPORAR ;DEC ASCII XA TOTAL
472. 032B: C9 RET
473. 032C: ;PROG P/IMPRIMIR LA LINEA DE TOTALES
474. 032C: ;DEL REPORTE
475. 032C: CD 58 08 IMPTOT CALL TOTALES ;IMP INICIO DE LINEA
476. 032F: CD 8A 08 CALL AREATOT ;IMP AREA TOTAL
477. 0332: CD 7B 08 CALL INTER ;36 ESPACIOS
478. 0335: CD 81 08 CALL PORAR ;IMP XA TOTAL

```

479.	0338:	CD 21 07	CALL ALINLI	;IMPRIMIR TODA LA LINEA
480.	033B:	CD 21 07	CALL ALIHLI	;DEJAR DOS LINEAS
481.	033E:	CD 21 07	CALL ALINLI	;EN BLANCO
482.	0341:	C9	RET	
20.	0000:		:PROGRAMA PARA IMPRIMIR EL	
21.	0000:		:CROMATOGRAMA DURANTE LA CORRIDA.	
22.	0500:	F5	GRAFICA	PUSH AF
23.	0501:	DD		DEC C ;DEC CONT PUNTOS
24.	0502:	28 02		JR Z,CROH
25.	0504:	F1		POP AF
26.	0505:	C9		RET
27.	0508:	CD 10 06	CROM	CALL ESPACI ;ESPAC. 1/244 PLG
28.	0508:	CD 20 06		CALL COMPRION ;MODO COMPRIMIDO
29.	050C:	F1		POP AF
30.	050D:	CB 3F		SRL A
31.	050F:	C6 04		ADD A,#04 ;OFFSET
32.	0511:	32 C9 12		LD (DATO),A
33.	0514:	21 CA 12		LD HL,DATOANT
34.	0517:	BE		CP (HL)
35.	0518:	28 23		JR Z,IMPRIM1 ;= SALTAR
36.	051A:	38 3D		JR C,IMPRIM2 ;< SALTAR
37.	051C:	21 CA 12		LD HL,DATOANT
38.	051F:	46		LD B,(HL)
39.	0520:	CD 2C 08		CALL BLANCOS ;ESPACIOS EN BCO
40.	0523:	3A C9 12		LD A,(DATO)
41.	0526:	21 CA 12		LD HL,DATOANT
42.	0529:	98		SUB (HL)
43.	052A:	47		LD B,A
44.	052B:	CD 34 08		CALL PUNTOS ;IMP PUNTOS
45.	052E:	CD 21 07		CALL ALIHLI ;ALIMENTAR LINEA
46.	0531:	3A C9 12		LD A,(DATO)
47.	0534:	32 CA 12		LD (DATOANT),A
48.	0537:	CD 3C 08		CALL RSTIMP ;RESETEAR IMPRESORA
49.	053A:	0E 0A		LD C,#14 ;CARGAR CONT
50.	053C:	C9		RET
51.	053D:	21 CA 12	IMPRIM1	LD HL,DATOANT
52.	0540:	48		LD B,(HL)
53.	0541:	05		DEC B
54.	0542:	CD 2C 08		CALL BLANCOS ;ESPACIOS EN BCO
55.	0545:	08 01		LD B,#01
56.	0547:	CD 34 08		CALL PUNTOS ;IMP 1 PUNTO
57.	054A:	CD 21 07		CALL ALIHLI ;ALIMENTAR LINEA
58.	054D:	3A C9 12		LD A,(DATO)
59.	0550:	32 CA 12		LD (DATOANT),A
60.	0553:	CD 3C 08		CALL RSTIMP ;RESETEAR IMPRESORA
61.	0556:	0E 0A		LD C,#14 ;CARGAR CONTADOR
62.	0558:	C9		RET
63.	0559:	21 C9 12	INPRIM2	LD HL,DATO
64.	055C:	3A CA 12		LD A,(DATOANT)
65.	055F:	98		SUB (HL)
66.	0580:	32 C8 12		LD (MENOR),A
67.	0583:	4E		LD B,(HL)
68.	0584:	CD 2C 08		CALL BLANCOS ;ESPACIOS EN BCO

```

89. 0507: 3A C8 12      LD A,(MENOR)
70. 0508A: 47          LD B,A
71. 0508B: CD 34 06    CALL PUNTOS      ;IMP PUNTOS
72. 0508E: CD 21 07    CALL ALIMLI     ;ALIMENTAR LINEA
73. 0571: 3A C8 12      LD A,(DATO)
74. 0574: 32 CA 12      LD (DATOANT),A
75. 0577: CD 3C 08      CALL RSTIMP     ;RESETEAR IMPRESORA
76. 057A: 0E 0A          LD C,#14        ;CARGAR CONTADOR
77. 057C: C9            RET
78. 057D:                ;PROGRAMA PARA FINALIZAR LA
78. 057D:                ;DETECCION DE PICOS.
80. 057D: DB 02          FIN             IN A,(PTOC)    ;SENSAR TECLA
81. 057F: CB 4F          BIT 1,A        ;FIN
82. 0581: 28 05          JR Z,CONTINUA
83. 0583: CB 57          BIT 2,A        ;SENSAR ABORTAR
84. 0585: 28 0B          JR Z,ABORT
85. 0587: C9            RET
86. 0588: CD 47 08      CONTINUA       CALL REBOTES   ;ELIMINAR REBOTES
87. 058B: DB 02          IN A,(PTOC)
88. 058D: CB 4F          BIT 1,A        ;SENSAR FIN
89. 058F: 28 13          JR Z,FIN2
90. 0591: C9            RET
91. 0592: CD 47 08      ABORT          CALL REBOTES
92. 0595: DB 02          IN A,(PTOC)
93. 0597: CB 57          BIT 2,A        ;SENSAR ABORTAR
94. 0599: 28 01          JR Z,ABORT1
95. 059B: C9            RET
96. 059C: C5            ABORT1        PUSH BC         ;RETARDO 1 SEG
97. 059D: CD 00 08      CALL RET1SEG   ;Y REGRESAR AL
98. 05A0: C1            POP BC         ;COMIENZO
99. 05A1: C3 00 00      JP COMIENZO
100. 05A4: 3E 60         FIN2          LD A,#80        ;PONER INDICADORES
101. 05A8: D3 02          OUT (PTOC),A
102. 05A8: DD 38 00 FF   LD (IX+00),$FF;MARCAR EL FIN
103. 05AC: DD 23          INC IX
104. 05AE: 3E 00         REVISION     LD A,#00        ;REVISAR SI
105. 05B0: D9            EXX           ;HUBO PICOS
106. 05B1: B2            OR D          ;SI NO HAY
107. 05B2: D9            EXX           ;IMPRIMIR "NO HUBO
108. 05B3: 28 0E          JR Z,NOHUDA   ;DATOS"
109. 05B5: D9            EXX           ;SI HAY HACER
110. 05B8: 7A            LD A,D        ;CALCULOS
111. 05B7: 32 CC 12      LD (NUMPIC),A
112. 05BA: D9            EXX
113. 05BB: C5            PUSH BC
114. 05BC: CD 00 06      CALL RET1SEG
115. 05BF: C1            POP BC
118. 05C0: C3 89 00      NOHUDA       JP CALCULOS
117. 05C3: CD 53 08      CALL ENCBAB   ;IMP ENCABEZADO
118. 05C8: CD 88 08      CALL MENSAJ   ;IMP "NO HUBO
119. 05C9: C5            PUSH BC       ;DATOS"
120. 05CA: CD 00 08      CALL RET1SEG   ;RETARDO 1SEG
121. 05CD: C1            POP BC

```



```

122. 05CE: C3 00 00                JP COMIENZO ;REGRESAR
123. 05D1:                ;PROGRAMA PARA IMPRIMIR EL
124. 05D1:                ;MENSAJE "LISTO"
125. 05D1: 08 07                LISTO LD B,$07 ;IMPRIME EL
126. 05D3: DD 21 CF OE          LD IX,LISTOPI ;MENSAJE
127. 05D7: DD 7E 00          OTRO LD A,(IX+00) ;"LISTO"
128. 05DA: CD 2C 07          CALL IMPRI
129. 05DD: DD 23              INC IX
130. 05DF: 10 F6            DJNZ OTRO
131. 05E1: CD 21 07          CALL ALIMLI
132. 05E4: C9              RET
133. 05E5:                ;PROGRAMA PARA SENSAR INDICACION
134. 05E5:                ;DE INICIO DE CORRIDA.
135. 05E5: DB 02            SWITCH IN A,(PTOC) ;SENSAR TECLA
136. 05E7: CB 4F            BIT 1,A ;INICIO/FIN
137. 05E9: 20 FA            JR NZ,SWITCH
138. 05EB: CD 47 08          CALL REBOTES ;ELIMINAR REBOTES
139. 05EE: DB 02            IN A,(PTOC) ;VOLVER A SENSAR
140. 05F0: CB 4F            BIT 1,A ;INICIO/FIN
141. 05F2: 20 F1            JR NZ,SWITCH ;SI NO SE TOCO
142. 05F4: C9              RET ;REGRESAR
143. 05F5:                ;PROGRAMA DE RETARDO DE 100 mS.
144. 05F5: 26 64            RETARDO LD H,$64 ;100
145. 05F7: 2E 7C            TIEMPO LD L,$7C ;mSEGUNDOS
146. 05F8: 2D              CUENTA DEC L
147. 05FA: 2D FD            JR NZ, CUENTA
148. 05FC: 25              DEC H
149. 05FD: 20 F8            JR NZ, TIEMPO
150. 05FF: C9              RET
151. 0800:                ;RETARDO DE 1 S.
152. 0800: 0E 04            RET1SRG LD C,$04 ;4 POR
153. 0802: 28 FA            RET240 LD H,$FA ;250
154. 0804: 2E 7C            CARGA1 LD L,$7C ;mSEGUNDOS
155. 0808: 2D              ESPERA2 DEC L
156. 0807: 20 FD            JR NZ, ESPERA2
157. 0809: 25              DEC H
158. 080A: 20 F8            JR NZ, CARGA1
159. 080C: 0D              DEC C
160. 080D: 20 F3            JR NZ, RET240
161. 080F: C9              RET
162. 0810:                ;PROGRAMAR ESPACIAMIENTO EN
163. 0810:                ;IMPRESORA DE 1/244 PLGS.
164. 0810: 3E 1B            ESPACI LD A,$1B ;ENVIAR CODIGO
165. 0812: CD 2C 07          CALL IMPRI ;ASCII PARA PONER
166. 0815: 3E 33            LD A,$33 ;ESPACIAMIENTO DE
167. 0817: CD 2C 07          CALL IMPRI ;1/244 PLGS.
168. 081A: 3E 01            LD A,$01
169. 081C: CD 2C 07          CALL IMPRI
170. 081F: C9              RET
171. 0820:                ;PROGRAMAR IMPRESORA EN
172. 0820:                ;MODO COMPRIMIDO.
173. 0820: 3E 0F            COMPRION LD A,$0F ;PROGRAMAR IMPRESION
174. 0822: CD 2C 07          CALL IMPRI ;COMPRIMIDA

```

175.	0825:	C9		RET	
176.	0828:			;RENOVER MODO COMPRIMIDO DE IMPRESORA	
177.	0828:	3E 12	COMPRIOFF	LD A,\$12	;QUITAR MODO
178.	0828:	CD 2C 07		CALL IMPRI	;COMPRINIDO
179.	082B:	C9		RET	
180.	082C:		;IMPRIMIR	ESPACIOS EN BLANCO.	
181.	082C:	3E 20	BLANCOS	LD A,\$20	;CARGAR ACUM CON
182.	082E:	CD 2C 07		CALL IMPRI	;CODIGO ASCII DE
183.	0831:	10 F9		DJNZ BLANCOS	;ESPACIOS EN BCO
184.	0833:	C9		RET	
185.	0834:		;IMPRIMIR	PUNTOS DECIMALES	
186.	0834:	3E 2E	PUNTOS	LD A,\$2E	;CARGAR ACUM CON
187.	0838:	CD 2C 07		CALL IMPRI	;CODIGO ASCII DE
188.	0839:	10 F9		DJNZ PUNTOS	;PUNTOS DECIMALES
188.	083B:	C9		RET	
190.	089C:		;RESETEAR	LA IMPRESORA	
191.	083C:	3E 1B	RSTIMP	LD A,\$1B	;CARGAR ACUM CON
192.	083E:	CD 2C 07		CALL IMPRI	;CODIGO ASCII DE
193.	0841:	3E 40		LD A,\$40	;RESETEAR
194.	0843:	CD 2C 07		CALL IMPRI	;IMPRESORA
195.	0848:	C9		RET	
198.	0847:		;ELIMINAR	REBOTES CON RETARDO DE 2 mS	
197.	0847:	21 FF 02	REBOTES	LD HL,\$02FF	;RETARDO DE
198.	084B:	F5		PUSH AF	;2 mSEG
198.	084B:	2D	ESPERA3	DEC L	
200.	084C:	20 FD		JR NZ,ESPERA3	
201.	084E:	25		DEC H	
202.	084F:	20 FA		JR NZ,ESPERA3	
203.	0851:	F1		POP AF	
204.	0852:	C9		RET	
205.	0853:		;IMPRIMIR	ENCABEZADO DE REPORTE	
206.	0853:	08 B5	ENCAB	LD B,\$B5	
207.	0855:	DD 21 00 0F		LD IX,TAB1	
208.	0859:	3E 40		LD A,\$40	;INDICADORES
208.	085B:	D3 02		OUT (PTOC),A	
210.	085D:	DD 7E 00	OTROD	LD A,(IX+00)	;CARGAR ACUM
211.	0880:	CD 2C 07		CALL IMPRI	;IMPRIMIR
212.	0863:	DD 23		INC IX	
213.	0885:	10 F6		DJNZ OTROD	
214.	0887:	C9		RET	
215.	0888:		;IMPRIMIR	"NO HUBO DATOS"	
216.	0888:	06 0F	MENSAJ	LD B,\$0F	
217.	088A:	DD 21 D7 0E		LD IX,LNHDT	
218.	088E:	DD 7E 00	OTRO1	LD A,(IX+00)	;CARGAR ACUM
219.	0871:	CD 2C 07		CALL IMPRI	;IMPRIMIR
220.	0874:	DD 23		INC IX	
221.	0878:	10 F8		DJNZ OTRO1	
222.	0878:	CD 21 07		CALL ALIMLI	;IMPRIMIR LINEA
223.	087B:	C9		RET	
224.	087C:		;CONTAR No.	DE PUNTOS DE UN PICO	
225.	087C:	01 00 00	CONTADOR	LD BC,\$0000	;LIMPIAR CONTADOR
226.	087F:	D9	OTROS	RXX	
227.	0880:	7E		LD A,(HL)	;CARGAR ACUM

```

228. 0881: 23          INC HL          ;INC DIRECC
229. 0882: D9          EXX
230. 0883: 03          INC BC          ;INC CONTADOR
231. 0884: FE FF       CP $FF         ;DATO-$FF
232. 0888: 20 F7       JR NZ,OTROS    ;DIFERENTES SALTAR
233. 0888: 08          DEC BC          ;DEC CONTADOR
234. 0889: C9          RET
235. 088A:             ;MULTIPLICAR POR 10 UN DATO
MULT10 LD A,$0A
236. 088A: 3E 0A       LD A,$0A
237. 088C: 21 00 00     LD HL,$0000
238. 088F: CD D1 08     CALL MULTIP
239. 0892: C9          RET
240. 0893:             ;MULTIP. POR 10 UN DATO Y DIVIDIRLO
241. 0893:             ;ENTRE TOAREDI.
DIVID  CALL MULT10
242. 0893: CD 8A 08     LD B,H
243. 0896: 44          LD C,L
244. 0897: 4D          LD DE,(TOAREDI);REC TOAREDI
245. 0898: ED 5B 75 12 CALL DIVISION
246. 089C: CD 00 07     RET
247. 089F: C9          RET
248. 08A0:             ;SUMAR TODAS A1 PARA OBTENER EL TOTAL
SUMARE LD IX,IPINT
249. 08A0: DD 21 79 12 LD HL,NUMPIC
250. 08A4: 21 CC 12     LD B,(HL)
251. 08A7: 46          LD HL,$0000 ;LIMPIAR HL
252. 08A8: 21 00 00     LD D,(IX+00) ;CARGAR DATOS
253. 08AB: DD 58 00     LD E,(IX+01) ;EN DE
254. 08AE: DD 5E 01     ADD HL,DE ;SUMATORIA
255. 08B1: 19          JR C,ARTOGRA ;SALTAR SI A.
256. 08B2: 38 0A       INC IX ;TOTAL ES GRANDE
257. 08B4: DD 23       INC IX
258. 08B8: DD 23       DJNZ OTRAR
259. 08B8: 10 F1       LD (TOTARE),HL;SALVAR A.TOTAL
260. 08BA: 22 77 12   RET
261. 08BD: C9          LD B,$1A ;IMPRIMIR MENSAJE
262. 08BE: 06 1A       LD IX,LATGRA ;"AREA TOTAL MUY
263. 08C0: DD 21 E6 0E LD A,(IX+00) ;"GRANDE"
264. 08C4: DD 7E 00     CALL IHFRI
265. 08C7: CD 2C 07     INC IX
266. 08CA: DD 23       DJNZ OTRO4
267. 08CC: 10 F6       JP COMIENZO
268. 08CE: C3 00 00     ;PROG. PARA MULTIP. DOS
269. 08D1:             ;DATOS DE 8 BITS CADA UNO.
MULTIP LD B,$08
270. 08D1:             ANALIZA SLA A
271. 08D1: 06 08       JR NC,CERO
272. 08D3: CB 27       SLA L
273. 08D5: 30 08       RL H
274. 08D7: CB 25       ADD HL,DE
275. 08D9: CB 14       DJNZ ANALIZA
276. 08DB: 18          RET
277. 08DC: 10 F5       RET
278. 08DE: C9          SLA L
279. 08DF: CB 25       RL H
280. 08E1: CB 14

```

```

281. 08E9: 10 EE          DJNZ ANALIZA
282. 08E5: C9            RET
35. 0000:                ;PROG. PARA DIVIDIR UN No. DE 18 BITS
36. 0000:                ;ENTRE OTRO DE 18 BITS.
37. 0700: 21 00 00     DIVISION LD HL,$0000 ;LIMPIAR REM
38. 0703: 3E 11        LD A,$11
39. 0705: 18 08        JR DIVA
40. 0707: E5          DIVI  PUSH HL
41. 0708: B7          OR A
42. 0709: ED 52        SBC HL,DE ;DE DIVISOR
43. 070B: 38 01        JR C,DIVO
44. 070D: E3          DIVO  EX (SP),HL
45. 070E: E1          DIVO  POP HL
46. 070F: 3F          CCF
47. 0710: CB 11        DIVA  RL C ;BC DIVIDENDO
48. 0712: CB 10        RL B
49. 0714: CB 15        RL L
50. 0716: CB 14        RL H
51. 0718: 3D          DEC A
52. 0719: 20 EC        JR NZ,DIVI
53. 071B: B7          OR A
54. 071C: CB 1C        RR H ;REMANENTE
55. 071E: CB 1D        RR L ;EN HL
56. 0720: C9            RET ;BC RESULTADO
57. 0721:                ;PROG. PARA ALIMENTAR UNA
58. 0721:                ;LINEA EN LA IMPRESORA
58. 0721: 3E 0A        ALINLI LD A,$0A ;LINE FEED
60. 0723: CD 2C 07     CALL IMPRI ;IMPRIMIRLO
61. 0726: 3E 0D        LD A,$0D ;CARRIAGE RETURN
62. 0728: CD 2C 07     CALL IMPRI ;IMPRIMIRLO
63. 072B: C9            RET
64. 072C:                ;ENVIAR DATOS A LA IMPRESORA
65. 072C:                ;E IMPRIMIRLOS
66. 072C: D3 00        IMPRI  OUT (PTOA),A ;DATO A BUFFER
67. 072E: 3E 00        LD A,$00 ;GENERAR EL
68. 0730: D3 02        OUT (PTOC),A ;STROBE
69. 0732: 00          NOP
70. 0733: 00          NOP
71. 0734: 3E 40        LD A,$40
72. 0736: D3 02        OUT (PTOC),A
73. 0738: DB 02        BUSY  IN A,(PTOC) ;SENSAR BUSY
74. 073A: CB 47        BIT 0,A
75. 073C: 20 FA        JR NZ,BUSY ;SI OCUPADA
76. 073E: C9            RET ;ESPERAR
77. 073F:                ;PROG. PARA INTERCAMBIAR DATOS
78. 073F:                ;ENTRE LOS REGISTROS INDICES.
78. 073F: D9          DALE  EXX
80. 0740: 21 CC 12     LD HL,NUMPIC ;CARGAR #PICOS
81. 0743: 48          LD B,(HL)
82. 0744: D9          OTROS EXX
83. 0745: DE 05        LD C,$05
84. 0747: DD 7E 00     LD A,(IX+00) ;CARGAR IY CON
85. 074A: FD 77 00     LD (IY+00),A ;DATO EN IX

```

```

98. 074D: DD 23          INC IX
97. 074F: FD 23          INC IY          ;CARGAR SIGUIENTE
98. 0751: DD 7E 00      LD A,(IX+00)   ;DATO
99. 0754: FD 77 00      LD (IY+00),A
90. 0757: DD 23          INC IX
91. 0759: FD 23          INC IY
92. 075B: 0D            DEC C           ;DEJAR 5 ESPACIOS
93. 075C: 20 FB          JR NZ,MASIY
94. 075E: D9            EXX
95. 075F: 10 E3          DJNZ OTROS
96. 0781: D9            EXX
97. 0782: C9            RET
98. 0783:                ;PROG. PARA CONVERSION HEX-DEC
99. 0783:                ;DESDE DECENAS
ONE LD DE,$000A          ;DIVIDIR DATO HEX
CALL DIVISION          ;ENTRE 10 PARA
CALL CODEAS           ;OBTENER DESDE
CALL CODEASU          ;DECENAS
RET
;PROG. PARA CONVERTIR HEX-DEC
;DESDE DECENAS DE MILLAR
TWO LD DE,$03E8         ;COLOCAR UN CERO
CALL COLCER           ;RECUPERAR DATO
LD DE,$2710           ;10000 (DEC MILL)
CALL DIVISION
CALL CODEAS
LD B,H
LD C,L
LD DE,$03E8           ;1000 (MILLARES)
CALL DIVISION
CALL CODEAS
LD B,H
LD C,L
LD DE,$0064           ;100 (CENTENAS)
CALL DIVISION
CALL CODEAS
LD B,H
LD C,L
LD DE,$000A           ;10 (DECENAS)
CALL DIVISION
CALL CODEAS
CALL CODEASU          ;UNIDADES
RET
;PROG. PARA CARGAR EN BC EL
;DATO DIRECCIONADO POR IY
THREE LD B,$00
LD C,(IY+00)
INC IY
RET
;PROG. PARA CONVERTIR DEC-ASCII
;CUALQUIER DATO MENOS UNIDADES
CODEAS LD A,C           ;CONV DEC-ASCII
CALL DECASC           ;DATO, MENOS

```

```

139. 07B0: C9                RET                ;UNIDADES
140. 07B1:                ;PROG. PARA CONVERTIR DEC-ASCII
141. 07B1:                ;SOLO UNIDADES
142. 07B1: 7D                CODEASU LD A,L      ;CONV DEC-ASCII
143. 07B2: CD B8 07         CALL DECASC       ;DATO, SOLO
144. 07B5: C9                RET                ;UNIDADES
145. 07B8:                ;PROG. PARA CONVERTIR DATOS
146. 07B8:                ;DECIMAL-ASCII
147. 07B8: C8 30           DECASC ADD A,#30   ;CONV DEC-ASCII
148. 07B8: DD 77 00         LD (IX+00),A     ;SUMANDOLE 30H
149. 07BB: DD 23           INC IX           ;AL DATO
150. 07BD: C9                RET
151. 07BE:                ;PROG PARA CARGAR EN BC LOS DATOS
152. 07BE:                ;DIRECCIONADOS POR IY
153. 07BE: FD 46 00         LLAMAR LD B,(IY+00) ;CARGAR DATOS
154. 07C1: FD 4E 01         LD C,(IY+01)    ;A CONVERTIR
155. 07C4: FD 23           INC IY           ;EN BC
156. 07C6: FD 23           INC IY
157. 07C8: C9                RET
158. 07C8:                ;PROG PARA CARGAR LA LOCALIDAD DIRECCIONADA
159. 07C9:                ;POR IX CON UN CERO EN CODIGO ASCII
160. 07C9: 3E 30           COLCER LD A,#30     ;CARGAR ACUM CON
161. 07CB: DD 77 00         LD (IX+00),A    ;CERO PARA IMP
162. 07CE: DD 23           INC IX
163. 07D0: C9                RET
164. 07D1:                ;PROG PARA IMPRIMIR UNA LINEA
165. 07D1:                ;DE LA TABLA DEL REPORTE
166. 07D1: DD 21 CB 10     RESULTAD LD IX,DATIMP
167. 07D5: 3E 18           LD A,#18        ;LIMPIAR BUFFER
168. 07D7: CD 2C 07         CALL IMPRI      ;DE IMPRESORA
169. 07DA: 0E 0C           LD C,#0C        ;12 ESPACIOS
170. 07DC: CD 08 08         CALL ESPACIOS  ;EN BLANCO
171. 07DF: 0E 02           LD B,#02        ;IMP #PICO
172. 07E1: CD 0F 08         CALL IMPRES
173. 07E4: 0E 08           LD C,#08        ;8 ESPACIOS
174. 07E6: CD 08 08         CALL ESPACIOS
175. 07E9: 0E 08           LD B,#08        ;IMP AREA
176. 07EB: CD 0F 08         CALL IMPRES
177. 07EE: 0E 0D           LD C,#0D        ;13 ESPACIOS
178. 07F0: CD 08 08         CALL ESPACIOS
179. 07F3: 0E 06           LD B,#06        ;IMP T.RET
180. 07F5: CD 0F 08         CALL IMPRES
181. 07F8: 0E 11           LD C,#11        ;17 ESPACIOS
182. 07FA: CD 06 08         CALL ESPACIOS
183. 07FD: 0E 05           LD B,#05        ;IMP XAREA
184. 07FF: CD 0F 08         CALL IMPRES
185. 0802: CD 21 07         CALL ALIMLI    ;INPRIMIR TODA
186. 0805: C9                RET                ;LINEA
187. 0808:                ;PROG. PARA IMPRIMIR
188. 0808:                ;ESPACIOS EN BLANCO
189. 0808: 3E 20           ESPACIOS LD A,#20  ;CARGAR ACUM CON
190. 0808: CD 2C 07         CALL IMPRI      ;CODIGO DE ESPACIOS
191. 080B: 0D                DEC C            ;EN BLANCO E IMP

```

```

182. 080C: 20 F8                JR NZ,ESPACIOS
183. 080E: C9                    RET
184. 080F:                        ;PROG. CARACTERES ALMACENADOS A
185. 080F:                        ;PARTIR DE LA LOCALIDAD IX+00
186. 080F: DD 7E 00              IMPRES  LD A,(IX+00) ;IMPRIMIR DATO
187. 0812: CD 2C 07              CALL IMPRI ;EN IX+00
188. 0815: DD 23                INC IX
189. 0817: 10 F8                DJNZ IMPRES
200. 0819: C9                    RET
201. 081A:                        ;PROG. P/CONVERTIR Y COLOCAR EL
202. 081A:                        ;AREA TOTAL
203. 081A: DD 21 BE 10          CONARTOT LD IX,ARTOT
204. 081E: 2A 77 12              LD HL,(TOTARE);CARGAR A. TOTAL
205. 0821: FD 74 00              LD (IY+00),H ;EN IY PARA
206. 0824: FD 75 01              LD (IY+01),L ;CONVERTIRLA
207. 0827: CD 70 07              CALL TWO ;HEX-ASCII
208. 082A: C9                    RET
209. 082B:                        ;PROG P/CONVERTIR Y COLOCAR
210. 082B:                        ;EL % DE AREA TOTAL
211. 082B: DD 21 C4 10          CONFORAR LD IX,ARFOR
212. 082F: 06 00                LD B,$00
213. 0831: 3A D0 11              LD A,(TOTENT) ;RECUPERAR XAREA
214. 0834: 4F                    LD C,A ;ENTEROS Y CONV
215. 0835: 11 84 00              LD DE,$0064 ;HEX ASCII
216. 0838: CD 00 07              CALL DIVISION
217. 083B: CD AC 07              CALL CODRAS
218. 083E: 44                    LD B,H
219. 083F: 4D                    LD C,L
220. 0840: CD 83 07              CALL ONE
221. 0843: DD 36 00 2E          LD (IX+00),$2E;P.DECIMAL EN ASCII
222. 0847: DD 23                INC IX
223. 0849: 06 00                LD B,$00
224. 084B: 3A D1 11              LD A,(TOTFRA) ;RECUPERAR XAREA
225. 084E: 4F                    LD C,A ;FRACC Y CONV
226. 084F: CD 63 07              CALL ONE ;HEXX-ASCII
227. 0852: DD 38 00 25          LD (IX+00),$25
228. 0856: DD 23                INC IX
229. 0858: C9                    RET
230. 0859:                        ;PROG PARA IMPRIMIR EL INICIO
231. 0859:                        ;DE LA LINEA DE TOTALES
232. 0859: 08 17                TOTALES LD B,$17
233. 085B: DD 21 B5 0F          LD IX,TOTALS ;DEJAR ESPACIOS
234. 085F: DD 7E 00              OTRO6 LD A,(IX+00) ;E IMPRIMIR
235. 0862: CD 2C 07              CALL IMPRI ;"TOTALES"
236. 0865: DD 23                INC IX
237. 0867: 10 F6                DJNZ OTRO6
238. 0869: C9                    RET
239. 086A:                        ;PROG. PARA IMPRIMIR EL A. TOTAL
240. 086A:                        ;EN LA LINEA DE TOTALES
241. 086A: 08 08                AREATOT LD B,$08
242. 086C: DD 21 BE 10          LD IX,ARTOT ;LOC. DEL DATO
243. 0870: DD 7E 00              OTRO7 LD A,(IX+00) ;CARGAR ACUM
244. 0873: CD 2C 07              CALL IMPRI ;IMPRIMIRLO

```

```

245. 0876: DD 23          INC IX
246. 0878: 10 F8        DJNZ OTRO7
247. 087A: C8           RET
248. 087B:              ;PROG PARA IMPRIMIR 36
249. 087B:              ;ESPACIOS EN BLANCO
250. 087B: 0E 24        INTER LD C,#24      ;CARGAR C CON 36
251. 087D: CD 08 08    CALL ESPACIOS ;IMP ESPACIOS
252. 0880: C8           RET
253. 0881:              ;PROG PARA IMPRIMIR EL TOTAL
254. 0881:              ;DEL XAREA.
255. 0881: 08 07        PORAR LD B,#07
256. 0883: DD 21 C4 10 LD IX,ARPOR ;RECUPERAR XAREA
257. 0887: DD 7E 00    OTROS LD A,(IX+00) ;IMPRIMIRLO
258. 088A: CD 2C 07    CALL IMPRI
259. 088D: DD 23        INC IX
260. 088F: 10 F8        DJNZ OTROS
261. 0891: C8           RET

```


OPERACION DEL INTEGRADOR

8.1 CONEXION DEL CROMATOGRAFO AL INTEGRADOR

Como es lógico, para que se pueda integrar la señal proveniente del Cromatógrafo, debe existir un medio que la lleve hasta el Integrador; para este fin se utiliza un cable especial, generalmente incluido entre los accesorios del Cromatógrafo que el fabricante entrega al momento de su compra. El cable tiene en uno de sus extremos un conector adecuado para conectarse al Cromatógrafo, el cual guarda la polaridad correcta. Por el otro lado, el cable tiene dos puntas para conectarse a la entrada del Integrador; debe tenerse cuidado en respetar la polaridad indicada, tanto en el cable que viene del Cromatógrafo, como en el conector del Integrador, + el ROJO y - el NEGRO, tal como se muestra en la figura 8.1. También debe tenerse cuidado de que el nivel máximo de la señal entregada por el Cromatógrafo sea de 1 Volt, pues si es mayor, pueden obtenerse resultados erróneos ya que el Integrador solo puede resolver señales entre 0 - 1 Volt. Los Cromatógrafos normalmente tienen sus salidas marcadas con el nivel máximo que entregan, ya que pueden contar hasta con tres diferentes.

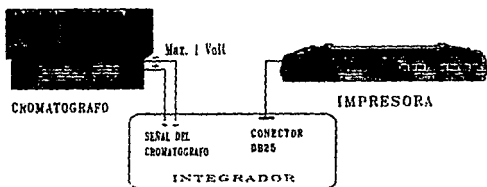


Figura 8.1

8.2 CONEXION DE LA IMPRESORA AL INTEGRADOR

La conexión entre la impresora y el integrador se hace por medio del cable que comunmente utiliza la impresora, conectando por un lado el integrador al conector DB-25 y por el otro la impresora al conector Centronics de 36 pins, tal como se muestra en la figura 8.1. Es importante señalar que la impresora debe utilizar papel continuo, debido a que la impresión del cromatograma no tiene una longitud establecida y una sola hoja puede no ser suficiente para imprimirlo junto con su correspondiente reporte, por lo que no es recomendable utilizar hojas sueltas. En cuanto al tamaño de la hoja, uno más grande que el carta esta sobrado, pues la impresión del cromatograma y el reporte esta programada para ese tipo de hoja.

Tambien cabe señalar que la impresora debe estar programada con el pitch de sus caracteres en modo PICA como se mencionó en un capítulo anterior. Muchas impresoras se programan en ese modo por default, es decir, al ser energizadas automaticamente se programa el modo de los caracteres en PICA.

8.3 COMO REALIZAR UNA CORRIDA

En primer lugar se debe checar que el Cromatógrafo este listo y en condiciones para hacer un analisis, es decir, sus flujos y temperaturas programados deben haber alcanzado su valor y estar estabilizados. Además el cable que conecta al Cromatógrafo con el Integrador y el que conecta a la impresora con el mismo deben estar colocados en forma correcta, como se muestra en la figura 8.1. Tambien debe verificarse que la impresora este cargada con suficiente papel para poder imprimir el cromatograma y la tabla de resultados.

En segundo lugar se procede a encender el integrador y posteriormente la impresora, teniendo cuidado de seguir ese orden, ya que si se hace de modo contrario la impresora puede tener problemas de operación.

En tercer lugar, se ajusta la línea base desde el Cromatógrafo con su correspondiente ajuste de CERO, teniendo cuidado en ponerla a un nivel un poco arriba de 0 mV, con el fin de que si existen picos negativos o la polaridad esta invertida esto pueda detectarse.

En cuarto lugar se inyecta la muestra ya sea manualmente o por medio de un muestreador neumático o eléctrico, e inmediatamente despues se inicia la corrida oprimiendo la tecla INICIO/FIN del Integrador, tal como se muestra en la figura 8.2(a).

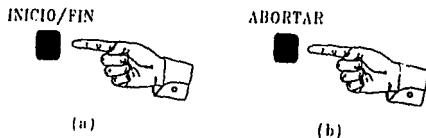


Figura 8.2

Para finalizar la corrida existen dos opciones, una es abortarla oprimiendo la tecla ABORTAR, como se observa en la figura 8.2(b), lo cual permite interrumpir la corrida sin obtener resultados, por ejemplo cuando se ha cometido un error. La otra es oprimiendo la tecla INICIO/FIN por segunda vez, lo que cortará la corrida e imprimirá la tabla de resultados respectiva.

Al terminar de imprimir el reporte o al abortar la corrida el Integrador volverá al inicio del programa imprimiendo el mensaje "LISTO" indicándole al usuario que puede iniciar otra corrida.

Duarnte la impresión del reporte, el Integrador puede enviar el mensaje "AREA TOTAL MUY GRANDE", lo que indica que el

volumen de la muestra inyectada es muy grande y saturó al contador de área, haciendo imposible un cálculo correcto. Para corregir ese detalle, se debe hacer una nueva corrida inyectando un volumen de muestra más pequeño.

Por otro lado, si el Integrador envía el mensaje "NO HUBO DATOS", está indicando que durante el tiempo que duró la corrida no detectó ningún pico.

CONCLUSIONES

Se puede concluir que los objetivos de la tesis se cumplieron satisfactoriamente, pues se logro el diseño del Integrador con las características contempladas inicialmente.

Uno de los logros más importantes es el de poder imprimir el cromatograma y los resultados en una impresora con interface paralelo, por medio del diseño de la interfaz y la programación adecuadas, eliminando con ello la utilización de una impresora que solo funcionara en este Integrador, ya que una del tipo arriba mencionado se encuentra ya en cualquier laboratorio que cuente con una computadora.

CARACTERISTICAS	INTEGRADOR AOI-001	INTEGRADOR COMERCIAL
GRAFICA CROMATOGRAMA	SI	SI
CALCULO DE AREAS Y %AREA	SI	SI
VELOCIDAD DE CARTA VARIABLE	NO	SI
EVENTOS EXTERNOS	NO	SI
RECALCULAR RESULTADOS	NO	SI
USO DE PAPEL ESPECIAL	NO	SI
USO DE TINTA O PAPEL ESPECIAL	NO	SI
DOBLE CANAL	NO	SI

Por otra, parte el costo del Integrador es de

aproximadamente \$ 500.00, mientras que uno comercial tiene un costo mínimo aproximado de 1000 Dls. USA; las diferencias entre uno y otro se muestran en la tabla de la pagina anterior.

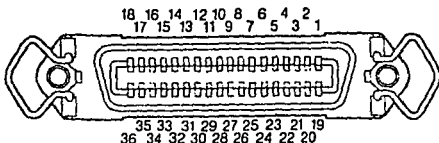
Cabe mencionar que el integrador, como se observa en la tabla, tiene algunas limitantes, pero no obstante puede ser utilizado en aquellos laboratorios que no necesiten de las características de las cuales este adolece, tales como los laboratorios con fines didácticos y otros más.

Por último, el diseño del integrador puede mejorarse utilizando a este como base, pero implementándolo con dispositivos más actuales, tales como los microcontroladores, los cuales le permitirían inclusive, interconectarse con una computadora, con el fin de que la manipulación de los datos tenga mayores posibilidades.

M A P A D E M E M O R I A

- (0000-03FF) Programa principal
- (0400-08FF) Subrutinas de utileria
- (0E00-0FFF) Tablas
- (108B-1000) LOSTPO. Area del Stack Pointer
- (108C-10BD) Area libre
- (10BE-10C9) ARTOT. Loc. donde se salva el AREA TOTAL antes de imprimirse.
- (10C4-10CA) ARPOR. Loc. donde se salva el %AREA antes de imprimirse.
- (10CB-10DE) DATIMP. Loc. para colocar toda una linea de resultados para imprimirse.
- (10DF) IPDACO. Inicio de pila de datos a convertir HEX-ASCII
- (10DF-11CB) IPDACOAR. Pila de areas por convertir.
- (10E1-11CD) IPDACOTR. " T. de Retención por convertir.
- (10E3-11CF) IPDAGOPA. " %Area por convertir.
- (11D0) TOTENT. Loc. para salvar Total Entero
- (11D1) TOTFRA. " " " " Fracciones
- (11D2) SPARENT. " " " Suma de enteros
- (11D3-1223) PARENT. Pila de enteros %Area
- (11D4-1224) PARFRA. " " Fracciones %Area
- (1225-1274) IPAIAT. Pila de At/Ar
- (1275-1276) TOAREDI. Loc. para salvar Ar/10
- (1277-1278) TOTARE. " " " Area Total
- (1279-12C8) IPINT. Pila de integrales
- (12C9) DATO. Loc. para salvar DATO al graficar
- (12CA) DATOANT. Loc. para salvar DATO ANTERIOR al graficar.
- (12CB) MENOR. Loc. para salvar DATO MENOR al graficar.
- (12CC) NUMPIC. Loc. para salvar el No. de picos.
- (12CD-131F) IPTRET. Pila de tiempos de Retención.
- (1320-2FFF) IPDAT. Pila de datos muestreados.

INTERFACE PARALELO GENTRONICS DE IMPRESION



CONECTOR GENTRONICS DE 36 PINS DE LA IMPRESORA

SENALES DESDE LA IMPRESORA

PIN#	SEÑAL	ENT/SAL	DESCRIPCION
1	<u>STROBE</u>	ENT	Pulso para leer datos de entrada. El ancho del pulso debe ser de más de 0.05µs en la terminal receptora.
2	DATO 1	ENT	Dato en paralelo de 8 bits. El 1 y 0 lógicos son ALTO y BAJO respectivamente.
3	DATO 2	ENT	
4	DATO 3	ENT	
5	DATO 4	ENT	
6	DATO 5	ENT	
7	DATO 6	ENT	
8	DATO 7	ENT	
9	DATO 8	ENT	
10	<u>ACKNLG</u>	SAL	Un BAJO indica que se ha recibido el dato y que la impresora esta lista para aceptar más datos. Pulso de 12µs aprox.
11	<u>BUSY</u>	SAL	Un ALTO indica que la impresora no puede recibir datos. La señal es un alto en los

PIN#	SEÑAL	ENT/SAL	DESCRIPCION
			siguientes casos: 1)Durante la entrada de datos 2)Durante la impresión. 3)Cuando la impresora esta fuera de línea. 4)Durante un estado de error de la impresora.
12	PE	SAL	Un ALTO indica que la impresora esta sin papel.
13	SEL	SAL	ALTO si la impresora esta en línea.
14	AUTO FEED XT	ENT	El papel avanza una línea despues de imprimir cuando esta señal es un BAJO.
15			No utilizada
16	GND		Tierra lógica.
17	CHASIS GND		Tierra del chasis de la impresora. separada de la tierra lógica.
18			No utilizada
19-30	GND		Retorno de señal aterrizada de pares retorcidos.
31	INIT	ENT	Un BAJO resetea a la impresora. El ancho del pulso debe ser de más de 50µs.
32	ERROR	SAL	Es un BAJO cuando la impresora esta: 1)Sin papel 2)Fuera de línea 3)En estado de error
33	GND		Igual que los pines 19-30
34			No utilizada

PIN#	SEÑAL	ENT/SAL	DESCRIPCION
35			5 Volts.
36	SLCT IN	ENT	Un ALTO habilita los códigos DC2 y DC3.



CONECTOR DB25 EN EL INTEGRADOR

SENALES DESDE EL SISTEMA

PIN#	SEÑAL	ENT/SAL	PIN#	SEÑAL	ENT/SAL
1	STROBE	SAL	13	SEL	ENT
2-9	D1-D8	SAL	14	A.FEED XT	SAL
10	ACKNLG	ENT	15	ERROR	ENT
11	BUSY	ENT	16	INIT	SAL
12	PE	ENT	17	SEL	SAL
18-25	GND'S				

C O D I G O A S C I I

TABLA DE CODIGOS DE CARACTERES

Dec	Hex	ASCII	Dec	Hex	ASCII	Dec	Hex	ASCII
0	00	NUL	26	1A		52	34	4
1	01	SOH	27	1B	ESC	53	35	5
2	02		28	1C		54	36	6
3	03		29	1D		55	37	7
4	04		30	1E		56	38	8
5	05		31	1F		57	39	9
6	06		32	20	SP	58	3A	:
7	07	BEL	33	21	!	59	3B	;
8	08	BS	34	22	"	60	3C	<
9	09	HT	35	23	#	61	3D	=
10	0A	LF	36	24	\$	62	3E	>
11	0B	VT	37	25	%	63	3F	?
12	0C	FF	38	26	&	64	40	@
13	0D	CR	39	27	'	65	41	A
14	0E	SD	40	28	(66	42	B
15	0F	SI	41	29)	67	43	C
16	10		42	2A	*	68	44	D
17	11	DC1	43	2B	+	69	45	E
18	12	DC2	44	2C	,	70	46	F
19	13	DC3	45	2D	-	71	47	G
20	14	DC4	46	2E	.	72	48	H
21	15		47	2F	/	73	49	I
22	16		48	30	0	74	4A	J
23	17		49	31	1	75	4B	K
24	18	CAN	50	32	2	76	4C	L
25	19		51	33	3	77	4D	M

CODIGO ASCII (CONTINUACION)

78	4E	N	95	5F	_	112	70	p
79	4F	O	96	60	`	113	71	q
80	50	P	97	61	a	114	72	r
81	51	Q	98	62	b	115	73	s
82	52	R	99	63	c	116	74	t
83	53	S	100	64	d	117	75	u
84	54	T	101	65	e	118	76	v
85	55	U	102	66	f	119	77	w
86	56	V	103	67	g	120	78	x
87	57	W	104	68	h	121	79	y
88	58	X	105	69	i	122	7A	z
89	59	Y	106	6A	j	123	7B	{
90	5A	Z	107	6B	k	124	7C	
91	5B	[108	6C	l	125	7D	}
92	5C	\	109	6D	m	126	7E	~
93	5D]	110	6E	n	127	7F	DEL
94	5E	^	111	6F	o			

ESPECIFICACIONES DEL MICROPROCESADOR Z80

Z80-CPU

Absolute Maximum Ratings

Temperature Under Bias	Specified maximum range
Storage Temperature	-55°C to +150°C
Voltage On Any Pin with Respect to Ground	-0.5V to +7V
Power Dissipation	1.5W

*Comments
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note: For Z80EP, pin 41 and 0E characteristics remain the same but the memory pin 40 is input L.

$V_{CC} = 7.0V_{max}$

Z80-CPU D.C. Characteristics

$T_A = 0^\circ C$ to $75^\circ C$, $V_{CC} = 5V \pm 5\%$ unless otherwise specified

Symbol	Parameter	Min	Typ	Max	Unit	Test Condition
V_{IL}	Chip Input Low Voltage	-0.3		0.45	V	
V_{IH}	Chip Input High Voltage			$V_{CC} - 0.3$	V	
V_{OL}	Output Low Voltage	-0.2		0.3	V	
V_{OH}	Output High Voltage	2.0		V_{CC}	V	
I_{OL}	Output Low Current			0.4	mA	$I_{OL} = 10mA$
I_{OH}	Output High Current	2.4			mA	$I_{OH} = -22mA$
I_{CC}	Power Supply Current			150	mA	
I_{IL}	Input Leakage Current			10	μA	$V_{IN} = 0$ to V_{CC}
I_{LOH}	Tri State Output Leakage Current on Bus			10	μA	$V_{OL} = 0.4$ to V_{CC}
I_{LOL}	Tri State Output Leakage Current on Bus			-10	μA	$V_{OH} = 0.4$ to V_{CC}
I_{ID}	Data Bus Leakage Current on Input Mode			110	μA	$0 \leq V_{IN} \leq V_{CC}$

Capacitance

$T_A = 25^\circ C$, $f = 1 MHz$, unmeasured pins returned to ground

Symbol	Parameter	Max	Unit
C_{in}	Chip Input Capacitance	15	pF
C_{IL}	Input Capacitance	1	pF
C_{OUT}	Output Capacitance	10	pF

Z80-CPU

Ordering Information

C - Ceramic
P - Plastic
S - Standard 1V 118° to 70°C
E - Extended 1V 115° to 80°C
M - Military 3V 110° to 125°C

Capacitance

$T_A = 25^\circ C$, $f = 1 MHz$, unmeasured pins returned to ground

Symbol	Parameter	Max	Unit
C_{in}	Chip Input Capacitance	15	pF
C_{IL}	Input Capacitance	1	pF
C_{OUT}	Output Capacitance	10	pF

Z80A-CPU

Ordering Information

C - Ceramic
P - Plastic
S - Standard 1V 115° to 70°C

Z80A-CPU D.C. Characteristics

$T_A = 0^\circ C$ to $75^\circ C$, $V_{CC} = 5V \pm 5\%$ unless otherwise specified

Symbol	Parameter	Min	Typ	Max	Unit	Test Condition
V_{IL}	Chip Input Low Voltage	-0.4		0.45	V	
V_{IH}	Chip Input High Voltage			$V_{CC} - 0.3$	V	
V_{OL}	Output Low Voltage	-0.3		0.4	V	
V_{OH}	Output High Voltage	2.0		V_{CC}	V	
I_{OL}	Output Low Current			0.4	mA	$I_{OL} = 10mA$
I_{OH}	Output High Current	2.4			mA	$I_{OH} = -22mA$
I_{CC}	Power Supply Current			100	mA	
I_{IL}	Input Leakage Current			10	μA	$V_{IN} = 0$ to V_{CC}
I_{LOH}	Tri State Output Leakage Current on Bus			10	μA	$V_{OL} = 0.4$ to V_{CC}
I_{LOL}	Tri State Output Leakage Current on Bus			-10	μA	$V_{OH} = 0.4$ to V_{CC}
I_{ID}	Data Bus Leakage Current on Input Mode			110	μA	$0 \leq V_{IN} \leq V_{CC}$

We reprint data sheets on pages 7-D2 through 7-D13 by permission of Zilog, Incorporated.

Z80A-CPU

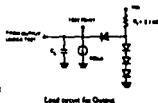
A.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = +5V ± 5%. Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Units	Test Conditions
φ	t ₁ (PH)	Clock Period	25	112.1	μsec	[17] t ₁ = t ₁₍₀₀₀₎ + t ₁₍₀₁₀₎ + t ₁ + t ₁
	t ₂ (PL)	Clock Pulse Width, Clock High	110	111	μsec	
	t ₃ (PL)	Clock Pulse Width, Clock Low	110	109	μsec	
A ₁₅	t ₁ (AD)	Address Output Delay	110	μsec	C _L = 50pF	[18] t ₁ = t ₁₍₀₀₀₎ + t ₁ + t ₁
	t ₂ (AD)	Address Setup Time to RD (Memory Cycle)	111	μsec		
	t ₃ (AD)	Address Setup Time to RD (I/O Cycle)	111	μsec		
	t ₄ (AD)	Address Setup Time to RD (RD or RD)	111	μsec		
	t ₅ (AD)	Address Setup Time to RD or RD During Float	111	μsec		
A ₁₆₋₂	t ₁ (D)	Data Output Delay	130	μsec	C _L = 50pF	[19] t ₁ = t ₁₍₀₀₎ + t ₁ + t ₁
	t ₂ (D)	Delay to Place Data on Bus Cycle	11	μsec		
	t ₃ (D)	Data Setup Time to Rising Edge of Clock During M1 Cycle	11	μsec		
	t ₄ (D)	Data Setup Time to Falling Edge of Clock During M1 Cycle	11	μsec		
	t ₅ (D)	Data Setup Time to RD (Memory Cycle)	11	μsec		
	t ₆ (D)	Data Setup Time to RD (I/O Cycle)	11	μsec		
	t ₇ (D)	Data Setup Time from RD	11	μsec		
t ₈	Hold Time for Setup Time	0	μsec			
RD	t ₁ (RD)	RD Delay From Falling Edge of Clock, RD Low	85	μsec	C _L = 50pF	[20] t ₁ = t ₁₍₀₀₎ + t ₁ + t ₁
	t ₂ (RD)	RD Delay From Rising Edge of Clock, RD High	85	μsec		
	t ₃ (RD)	RD Delay From Falling Edge of Clock, RD High	121	μsec		
	t ₄ (RD)	Pulse Width, RD High	191	μsec		
RD	t ₁ (RD)	RD Delay From Rising Edge of Clock, RD Low	71	μsec	C _L = 50pF	[21] t ₁ = t ₁₍₀₀₎ + t ₁₍₀₁₀₎ + t ₁ + t ₁
	t ₂ (RD)	RD Delay From Falling Edge of Clock, RD Low	85	μsec		
	t ₃ (RD)	RD Delay From Rising Edge of Clock, RD High	85	μsec		
	t ₄ (RD)	RD Delay From Falling Edge of Clock, RD High	85	μsec		
RD	t ₁ (RD)	RD Delay From Rising Edge of Clock, RD Low	85	μsec	C _L = 50pF	[22] t ₁ = t ₁₍₀₀₎ + t ₁ + t ₁
	t ₂ (RD)	RD Delay From Falling Edge of Clock, RD Low	85	μsec		
	t ₃ (RD)	RD Delay From Rising Edge of Clock, RD High	85	μsec		
	t ₄ (RD)	RD Delay From Falling Edge of Clock, RD High	85	μsec		
RD	t ₁ (RD)	RD Delay From Rising Edge of Clock, RD Low	65	μsec	C _L = 50pF	[23] t ₁ = t ₁₍₀₀₎ + t ₁ + t ₁
	t ₂ (RD)	RD Delay From Falling Edge of Clock, RD Low	85	μsec		
	t ₃ (RD)	RD Delay From Rising Edge of Clock, RD High	85	μsec		
	t ₄ (RD)	RD Delay From Falling Edge of Clock, RD High	1101	μsec		
RD	t ₁ (RD)	RD Delay From Rising Edge of Clock, RD Low	100	μsec	C _L = 50pF	[24] t ₁ = t ₁₍₀₀₎ + t ₁ + t ₁
	t ₂ (RD)	RD Delay From Rising Edge of Clock, RD High	100	μsec		
RD	t ₁ (RD)	RD Delay From Rising Edge of Clock, RD Low	130	μsec	C _L = 50pF	[25] t ₁ = t ₁₍₀₀₎ + t ₁ + t ₁
	t ₂ (RD)	RD Delay From Rising Edge of Clock, RD High	130	μsec		
RD	t ₁ (RD)	RD Setup Time to Falling Edge of Clock	70	μsec		
RD	t ₁ (RD)	RD Delay Time From Rising Edge of Clock	300	μsec	C _L = 50pF	
RD	t ₁ (RD)	RD Set-up Time to Rising Edge of Clock	80	μsec		
RD	t ₁ (RD)	Pulse Width, RD Low	80	μsec		
RD	t ₁ (RD)	RD Setup Time to Rising Edge of Clock	30	μsec		
RD	t ₁ (RD)	RD Delay From Rising Edge of Clock, RD Low	180	μsec	C _L = 50pF	[26] t ₁ = t ₁₍₀₀₎ + t ₁₍₀₁₀₎ + t ₁ + t ₁
RD	t ₁ (RD)	RD Delay From Falling Edge of Clock, RD High	180	μsec		
RD	t ₁ (RD)	RD Setup Time to Rising Edge of Clock	80	μsec		
RD	t ₁ (RD)	Delay to Place (RD , RD , RD and RD)	80	μsec		
RD	t ₁ (RD)	RD Setup Time to RD (Interrupt Ack 1)	111	μsec		

NOTES

- Delay should be measured when the CPU does bus when **RD** is active. During interrupt acknowledge phase should be measured when **RD** and **RD** are both active.
- All critical signals are unambiguously synchronized, so they may be loaded up continuously without respect to the clock.
- The **RD** signal must be active for a minimum of 3 clock cycles.
- Charge Delay on Load Capacitance:
T_A = 25°C, V_{CC} = +5V ± 5%
Add Flipping Delay for each 50pF increase to load up to maximum of 200pF for data bus and 100pF for address & control buses.
- Although some by design, timing parameter t₁₍₀₀₀₎ of 200 μsec minimum.



Z80-CPU

A.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = +5V ± 5%, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition
φ	t ₁	Clock Period	4	12.5	μsec	
	t ₁₍₁₀₀₎	Clock Pulse Width (Clock High)	100	150	ns	
	t ₁₍₁₀₁₎	Clock Pulse Width (Clock Low)	100	150	ns	
	t ₁₂	Clock Rise and Fall Time	—	30	ns	
A ₀₋₁₅	t _{2(AS)}	Address Setup Delay	—	100	ns	
	t _{2(AS)}	Delay to Flash	—	110	ns	
	t _{2(A)}	Address Stable From \overline{RD} (RD) (Minimum Credit)	—	100	ns	C _L = 50pF
	t _{2(A)}	Address Stable From \overline{RD} , \overline{RD} or \overline{RD} (RD) (Credit)	—	175	ns	
	t _{2(A)}	Address Stable From \overline{RD} , \overline{RD} , \overline{RD} or \overline{RD} (RD) (Minimum Credit)	—	175	ns	
E ₀₋₇	t _{2(ES)}	Data Output Delay	—	230	ns	
	t _{2(ES)}	Delay to Flash (During Write Cycle)	—	30	ns	
	t _{2(ES)}	Output Setup Time to Rising Edge of Clock (During M1 Cycle)	—	30	ns	C _L = 50pF
	t _{2(ES)}	Data Setup Time to Falling Edge of Clock (During M2 to M7)	—	30	ns	
	t _{2(ES)}	Data Stable From \overline{WE} (Minimum Credit)	—	100	ns	
	t _{2(ES)}	Data Stable From \overline{WE} (1:0 Cycle)	—	100	ns	
	t _{2(ES)}	Data Stable From \overline{WE}	—	175	ns	
t ₂	Any Hold Time for Setup Time	0	—	ns		
RD ₀	t _{2(RD)}	\overline{RD} Delay From Falling Edge of Clock (\overline{RD}) Low	—	100	ns	
	t _{2(RD)}	\overline{RD} Delay From Rising Edge of Clock (\overline{RD}) High	—	100	ns	C _L = 50pF
	t _{2(RD)}	\overline{RD} Delay From Falling Edge of Clock (\overline{RD}) High	—	100	ns	
	t _{2(RD)}	Pulse Width, \overline{RD} Low	—	100	ns	
RD ₀	t _{2(RD)}	\overline{RD} Delay From Rising Edge of Clock (\overline{RD}) Low	—	90	ns	
	t _{2(RD)}	\overline{RD} Delay From Falling Edge of Clock (\overline{RD}) Low	—	100	ns	C _L = 50pF
	t _{2(RD)}	\overline{RD} Delay From Rising Edge of Clock (\overline{RD}) High	—	100	ns	
	t _{2(RD)}	\overline{RD} Delay From Falling Edge of Clock (\overline{RD}) High	—	110	ns	
RD ₀	t _{2(RD)}	\overline{RD} Delay From Rising Edge of Clock (\overline{RD}) Low	—	100	ns	
	t _{2(RD)}	\overline{RD} Delay From Falling Edge of Clock (\overline{RD}) Low	—	100	ns	C _L = 50pF
	t _{2(RD)}	\overline{RD} Delay From Rising Edge of Clock (\overline{RD}) High	—	100	ns	
	t _{2(RD)}	\overline{RD} Delay From Falling Edge of Clock (\overline{RD}) High	—	110	ns	
WE	t _{2(WE)}	\overline{WE} Delay From Rising Edge of Clock (\overline{WE}) Low	—	80	ns	
	t _{2(WE)}	\overline{WE} Delay From Falling Edge of Clock (\overline{WE}) Low	—	90	ns	C _L = 50pF
	t _{2(WE)}	\overline{WE} Delay From Rising Edge of Clock (\overline{WE}) High	—	100	ns	
	t _{2(WE)}	Pulse Width, \overline{WE} Low	—	100	ns	
RD ₀	t _{2(RD)}	\overline{RD} Delay From Rising Edge of Clock (\overline{RD}) Low	—	130	ns	C _L = 50pF
	t _{2(RD)}	\overline{RD} Delay From Rising Edge of Clock (\overline{RD}) High	—	130	ns	
RD ₀	t _{2(RD)}	\overline{RD} Delay From Rising Edge of Clock (\overline{RD}) Low	—	130	ns	C _L = 50pF
	t _{2(RD)}	\overline{RD} Delay From Rising Edge of Clock (\overline{RD}) High	—	130	ns	
RD ₀	t _{2(RD)}	\overline{RD} Setup Time to Falling Edge of Clock	—	70	ns	
RD ₀	t _{2(RD)}	\overline{RD} Delay Time From Falling Edge of Clock	—	200	ns	C _L = 50pF
RD ₀	t _{2(RD)}	\overline{RD} Setup Time to Rising Edge of Clock	—	80	ns	
RD ₀	t _{2(RD)}	Pulse Width, \overline{RD} Low	—	80	ns	
RD ₀	t _{2(RD)}	\overline{RD} Setup Time to Rising Edge of Clock	—	80	ns	
RD ₀	t _{2(RD)}	\overline{RD} Delay From Rising Edge of Clock (\overline{RD}) Low	—	110	ns	C _L = 50pF
	t _{2(RD)}	\overline{RD} Delay From Falling Edge of Clock (\overline{RD}) High	—	110	ns	
RD ₀	t _{2(RD)}	\overline{RD} Setup Time to Rising Edge of Clock	—	90	ns	
RD ₀	t _{2(RD)}	Delay to Flash (\overline{RD} , \overline{RD} , \overline{RD} and \overline{RD})	—	100	ns	
RD ₀	t _{2(RD)}	\overline{RD} Stable From \overline{RD} (Minimum Credit)	—	110	ns	

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)} + t_{2(ES)}$$

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)}$$

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)}$$

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)}$$

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)}$$

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)}$$

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)}$$

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)}$$

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)}$$

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)}$$

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)}$$

$$t_{2(ES)} = t_{2(ES)} + t_{2(ES)}$$

NOTES

A. Data should be loaded into the CPU. Data bus where RD is active. During setup or at the edge of data should be constant where RD and RD are both active.

B. All control signals are normally synchronous. They may be setup asynchronously with respect to the clock.

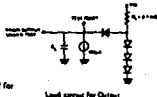
C. The RESET signal must be active for a minimum of 1 clock cycle.

D. Output Delay is Limited's maximum.

E. V_A = 0V, V_{CC} = +5V ± 5%

Add 100 ns delay for each 50pF increase in load up to a maximum of 300pF for the data bus & 100pF for address & control lines.

F. Although not a design timing parameter, t_{2(ES)} is 200 ns maximum.

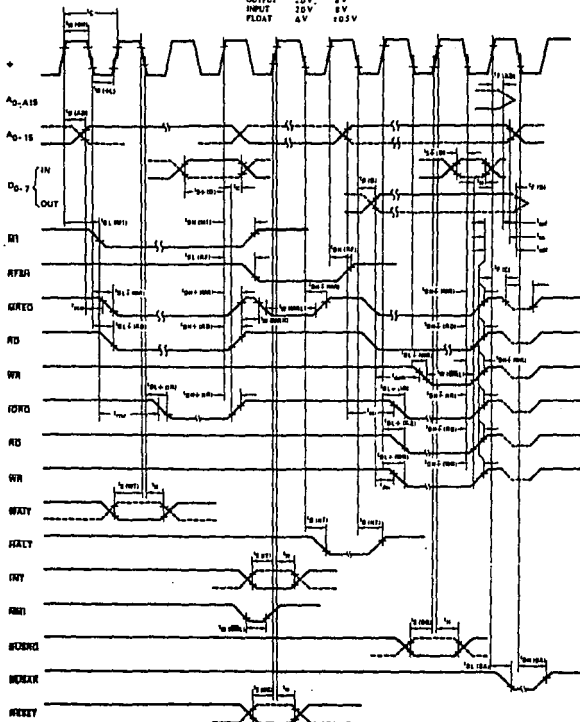


Z80-CPU

A.C. Timing Diagram

Timing measurements are made at the following voltages, unless otherwise specified:

CLOCK	1V	1V
OUTPUT	5V	4.5V
INPUT	1.5V	0.5V
FLOAT	4V	0.5V



BIBLIOGRAFIA

- 1- James W. Coffron
Z80 Applications
SYBEX, INC. 1983

- 2.- Analog Devices
Analog-Digital Conversion Notes
ANALOG DEVICES INC

- 3.- Adam Osborne
An Introduction To Microcomputer Vol. 3
OSBORNE Y ASSOCIATES INC 1979

- 4.- Adam Osborne
An Introduction To Microcomputer Vol. 2
OSBORNE Y ASSOCIATES INC 1979

- 5.- Duoglas V. Hall
Microprocessor And Interfacin. Programmin And Hardware
Mc GRAW HILL 1986

- 6.- Micro Kits
MKE Z80 Manual del usuario. PROGRAMACION
MICROCALLI SA DE CV 1983

- 7.-Lawre Altman
Applying Microprocessors
Mc GRAW HILL 1976

- 8.- Milton H. Aronson
Microprocessors Series. Professional Courses
MEASUREMENTS AND DATA CORP 1980

- 9.- D. P. Burton
Microprocessor System. Hand Book
ANALOG DEVICES, INC 1977
- 10.- Granino A. Korn
Microprocessors And Small Digital Computer Systems for
Engineers And Scientist.
Mc GRAW HILL 1979
- 11.- Texas Instrument
Interface Circuits Data Book
TEXAS INSTRUMENT CORP 1981
- 12.- National Semiconductor
Linear Data Book Vol. 1
NATIONAL SEMICONDUCTOR CORP 1988
- 13.- National Semiconductor
Linear Data Book Vol. 3
NATIONAL SEMICONDUCTOR CORP 1988
- 14.- National Semiconductor
Interface Data Book
NATIONAL SEMICONDUCTOR CORP 1986
- 15.- Intel
MCS-80/85 Family User's Manual
INTEL CORP 1979
- 16.- Texas Instrument
Mos Memory Data Book
TEXAS INSTRUMENT CORP 1986

17.- Luthé, Olivera, Schutz
Métodos Numéricos
LIMUSA 1988

18.- Garcia, Steve
Construya una microcomputadora basada en el Z80
MC GRAW HILL