

12  
28j  
C



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

**FACULTAD DE ESTUDIOS SUPERIORES  
CUAUTITLAN**

**“DESARROLLO DE UN SISTEMA  
DE GRAFICADO BASADO EN  
COMPUTADORA”**

**T E S I S**

**QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA  
P R E S E N T A :  
GERARDO RENE ESPINOSA PEREZ**

**CUAUTITLAN IZCALLI, EDO. DE MEXICO**

**1987**



## **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.

# INDICE

	pag.
INTRODUCCION. ....	I-1
CAPITULO CERO. LA COMPUTADORA HUESPED: FAP-100 ....	1
0.1 La microcomputadora FAP-100 .....	2
0.2 Tarjeta de CPU .....	5
0.3 Tarjeta de memoria .....	7
0.4 Tarjeta de puertos .....	7
0.5 Tarjetas de aplicación .....	10
CAPITULO UNO. SISTEMA CONVERTIDOR DIGITAL / ANALOGICO .....	 12
1.1 Arquitectura del sistema convertidor D/A .....	 13
1.2 Módulo de conversión D/A .....	14
1.3 Latches de 12 bits .....	26
1.4 Lógica de control .....	29
1.5 Puertos .....	33
1.6 Funcionamiento sistema convertidor D/A .....	 37
1.7 Alimentación .....	40

1.8 Construcción .....	42
 CAPITULO DOS. PROGRAMACION SISTEMA DE	
GRAFICADO 12 BITS FAP-100 .....	50
2.1 Programas de graficación .....	51
2.1.1 Posicionamiento en punto	
inicial .....	53
2.1.2 Transmisión de datos .....	60
2.1.3 Posicionamiento final .....	65
2.2 Programas de transformación de datos para graficación	
2.2.1 Algoritmo de Bresenham para segmentos de rectas	71
2.2.2 Algoritmo de Bresenham para círculos	97
2.2.3 Programa generador de rectas y círculos	128
2.3 Programas de transformación	139
2.3.1 Desplazamiento	141
2.3.1 Reducción	156
2.3.3 Amplificación	169
2.3.4 Rotación	178
 CAPITULO TRES. INTERFAZ MAQUINA - USUARIO	
3.1 Características generales	199
3.2 Programas básicos de comunicación	202

3.3 Alternativas disponibles .....	209
3.4 Características particulares de alternativas .....	214
3.4.1 Programa principal .....	215
3.4.2 Programa de modificación .....	219
3.4.3 Generación de tabla de datos .....	223
3.4.4 Cargar tabla .....	242
3.4.5 Almacenar tabla .....	250
3.4.6 Graficar tabla .....	255
3.4.7 Modificación dato a dato .....	256
3.4.8 Escalamiento .....	269
3.4.9 Desplazamiento .....	278
3.4.10 Rotación .....	282
CONCLUSIONES: .....	289
APENDICE A. Listados de programas .....	A-1
APENDICE B. Localidades de servicio para sistema graficador FAP-100 .....	B-1
APENDICE C. Algoritmo para división entera binaria .....	C-1
APENDICE D. Algoritmo de adición múltiple para multiplicación .....	D-1

APENDICE D. Algoritmo de adición múltiple para multiplicación .....	D-1
APENDICE E. Algoritmo para obtención funciones seno y coseno .....	E-1
APENDICE F. Algoritmo para multiplicación entera binaria .....	F-1
APENDICE G. Ejemplos de gráficas .....	G-1
BIBLIOGRAFIA .....	b-1

# INTRODUCCION

Es indudable que el medio de comunicación con el que se logra una asimilación de información más efectiva, es aquel que toma como forma de expresión a las gráficas. Por medio de ellas, la relación tanto entre seres humanos, como entre éstos y el medio ambiente que los rodea, se hace más estrecha, debido a que el entendimiento durante el intercambio de experiencias e ideas se facilita enormemente.

Conocedor de estas ventajas, el ser humano se ha preocupado durante toda su existencia, por mejorar las herramientas de las que se vale para llevar a cabo esta tarea, así, los instrumentos de comunicación gráfica han sufrido grandes transformaciones desde su origen hasta nuestros días.

En la actualidad, uno de los elementos que más se destaca por su eficacia, rapidez y precisión para este tipo de funciones, es la computadora digital.

El uso de este tipo de dispositivos en la generación de gráficas, hoy en día es común, debido a que el alto costo que en sus inicios presentaban, se ha reducido al ser más pequeña la relación precio/beneficio que ofrecen ahora, lo cual, en

conjunción con el desarrollo de elementos de almacenaje muy potentes con costos también muy bajos, permiten la fácil adquisición de un equipo capaz de ejecutar este trabajo con un gasto económico relativamente pequeño.

Aunque en forma general las gráficas por computadora se pueden definir como " la creación, almacenamiento y manipulación de modelos y sus imágenes por medio de una computadora "[1], se pueden delinear dos grandes grupos de éstas, en donde la diferencia principal radica en el grado de control que puede tener el usuario sobre la generación de la imagen de un modelo determinado. Estos dos grupos son:

- Gráficas pasivas. Dentro de este grupo se encuentran todas aquellas gráficas en donde el control del usuario se reduce a la creación del modelo de la gráfica a generar. Ejemplos de este tipo de gráficas son las generadas por impresoras de impacto de alta velocidad, las generadas para películas, televisión, etc.
- Gráficas Interactivas. Es el campo de las gráficas por computadora, en donde el usuario puede controlar las imágenes, y con esto su contenido, formato, tamaño y color, sobre una superficie por medio de dispositivos de interacción, tales como teclado



alfanumérico, palancas de mando (Joystick) o ratón (mouse). Ejemplos de este tipo de gráficas son las generadas por los sistemas CAD/CAM [2], y los programas de simulación y animación actualmente disponibles en el mercado.

El objetivo del presente trabajo, es el de exponer la estructura de los elementos que constituyen al sistema graficador basado en computadora FAP-100, con el cual es posible desarrollar figuras pertenecientes al segundo grupo de los antes mencionados.

En el, se estudian con detalle, los tres subsistemas que lo conforman, y que por otro lado, coinciden con los que posee en general cualquier sistema graficador de este tipo. Estos son:

- \* Dispositivo de entrada.
- \* Computadora, y
- \* Dispositivo de salida.

El orden en que estos son abordados, permite un entendimiento más fácil tanto de su funcionamiento como de la relación que guardan con los otros elementos.

Así, el capítulo primero se dedica a la microcomputadora huesped. En éste, se proporciona un bosquejo general tanto de las características de construcción, como de funcionamiento

de la microcomputadora FAP-100, coordinadora de todas las funciones que el sistema graficador es capaz de realizar.

El segundo capitulo, hace referencia a la circuiteria desarrollada para el sistema, y que corresponde al dispositivo de salida del mismo. Se presentan tanto el diseño, como las características físicas de la interfaz que conecta a la computadora con el elemento graficador. Adicional a esto, se hace referencia a los requerimientos de programación necesarios para que el funcionamiento correcto de este puente de comunicación se lleve a cabo.

En el capitulo dos (tercero de este trabajo), se inicia el estudio de la programación implementada para el sistema, y específicamente se abordan los programas que hacen posible tanto la generación de la imagen de un determinado modelo, como la manipulación del mismo. Se muestran los elementos matemáticos en los que se basan y las características particulares del funcionamiento de cada uno de ellos.

Finalmente, en el último capitulo se presenta lo que podría ser el elemento más significativo con relación al usuario, la interfaz que hace posible la comunicación de éste con la computadora, y que corresponde al dispositivo de entrada del sistema. Dado que este elemento es también implementado en base a programas, la exposición de sus características se realiza de tal forma, que al proporcionar éstas, se capacita al mismo tiempo al lector sobre la forma en que se debe hacer uso de las posibilidades que ofrece el sistema.

# CAPITULO CERO

## LA COMPUTADORA HUESPED FAP-100.

En el desarrollo de un sistema gráfico basado computadora, el primer problema al que se enfrenta el diseñador, inevitablemente, es precisamente al de la selección de ésta: la computadora. Esto se debe a que en base a este dispositivo, todos los demás elementos que componen el sistema llevarán a cabo sus funciones y dependerán de ella para la buena realización de éstas.

Debido al notable desarrollo que se ha alcanzado durante las últimas décadas en las técnicas de fabricación de circuitos integrados, actualmente se cuenta con una clase de dispositivos que, debido a su gran poderío, son capaces de manejar todos los recursos de una computadora, y que además, debido a su construcción, se presentan en un solo encapsulado (circuito integrado). A estos dispositivos se les conoce como Microprocesadores, y a las computadoras que basan su funcionamiento en ellos se les da el nombre de Microcomputadoras.

Este primer capítulo está dedicado precisamente a la

computadora que sirve como directora de todos los elementos del sistema graficador: la microcomputadora FAP-100. En él se expondrán los elementos que la componen y se citarán las principales características de cada uno de ellos.

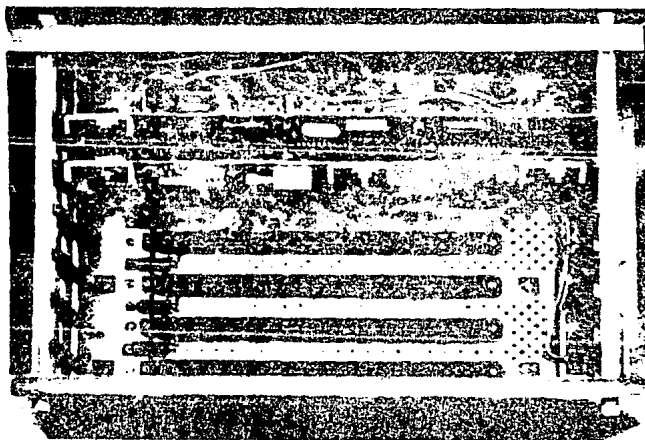
### 0.1 La microcomputadora FAP-100. [3]

La microcomputadora FAP-100, al igual que el sistema graficador que en este trabajo se presenta, se desarrolló en los laboratorios de Física Aplicada pertenecientes a la Escuela Superior de Física y Matemáticas del Instituto Politécnico Nacional, como parte del proyecto de investigación 240/83 "Sistemas Basados en Microprocesador".

FAP (Física APlicada)-100, es una microcomputadora de 8 bits que basa su funcionamiento en el microprocesador Z-80 fabricado por Zilog [4]. El diseño de la máquina se realizó de tal forma, que el resultado fué la construcción de una microcomputadora con estructura modular, esto es, que todos sus elementos son independientes entre sí, y solo se interconectan por medio de la tarjeta madre (motherboard), como se observa en la fotografía 0.1.

El que la máquina se construyera de tal forma, trajo como consecuencia que ésta presentara diversas ventajas sobre máquinas construidas en una sola tarjeta, de las que destacan las referentes al mantenimiento, el cual se realiza mucho más facilmente, y las posibilidades de expansión, a la que se

tiene acceso en forma más libre con respecto al otro tipo de presentación.



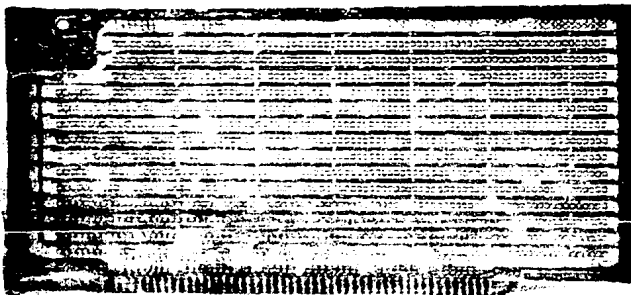
FOTOGRAFIA 0.1.

Microcomputadora modular FAP-100.

La construcción de la microcomputadora se realizó en base a un tipo de bus estandarizado, el Bus -S-100 [5], el cual cumple con la norma IEEE-696, para lo que se hizo necesario que todos los elementos de la máquina se implementaran en tarjetas compatibles con este tipo de bus, y de las cuales la fotografía 0.2 muestra una de ellas.

FAP-100 requiere para su funcionamiento básico de tres

tarjetas fundamentalmente: la tarjeta de CPU, la tarjeta de memoria y la tarjeta de puertos (dispositivos entrada/salida). Además cuenta con algunas otras tarjetas que son utilizadas para aplicaciones específicas de la máquina. A continuación se da una breve descripción de cada una de ellas.



FOTOGRAFIA 0.2.

Tarjeta estandar bus S-100.

## 0.2 Tarjeta de CPU.

En esta tarjeta se encuentra el dispositivo central de la microcomputadora, el microprocesador Z-80, y los circuitos de soporte que requiere para su funcionamiento.

Estos circuitos son:

- Reloj del sistema: basado en un cristal de cuarzo el cual oscila a una frecuencia de 2.5 Mhz. Es el encargado de sincronizar todos los "movimientos" de la computadora.

- Circuito de reset: tiene como función la de inicializar al sistema. Existen dos modalidades para realizar ésta función, reset automático (power on reset), el cual se activa siempre que la máquina se enciende, y reset manual, que inicializa al sistema siempre y cuando se active, no importando en que etapa de funcionamiento se encuentre la máquina. La segunda modalidad se activa por medio de un interruptor y es operado por el usuario.

- Interfaz de buses: tiene como función la de incrementar las posibilidades del microprocesador para manejar cargas (fan out) en cada uno de los tres buses con que cuenta: direcciones, datos y control. Los dispositivos que realizan ésta función son latches uni o bi-direccionales, según lo requiera el bus. Como resultado de la implementación de estas interfaces, el bus de direcciones es capaz de manejar 40 cargas TTL-LS, el bus de datos 10 y el bus de control 10 cargas.

Desde ésta tarjeta se generan también todas las señales

requeridas por los demás dispositivos del sistema para su interconexión con el CPU. Estas son disponibles en forma invertida y no invertida.

TABLA 0.1

Mapa de memoria FAP-100.

LOCALIDADES	TIPO	FUNCION
0000H-0FFFH	EPROM	Almacenamiento sistema operativo y programas de comunicación con dispositivos entrada/salida.
1000H-1103H	RAM	Localidades de servicio del sistema operativo. NO disponibles al usuario.
1104H-3FFFH	RAM	Localidades disponibles al usuario para programas de trabajo.
4000H-BFFFH	--	IMPLEMENTADA
C000H-CFFFH	EPROM	Almacenamiento procesador de textos FAP.
D000H-FFFFH	--	NO IMPLEMENTADA



### 0.3 Tarjeta de memoria.

En esta tarjeta residen los elementos de almacenamiento (memoria) en los que se apoya el microprocesador para su funcionamiento. Existen dos tipos:

- Memoria EPROM (Erasable Programmable Read Only Memory). No pierde la información que almacena cuando es desenergizada la máquina, por lo que se conoce también como no volátil. Es implementada con dispositivos 2716 y 2732.[6]

- Memoria RAM (Random Access Memory). Este tipo de memoria es volátil, esto es, pierde la información que almacena cuando la máquina se desenergiza. Es implementada con dispositivos 2114.[6]

En total se cuenta con 20KBytes de memoria con decodificación completa, lo que da la alternativa de expansión hasta 64KBytes. La tabla 0.1 muestra la distribución, así como la función de la memoria de FAP-100.

En esta tarjeta se encuentran también los circuitos decodificadores que son utilizados por el microprocesador para el acceso a la memoria.

### 0.4 Tarjeta de puertos.

Los circuitos implementados en esta tarjeta son el medio de enlace por el cual FAP-100 tiene la facilidad de establecer comunicación con el mundo exterior. Gracias a

ellos, el microprocesador puede transferir y recibir información de los diferentes dispositivos periféricos con los que tiene contacto. En general son dos clases de circuitos los que residen en la tarjeta: circuitos decodificadores y circuitos de interfaz.

Los circuitos decodificadores tienen la función de seleccionar el canal de comunicación (puerto) por el que la información a transmitir fluita. Debido a la estructura del microprocesador Z-80, FAP-100 goza de un espacio direccionable para la selección de puertos de 256 posibles alternativas, las cuales pueden ser utilizadas como entrada o salida indistintamente.

Los circuitos de interfaz son los encargados de transferir la información desde el exterior al interior o viceversa, es decir, son el puente por el que los datos pasarán de un elemento a otro. Su estructura depende del tipo de comunicación utilizada por la máquina para recibir o mandar información.

FAP-100 es capaz de establecer contacto con dispositivos periféricos en dos modalidades: entrada/salida paralela y entrada/salida serie.

En el modo paralelo, los 8 bits de que consta un dato se transfieren simultáneamente, por lo que para este tipo de comunicación los circuitos de interfaz son latches 74LS273 [7] para puertos de salida e impulsores de tercer estado 74LS125 [7] para puertos de entrada. Este modo ha sido ampliamente usado para comunicar a la máquina con diferentes

tarjetas de aplicación específica.

En modo serie, los 8 bits se tienen que transmitir uno seguido de otro, por lo que se implementó el circuito UART (Universal Asynchronous Receiver and Transmitter) AY-5-1013 [8], el cual recibe un dato en paralelo y lo entrega en serie. En este caso la circuitería de interfaz se desarrollo de tal forma que se tiene opción a dos distintos formatos de transmisión:

- RS-232. Representa un nivel lógico 1(uno) con un voltaje de -12 volts d.c., y un nivel lógico 0(cero) con uno de 12 volts.

- Norma Kansas City. Una señal senoidal de 2400 Hz de frecuencia equivale a un 1(uno) lógico, y una señal de 1200 Hz a un 0(cero) lógico.

Las unidades periféricas que actualmente se comunican con el microprocesador en modo serie son:

- Teclado alfanumérico. Dispositivo de entrada que transmite en formato RS-232 a una velocidad de 9600 bps.

- Pantalla CRT. Dispositivo de salida. Utiliza formato RS-232 a 9600 bps de velocidad.

- Impresora de matriz. Dispositivo de salida. Trabaja a una velocidad de 1200 bps y utiliza el formato RS-232.

- Unidad de cinta magnética. Trabaja como dispositivo de entrada/salida a una velocidad de 300 bps y la transmisión se realiza utilizando la Norma Kansas City.

## 0.5 Tarjetas de aplicación.

Actualmente FAP-100 cuenta con cuatro aplicaciones desarrolladas específicamente para ella, con lo que se incrementa el valor de uso de la máquina.

Los circuitos necesarios para este fin se han desarrollado en diferentes tarjetas con lo que se conserva la construcción modular inicial, a excepción del procesador de textos, citado anteriormente, y el cual reside en un espacio de memoria.

Las tres aplicaciones restantes son:

- Sistema de adquisición de datos de 8 canales simultáneos. Por medio de esta tarjeta (y la programación necesaria también desarrollada), se pueden medir simultáneamente 8 diferentes valores de voltaje en un rango de 0 a 200 volts tanto en D.C. como en A.C. Además el sistema consta de un digitalizador de voz el cual proporciona acústicamente el valor medido en cada uno de los diferentes canales.

- Sistema de comunicación con HP 3000. Por medio de esta tarjeta, FAP-100 tiene la opción de establecer comunicación con la computadora HP 3000, lo que genera la alternativa de contar, entre otras cosas, con un ensamblador residente en la minicomputadora, el cual puede generar directamente en la memoria de FAP-100 código de operación de programas de trabajo.

- Convertidor D/A de 12 bits. Esta tarjeta se desarrolló

para el manejo de un graficador analógico por parte del sistema graficador, tema central de este trabajo. En capítulos posteriores se abordará en forma detallada.

# CAPITULO UNO

## Sistema convertidor digital/analógico.

El graficador analógico es uno de los dispositivos más frecuentemente utilizados en casi cualquier tipo de laboratorio (escolares, de investigación e industriales), para presentar en forma gráfica el comportamiento de diferentes fenómenos. Esto se debe a que el precio de éste es relativamente bajo, y que la estructura del mismo permite su uso en una gama muy amplia de aplicaciones.

Como resultado de su alto grado de accesibilidad y versatilidad, se decidió utilizar un dispositivo de esta clase como elemento graficador en el sistema FAP-100.

Para su inclusión dentro del sistema, se hizo necesario implementar la circuitería que permitiera suministrar la información analógica requerida por éste, a partir de la digital manejada por la microcomputadora. El sistema convertidor digital/analógico, que es tema central del presente capítulo y en el que se expondrán tanto sus características principales como su funcionamiento.

## 1.1 Arquitectura del sistema convertidor D/A.

El sistema convertidor D/A es la interfaz entre la microcomputadora y el graficador analógico. Su función principal es la de recibir un dato en formato digital y transmitir su equivalente en formato analógico, ajustándose a un rango de operación establecido por su construcción interna. Se basa para su funcionamiento en el convertidor D/A de 12 bits DAC 1222 [10] y en circuitería adicional que permite la formación de palabras digitales de 12 bits a partir de datos de 8 bits. La figura 1.1 muestra el diagrama a bloques del sistema, en donde se pueden observar los cuatro módulos principales de los que está formado.

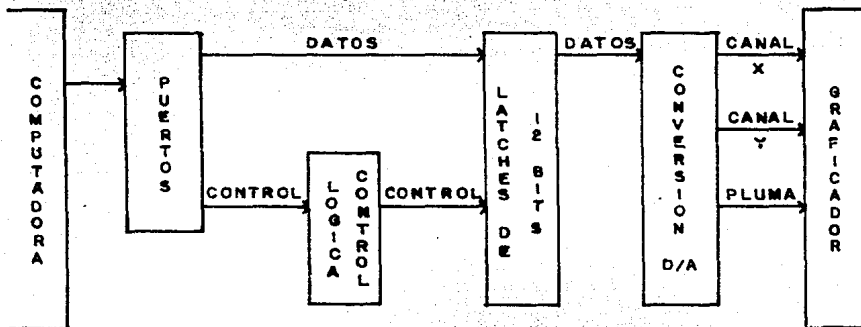


FIGURA 1.1.

Diagrama a bloques del sistema convertidor D/A.

La interfaz está diseñada para poder transmitir

información analógica por dos canales independientes (x o y), y tiene la capacidad de poder controlar la posición de la pluma del graficador (arriba o abajo).

En forma general su funcionamiento es el siguiente:

La información proveniente de la computadora es transportada al sistema convertidor por medio de los puertos de salida y entregada a los latches de 12 bits, si se trata de un dato válido para graficarse, o al módulo de control, si es un parámetro de habilitación. Los latches de 12 bits son los encargados de construir una palabra de esta longitud para posteriormente transferirla al módulo de conversión D/A, mientras que el módulo de control se encarga de seleccionar el canal (x o y) por el que la información fluirá. Finalmente, la palabra formada por los latches de 12 bits, se alimenta al módulo de conversión el cual obtiene el valor equivalente de ésta en formato analógico, dato que servirá como señal de entrada al graficador.

### 1.2 Módulo de conversión D/A.

Para la generación de la señal requerida por el graficador, el módulo de conversión D/A procesa la información digital proveniente de la computadora a través de los cuatro elementos que lo conforman:

- Convertidor digital a analógico.
- Convertidor de corriente a voltaje.



- Circuito inversor de voltaje.
- Filtro pasa-bajas.

El módulo contiene además un quinto elemento, que si bien no procesa la información, si es determinante para que la interfaz sea compatible con el dispositivo graficador. Este elemento es el que genera el voltaje de referencia para el convertidor D/A, y debido al funcionamiento de éste, determina el rango de variación de la señal analógica entregada por la interfaz, como se verá más adelante.

En la sección G2 del diagrama 1.1 se muestra la implementación de éste generador de voltaje. El diseño se hizo en base al circuito MC 1403 [10] fabricado por Motorola, el cual entrega un voltaje constante de 2.5V a partir de uno de entrada que puede variar dentro del rango de 4.5 a 40V. En este caso se alimentó con 5V regulados y se adicionaron dos capacitores como filtros (Cer y Csr), tanto a la entrada como a la salida del circuito. Además se incorporó al potenciómetro Pr, el cual permite variar la alimentación hacia los convertidores. Cuando Pr está en su valor máximo, el voltaje de referencia (Vr) es de 0.9V, mientras que cuando está en su valor mínimo es de 2.5V.

Con lo anterior, se tiene acceso a un área de graficado donde los valores máximos que se pueden alcanzar tanto en el eje x como en el y son Vr, y los valores mínimos son cero volts.

Debido a la naturaleza de la fuente de información, este rango de valores no se puede acceder en su totalidad, sino

que la interfaz generará valores de voltaje discretos, ya que las palabras digitales que dan origen a éstos pueden tomar valores enteros unicamente. Así, en realidad el area de graficado a la que se tiene acceso es una red de puntos en donde cada uno de éstos está dado por el valor de la palabra digital y le corresponde un determinado voltaje dentro de los límites establecidos.

La generación de estos valores de voltaje discretos, es realizada por los cuatro elementos de este módulo, de los cuales el primero recibe una palabra digital y el último entrega la señal analógica equivalente al valor de la información de entrada.

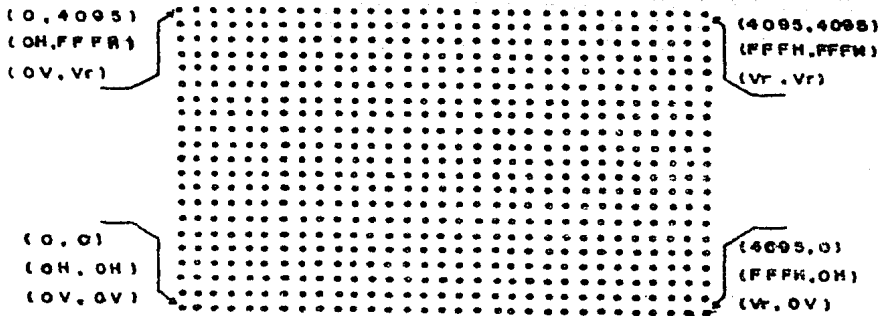


FIGURA 1.2(a).

Area de graficado accesible por sistema convertidor D/A.

El convertidor D/A utilizado en este trabajo es el DAC 1222 [9] fabricado por National. Este circuito maneja

palabras de 12 bits de longitud, por lo que el número de puntos que conforman la red de graficado es de  $2^{12} = 4096$ , y los valores de voltaje generados estarán dados en intervalos de  $V_r/4096$ . La figura 1.2(a) muestra el área de graficado a la que se tiene acceso con la interfaz, mientras que la 1.2(b) muestra la equivalencia entre algunos de los valores que puede tomar la palabra digital y el voltaje generado (para  $V_r=2.5$ ).

PALABRA DIGITAL	EQUIVALENTE DECIMAL	VOLTAJE GENERADO (mV)
0000 0000 0000	0	0.000
0000 0000 0001	1	0.611
0000 0000 1010	10	6.11
0000 0110 0100	100	61.10
0011 1110 1000	1000	611.0
0111 1101 0000	2000	1220.7
1000 0000 0000	2048	1250.0
1011 1011 1000	3000	1831.0
1111 1010 0000	4000	2441.4
1111 1111 1111	4095	2500.0

FIGURA 1.2(b).

Tabla equivalencia digital-analógico para  $V_r = 2.5V$ .

La relación existente entre la palabra digital y el voltaje analógico generado se puede generalizar fácilmente si se hace un análisis detallado del funcionamiento del circuito convertidor D/A. La figura 1.3 muestra la estructura interna de éste, de donde se observa que se trata de un dispositivo

del tipo escalera R-2R invertido.

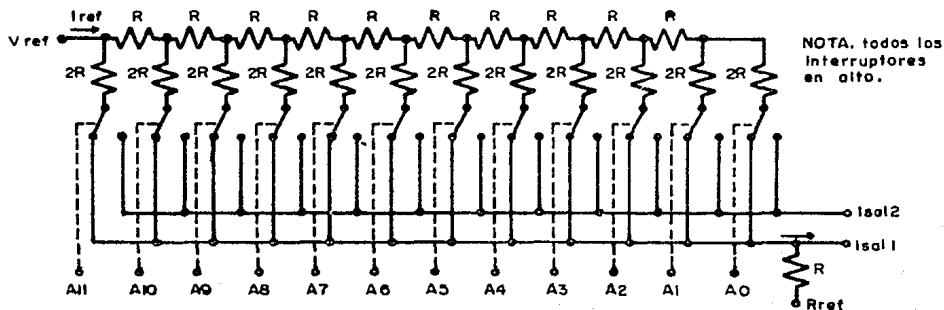


FIGURA 1.3.

Estructura interna convertidor DAC 1222.

Este tipo de convertidor es también conocido como de interrupción de corriente, ya que la señal analógica que genera es de este tipo. Para su implementación dentro de la interfaz, la terminal Isal2 se conectó a tierra, y la información obtenida en la salida Isal1 fue utilizada como alimentación a la siguiente etapa del sistema, mientras que el voltaje de referencia se obtuvo del CI MC 1403.

Conectado de esta forma, y con todos los interruptores en bajo, se obtiene que

$$V_{ref} = I_{ref} \cdot R_{eq}$$

donde, debido a que los valores de las resistencias son R y 2R unicamente

$$Req = R$$

por lo que

$$Vref = Iref \cdot R$$

de donde, despejando

$$Iref = Vref/R \dots\dots (1)$$

Si todos los interruptores están en un estado alto, entonces la corriente que fluye a la salida será

$$\begin{aligned} Isal &= Iref \left( \frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \dots + \frac{1}{4096} \right) \\ &= Iref \left( \frac{4095}{4096} \right) \end{aligned}$$

que es el valor que corresponde a la palabra de mayor valor que se puede obtener con 12 bits (FFFH).

Si todos los interruptores están en bajo, la salida será nula.

Con esto, los valores que puede tomar la señal generada caerán dentro del rango

$$0 \leq Isal \leq \left( \frac{4095}{4096} \right) Iref$$

y la relación general para obtener el valor de la señal analógica para una determinada palabra digital será

$$Isal = Iref \left( A1/2 + A2/4 + A3/8 + \dots + A12/4096 \right)$$

donde A1-A12 son los bits que conforman la palabra digital.

En base a esta relación, los valores intermedios que

podrá tomar la palabra digital generarán las siguientes señales

$(1/4096)I_{ref}$ ,  $(2/4096)I_{ref}$ , ...,  $(4094/4096)I_{ref}$

Las secciones G1 y G3 del diagrama 1.1 muestran la configuración implementada para los convertidores D/A dentro de la interfaz, en donde se observa que además de las características antes citadas, para su funcionamiento requieren de una señal de 12V c.d. como alimentación, y para lo cual los capacitores Cda1 y Cda2 sirven como filtros.

Ya que la naturaleza de la señal generada hasta esta etapa es incompatible con los requerimientos del graficador (ya que éste es alimentado con voltaje y no con corriente), se hizo necesario implementar los circuitos convertidores de corriente a voltaje como siguiente etapa del módulo. La estructura básica de éstos se muestra en la figura 1.4, donde Isal1, Isal2 y Rret son tomadas directamente del DAC.

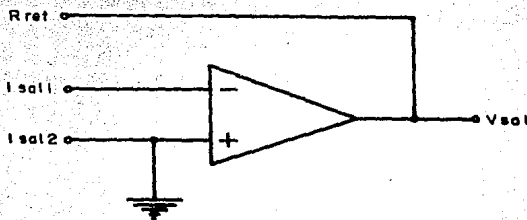


FIGURA 1.4.

Estructura básica convertidor corriente-voltaje.  
Con la inclusión de este dispositivo, la señal analógica

generada estará dada por la relación

$$V_{sal} = -(I_{sal} \cdot R_{ret})$$

donde

$$R_{ret} = R$$

por lo que

$$V_{sal} = -(I_{sal} \cdot R)$$

y dado que  $I_{sal}$  tiene un rango de variación, el voltaje de salida obtenido tomará también diferentes valores. Así, el voltaje máximo será

$$V_{sal} = -(4095/4096)I_{ref} \cdot R \dots\dots (2)$$

y sustituyendo (1) en (2)

$$V_{sal} = -(4095/4096)(V_{ref}/R)R = -(4095/4096)V_{ref}$$

El voltaje mínimo que se puede obtener es cero volts.

Realizando el mismo tipo de sustitución que la anterior, se obtiene la relación general para encontrar el valor del voltaje de salida en base a una determinada palabra digital. Esta es

$$V_{sal} = -V_{ref} (A1/2 + A2/4 + A3/8 + \dots + A12/4096)$$

de donde se obtiene que los valores intermedios serán

$$-(1/4096)V_{ref}, -(2/4096)V_{ref}, \dots, -(4094/4096)V_{ref}$$

que es la relación en la cual se basan los valores mostrados en la figura 1.2(b).

Las secciones H1 y H2 del diagrama 1.1 muestran la implementación de los convertidores C/V para los canales x, y respectivamente. El amplificador operacional utilizado para este fin es el TL 070 [11] fabricado por Texas Instruments, el cual es un dispositivo de bajo nivel de ruido con compensación externa y ajuste de OFF-SET.

La razón fundamental para la selección de este circuito, fue que el intervalo entre dos subsecuentes valores de voltaje generado es muy pequeño, por lo que la disminución de ruido en la señal tenía que ser maximizada.

La compensación externa para estos amplificadores se realiza por medio de los capacitores Cap1 y Cap2 para cada uno de los canales, mientras que el ajuste de OFF-SET se logra variando la calibración de los potenciómetros Papi y Pap2.

Adicionalmente, en la red de realimentación, se incluyó al potenciómetro Pgl en el canal x y Pg2 en el y. El objetivo de éstos, es el de proporcionar una ganancia de voltaje mayor a la obtenida con la sola resistencia interna de realimentación del DAC (Rret). Con esto, cuando Pgl,2 tienen un valor igual a cero, la señal de salida es

$$V_{sal} = - I_{sal} \cdot R_{ret}$$

mientras que cuando tienen un valor distinto de cero, el voltaje generado es



$$V_{sal} = - I_{sal} * (R_{ret} + R_g)$$

El signo menos que aparece en todas las relaciones que hacen referencia al voltaje de salida ( $V_{sal}$ ) se debe a que el convertidor C/V solo existe en la modalidad de inversor. Así, al tener  $I_{sal}$  polaridad positiva, el resultado final será negativo. Esto trajo como consecuencia la implementación del tercer elemento del módulo: El circuito inversor de voltaje.

Este circuito es el encargado de realizar el cambio de polaridad de la señal que recibe sin modificarla en su naturaleza. Esto es, ya que la señal de entrada es un voltaje, la señal de salida tendrá que ser también de la misma naturaleza pero con signo inverso.

La utilización de este dispositivo se hizo necesaria ya que los demás elementos del sistema graficador (programación fundamentalmente), fueron diseñados en base a las características mostradas anteriormente del área de graficado (límite inferior izquierdo coordenadas 0,0 y límite superior derecho coordenadas máximas) por lo que la interfaz D/A tenía que cumplir con éstas forzosamente.

El diseño de estos inversores se realizó de tal forma que la señal de entrada se ve modificada en su polaridad, pero se conservan su amplitud y su naturaleza, esto es

$$|V_{sal}| = |V_s|$$

donde  $V_s$  es la señal de salida de estos circuitos. Esto se hizo posible asignándole a este amplificador una ganancia

igual a uno. Las secciones 11 y 13 del diagrama 1.1 muestran la estructura de estos dispositivos, donde se observa que tanto las resistencias de realimentación de cada uno de ellos ( $R_{r1,2}$ ) como las de entrada ( $R_{e1,2}$ ) son del mismo valor, con lo que se logra la ganancia unitaria, ya que

$$V_s = -(R_{r1}/R_{e1})V_{sal}$$

Debido a que la supresión de ruido en esta nueva señal también tenía que ser maximizada, para su generación se utilizó el circuito TL 072 [10] fabricado por Texas Instruments, el cual contiene dos amplificadores operacionales de bajo nivel de ruido, con compensación interna y sin ajuste de OFF-SET. Estos, al igual que los amplificadores TL 070 se alimentaron con  $\pm 12V$ .

Así, la nueva señal generada tendrá un rango de variación igual que el de  $V_{sal}$  pero con signo inverso, esto es

$$0 \leq V_s \leq I_{sal}(R_{ret} + R_g)$$

Al igual que lo anterior, los valores intermedios son los mismos que  $V_{sal}$  pero con polaridad positiva.

El cuarto elemento del módulo, el filtro pasa-bajas, nace del hecho de la aparición del fenómeno de reflexión (ringing) en cada variación de la señal generada.

Este fenómeno se presenta como una señal oscilatoria de frecuencia muy alta durante un período de inestabilidad antes de alcanzar un nuevo valor deseado. Gráficamente su

comportamiento se observa en la figura 1.5.

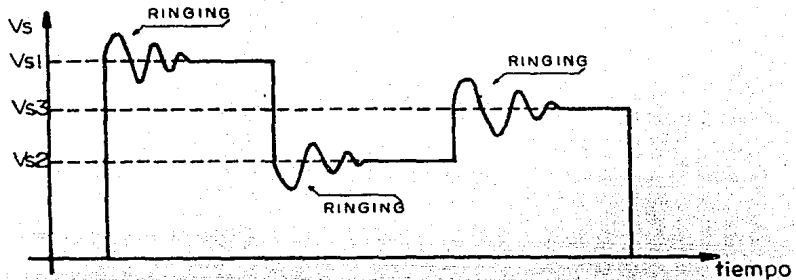


FIGURA 1.5.

Fenómeno de reflexión (ringing) en \$V\_s\$.

El objetivo de este filtro es el de suprimir esta señal no deseada sin modificar la esencia de la información transmitida al graficador. Para su diseño, se tuvo cuidado de elegir la frecuencia de corte adecuada que permitiera eliminar las oscilaciones y al mismo tiempo conservar al máximo las características de la señal de salida de la interfaz.

Las secciones 11 e 13 del diagrama 1.1 muestran la implementación física de estos filtros para el canal x y el y respectivamente. Los valores elegidos para el capacitor \$C\_{f1,2}\$ y \$R\_{f1,2}\$ son tales que la frecuencia de corte del circuito es de aproximadamente 1Mhz, basandose en que la expresión para encontrar ésta es

$$f_c = 1 / (2\pi \cdot R_f \cdot C_f)$$

con lo que se logra la supresión del fenómeno antes citado sin afectar la esencia de la señal de salida Vs.

### 1.3 Latches de 12 bits.

La función principal de este módulo es la generar las palabras digitales de 12 bits de longitud que servirán como información de entrada al módulo de conversión D/A para su posterior procesamiento. La construcción de éstas, se realiza en base a dos palabras de 8 bits (longitud impuesta por la estructura del microprocesador) las cuales son transmitidas secuencialmente al módulo.

La primera de estas palabras contiene en los 6 bits menos significativos la sección baja de la palabra final, mientras que la segunda contiene (en los mismos bits) la sección alta. Gráficamente la construcción de la palabra de 12 bits se observa en la figura 1.6.

Las secciones F1,2 y 3 del diagrama 1.1 muestran la implementación física de este módulo, en donde se puede observar que se divide en dos secciones exactamente iguales (una en F1 y 2, y otra en F2 y 3), la primera para generar la palabra de alimentación al canal x y la segunda para el canal y.

Cada una de estas secciones está constituida a su vez por tres elementos: latch secundario y latch principal alto y bajo. El primero activado por la señal SECx/y y los dos

últimos por la señal PRINx/y.

Los circuitos utilizados para la implementación de este módulo son dispositivos TTL 74LS273 [7] fabricados por National, los cuales están compuestos por 8 flip-flop tipo D utilizados como elementos de almacenamiento, mientras que las señales activadoras son generadas por el módulo de control, el cual se estudia adelante.

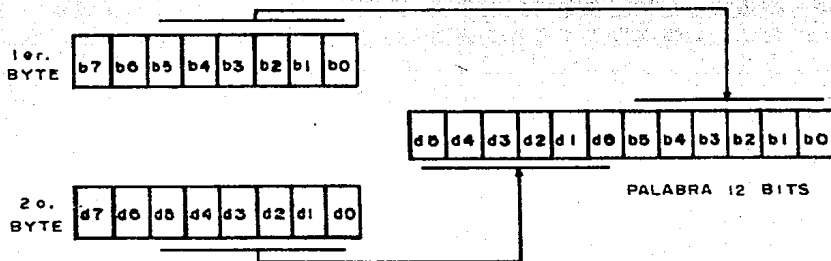


FIGURA 1.6.

Construcción palabra de 12 bits.

Como se observa en el diagrama, el bus de datos se conecta a las entradas del latch secundario y del principal alto directamente en cada uno de los canales, mientras que el principal bajo toma su información de entrada de las salidas del secundario. La información que se transfiere al módulo de conversión D/A se toma entonces de las 6 líneas que contienen los bits menos significativos de cada uno de los latches principales, debido a la estructura de las palabras

originales de 8 bits.

La capacidad que posee este módulo para construir la palabra requerida, depende totalmente del orden en que se recibe la información del bus de datos así como de la secuencia que sigue el módulo de control para activar los diferentes latches. En general, son cuatro las etapas que se pueden distinguir en el proceso de construcción, y de las cuales el orden es inalterable:

1. Se recibe en el bus de datos la primer palabra de 8 bits, la cual es disponible entonces para el latch secundario y para el principal alto.

2. El módulo de control hace activa la señal SECx/y, con lo que la información en el bus es almacenada por el latch secundario y así disponible para el principal bajo.

3. Se recibe la segunda palabra de 8 bits en el bus de datos, la cual es disponible nuevamente para el secundario y el principal alto. Con esto, cada uno de los latches principales tiene acceso a la sección correspondiente de la palabra de 12 bits.

4. Se hace activa la señal PRINx/y y la información disponible a cada latch principal es almacenada, con lo que la transferencia de la información requerida por el módulo de conversión D/A es realizada.

Cabe hacer notar que la secuencia anterior es realizada en forma independiente en cada uno de los canales y no puede llevarse a cabo simultáneamente, esto es, la transferencia por el canal x se hace cuando y está deshabilitado y

viceversa. Las señales que hacen posible esto, son intrínsecas al módulo de control y se estudian a continuación.

#### 1.4 Lógica de control.

El módulo de control engloba a los circuitos que son los encargados de dirigir el flujo de la información a través de los elementos que conforman el sistema convertidor.

Su función principal es la generar las señales de control que activarán los diferentes circuitos que conforman al módulo de latches de 12 bits, para que en base a éstas se contruya los datos con que se alimentará a la etapa de conversión. Además, es el encargado de generar la señal que controla la posición de la pluma del graficador (arriba o abajo).

La información que requiere para poder llevar a cabo esta tarea la recibe del módulo de puertos, el cual está formado por dos de estos: uno de datos (04) y otro de control (03), y que se estudian con detalle posteriormente. En cuanto a ésta, el módulo es capaz de distinguir entre los dos diferentes tipos que pueden existir: una palabra de control, la cual debe ser asimilada por él, y un dato a graficarse, y que debe ser almacenado por los latches de 12 bits.

La capacidad que posee para poder discriminar entre estas dos clases de información, recae en el orden en que

ésta le es transmitida y del puerto por el que recibe el dato.

El módulo de control está constituido por tres submódulos:

- Latch de selección y control de pluma.
- Módulo de habilitación.
- Generador de señal de almacenamiento.

El latch de selección y control de pluma tiene como función la de entregar dos señales lógicas complementarias al generador de señal de almacenamiento, para que en base a éstas se seleccione el canal por el que los siguientes datos a graficarse se procesarán. Además tiene la tarea de almacenar y mantener en un estado determinado el bit que controla la posición de la pluma del graficador.

La sección D1 del diagrama 1.1, en su extremo izquierdo muestra la implementación de este elemento, para el que se utilizó el dispositivo TTL 7474 [7] fabricado por National, y el cual está formado por dos flip-flop tipo D. El primero de estos circuitos genera las señales complementarias via sus salidas Q y Q negada, tomando como información de entrada el estado del bit 0 del puerto de datos. El segundo almacena el bit de control de pluma, el cual es copia directa del bit 1 del mismo puerto. Ambos están activados por la misma señal, procedente del módulo de habilitación.

Este segundo submódulo realiza dos funciones secuenciales: primero habilita ya sea al latch de selección o al canal seleccionado por éste último, y segundo, manda el



pulso que activa al elemento direccionado para el almacenamiento de información. La primer función la realiza en base a los bits 0 y 1 del puerto de control, mientras el pulso de activación lo toma directamente del bit 2 del mismo puerto.

La secciones D2 y D3 del diagrama 1.1 muestran la implementación de este submódulo. Los circuitos utilizados son dispositivos TTL; un 7442 [7] y un 74LS02 [7], ambos fabricados por National. El primero es un decodificador BCD a decimal mientras el segundo son 4 compuertas NO-O (NOR) de dos entradas cada una.

El circuito decodificador tiene como entradas A y B los bits 0 y 1 del puerto de control, mientras que las entradas C y D están en un estado bajo permanentemente. Con esto, solo será capaz de seleccionar las salidas 0 a la 3, cada una de las cuales sirven como una de las entradas de las compuertas NO O en la forma mostrada en el diagrama 1.1. La otra entrada de las cuatro compuertas es común, y se conectan al bit 2 del puerto de control.

Quando una palabra de control válida es transmitida al circuito decodificador, éste activa la salida seleccionada con un estado bajo, mientras que las otras están en alto. Cabe mencionar, que para esta etapa el bit 2 del puerto O3 debe de ser un 1 lógico. Al guardar estos estados las entradas de las compuertas, todas mantendrán un cero a su salida, pero la que se conecta a la salida seleccionada del decodificador solo lo mantendra mientras el estado del bit 2

esté en alto. Con esto, la etapa de habilitación de dispositivo es realizada. La figura 1.7 muestra la tabla de verdad en la que se basa este submódulo para realizar esta tarea.

BO	Bi	DISPOSITIVO HABILITADO
0	0	canal x/y SEC
0	1	canal x/y PRIN
1	0	latch selección
1	1	no implementado

FIGURA 1.7.

Tabla verdad selección habilitada.

Después de que la información es disponible al dispositivo habilitado, entonces el estado del bit 2 es invertido, con lo que la salida de la compuerta seleccionada se convierte en uno lógico, y con esto el almacenamiento se lleva a cabo.

El tercer submódulo, el generador de señal de almacenamiento, tiene como función la de generar las señales SECx/y y PRINx/y, utilizadas por el módulo de latches de 12 bits. Esta tarea la realiza compaginando la información recibida de los dos anteriores submódulos.

La sección E2 del diagrama 1.1 muestra en la parte izquierda la implementación de éste. El circuito utilizado es el 7408 [7] fabricado por National, el cual está compuesto de

cuatro compuertas Y (AND) de dos entradas cada una.

Dos de estas compuertas son utilizadas para generar las señales del canal x por lo que tienen una entrada común conectada a la salida Q negada del latch de selección. Las otras dos se utilizan para el canal y, teniendo como entrada común la salida Q del mismo latch.

En cada par de compuertas, una se encarga de transferir el pulso de almacenamiento al latch secundario del canal correspondiente (señal SECx/y), mientras la otra lo hace a los principales (señal PRINx/y). Por esto, las compuertas secundarias tienen su entrada restante conectada en forma común a la salida SEC del submódulo de habilitación, y las principales a la PRIN del mismo.

Para su funcionamiento, primero recibe la información del latch de selección vía las salidas Q y Q negada, con lo que una de las dos parejas (x o y) queda deshabilitada. Después de esto espera el pulso de almacenamiento del módulo de habilitación, el cual puede ser direccionado hacia el latch secundario o al principal, dependiendo de la etapa en que se encuentre el proceso de construcción de la palabra de 12 bits.

### 1.5 Puertos.

Los circuitos englobados en este módulo trabajan como puente de enlace entre el microprocesador y los demás

elementos de la interfaz. Es por ellos por donde la información proveniente de los registros internos del CPU pasará para ser procesada por el sistema convertidor D/A.

El módulo está formado a su vez por tres submódulos:

- Buffers impulsores.
- Puertos de salida.
- Circuito decodificador.

Los buffers impulsores son los elementos que están situados en cada una de las líneas utilizadas de los tres buses del microprocesador. Tienen la función de incrementar las posibilidades de éste para el manejo de cargas, con lo que se hace posible la comunicación CPU/Interfaz sin poner en peligro al primero. La sección A1 del diagrama 1.1 muestra la implementación de éstos en el bus de datos, mientras que las secciones A2 y A3 muestran los utilizados en el bus de direcciones y control.

Los circuitos utilizados son dispositivos 74LS00 [7] fabricados por National. Estos son 4 compuertas NO Y (NAND) de dos entradas cada una, y las cuales fueron alambradas para que su comportamiento fuera como inversor. El tener esta clase de buffers trajo como consecuencia que tanto datos como direcciones y señales de control fueran tomadas en complemento a 1 para el diseño de los demás elementos del sistema graficador.

El módulo consta de dos puertos, que vistos desde el microprocesador serán de salida: El puerto de datos y el puerto de control. Ambos son implementados con latches

74LS273 [7] fabricados por National, los cuales son habilitados por el circuito decodificador. Su implementación física se muestra en el extremo izquierdo de las secciones C1 y C2 del diagrama 1.1 para el primero y segundo respectivamente.

Por el puerto de datos fluirán en sus 8 líneas los datos que posteriormente formarán las palabras de 12 bits utilizadas por el módulo de conversión. Además en las dos líneas que manejan los bits menos significativos se transmitirán el bit que controla la posición de la pluma y el que controla la selección de canal. Este puerto responde al número 4 dentro del espacio direccionable por el microprocesador.

Del puerto de control únicamente se utilizan los tres bits menos significativos. Los dos primeros generan la palabra que habilita el dispositivo deseado en base a la tabla de la figura 1.7. El tercer bit se utiliza para generar un pulso con el que se activará a éste. El puerto de control es el número 3 dentro del espacio direccionable por el microprocesador.

El circuito decodificador tiene la tarea de habilitar el puerto seleccionado en base a la dirección transmitida por el microprocesador en las líneas del bus que sirve para este fin. La sección A3 del diagrama 1.1 muestra la disposición física de este submódulo. Para su implementación se utilizaron dispositivos TTL 74LS00 [7] (cuatro compuertas NO Y de dos entradas), 7432 [7] (cuatro compuertas O de dos

entradas) y 7442 [7] ( decodificador BCD a decimal).

Su funcionamiento es el siguiente: Cuando se direcciona alguno de los dos puertos, las líneas IORQ y WR irán a un estado bajo, por lo que las salidas de sus respectivos buffers serán unos lógicos. Con esto, la primer compuerta NO Y (NAND) generará un cero a su salida, misma que sirve como entrada a la primer compuerta O (OR). La otra entrada de esta compuerta se define en base a el estado que guardan las líneas A4 y A5 del bus de direcciones, las cuales serán cero siempre que se direcciona alguno de los dos puertos en cuestión, por lo que la segunda entrada de la compuerta O será también cero. Así, la entrada D del decodificador será alimentada con un estado bajo, lo que ocasionará que este circuito sea capaz de activar solamente las salidas 0 a la 7.

Si en bus de direcciones aparece un 03H, entonces los buffers de las líneas A2 a la A7 tendrán en sus salidas unos lógicos, mientras los que corresponden a A0 y A1, cero. Con esto, la entrada A del decodificador será cero y la B uno. La segunda compuerta NO Y genera un cero lógico, es cual es alimentado a una de las entradas de la segunda compuerta O, mientras que la otra entrada de ésta es definida por la salida de la cuarta compuerta NO Y, la cual entrega un uno. Debido a esto, la entrada C del circuito decodificador será uno, por lo que la salida seleccionada será la número 6, la cual por medio de un buffer inversor es conectada a la habilitación del puerto de control.

La misma secuencia se aplica cuando en el bus de

direcciones aparece un 04H, con la diferencia de que la entrada A del decodificador será uno, por lo que la salida 7 es seleccionada. Esta, en forma invertida sirve como señal de habilitación del puerto de datos.

Cuando aparezca cualquier otro valor en el bus de direcciones, la salida seleccionada será una de las que no son utilizadas en el decodificador.

La implementación de los buffers inversores en las señales habilitadoras de puertos, se debe a que éstas son activas en cero, contrariamente a lo requerido por los latches de puertos para su activación.

### 1.6 Funcionamiento Sistema Convertidor D/A.

Esta sección está dedicada a la explicación del funcionamiento de la interfaz D/A en su totalidad. Esto es, el comportamiento de los cuatro módulos que la conforman vistos como un solo elemento.

Para realizar esta exposición se tomarán en cuenta varios aspectos: primero, el funcionamiento del circuito decodificador se da por entendido y solo se citarán las señales que habilitan tanto al puerto de datos como al de control. Segundo, las palabras de control utilizadas para la formación de la información requerida por el módulo de conversión, se citarán en relación al estado que guardan después de que fueron invertidas por los buffers impulsores,

y tercero, la explicación se hace para la transmisión de un dato por el canal x y solo se citarán las variantes necesarias para que ésta se realice por el y.

Tomando en cuenta lo anterior, el funcionamiento es el siguiente:

La secuencia principia mandando por el puerto de datos la palabra que selecciona el canal x y determina la posición de la pluma, FEH para pluma abajo y FFH para pluma arriba. Con esto, las salidas del flip-flop de selección serán Q=1 y Q negada=0, mientras que la de control de pluma es Q=0 para pluma abajo y Q=1 para pluma arriba. Así, las compuertas NO Y del canal y quedan deshabilitadas. Las palabras para el canal y son FCH y FDH respectivamente.

Después de esto, se manda por el puerto de control la palabra que direcciona el latch de selección con la señal de pulso de almacenamiento en un estado bajo. Esta es FDH para ambos canales, y el resultado que se obtiene es que el decodificador del módulo de control activa su salida 2, con lo que la compuerta que maneja la señal LATCH queda habilitada, pero con un estado bajo a su salida debido al nivel del bit que maneja la señal de pulso. Inmediatamente después se manda la palabra que invierte el estado de éste (F9H), con lo que el latch de selección almacena la información disponible a él.

El siguiente paso es habilitar la compuerta NO 0 que maneja la señal SEC, pero manteniendo el pulso de activación en un estado bajo. Esto se logra mandando por el puerto de



control la palabra FEH, con lo que el decodificador del módulo de control activa su salida 1, la cual, en conjunción con el estado del bit de pulso, ocasionan que la compuerta en cuestión genere un cero a su salida. Con esto, el latch secundario del canal x queda direccionado y en espera del pulso de almacenamiento.

En esta etapa, se manda por el puerto de datos la primer palabra de 8 bits que contiene la sección baja de la palabra de 12 próxima a generarse.

Ya que la información es disponible al latch secundario, entonces por el puerto de control se manda la palabra FAH, que produce el pulso de almacenamiento para éste, con lo que la información es disponible ahora para el principal bajo.

Hecho esto, se habilita la compuerta NO O que maneja la señal PRIN via el puerto de control, por el que se transmite la palabra FCH, con lo que el decodificador del módulo de control activa su salida O. Ya que el bit de pulso está en alto, entonces la señal de activación de los latches principales es 0 en ese momento.

La segunda palabra de 8 bits es entonces transferida a la interfaz por el puerto de datos, misma que queda disponible al latch principal alto.

Con la totalidad de la información requerida por el módulo de latches de 12 bits disponible, entonces por el puerto de control se manda la palabra que activa la señal PRINx, con lo que la transferencia al módulo de conversión se realiza. Esta palabra es F8H.

Finalmente, la información pasa por los cuatro elementos de este último módulo y es entregada al graficador en el formato adecuado y con las características necesarias para su asimilación.

Como puede observarse, el orden de la secuencia es inalterable, ya que ésto ocasionaría la construcción de una palabra de 12 bits errónea.

La generación de las palabras de control adecuadas, así como la secuencia de transferencia son controladas por medio de programación, y el análisis detallado de ésta es tema del siguiente capítulo.

### 1.7 Alimentación.

Para el funcionamiento del sistema convertidor D/A, fue necesario implementar tres diferentes niveles de voltaje, los cuales funcionan como alimentaciones a los diferentes dispositivos que conforman a éste.

Estos tres valores son: +5V c.d., +12V c.d. y -12V c.d.

La necesidad de implementar a éstos, se debe al hecho de contar en la interfaz tanto con elementos digitales como analógicos, ya que los primeros requieren del primer valor para su funcionamiento, y los segundos de uno o ambos de los últimos.

La figura 1.8 muestra el diagrama eléctrico par cada uno de estos niveles de voltaje, los cuales toman su señal de

entrada de la fuente que alimenta a las demás tarjetas de la microcomputadora.

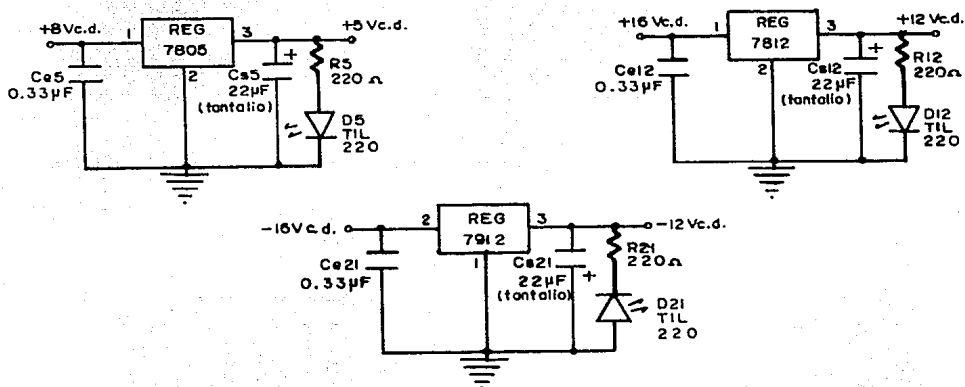


FIGURA 1.8.

Diagrama eléctrico de alimentaciones para sistema D/A.

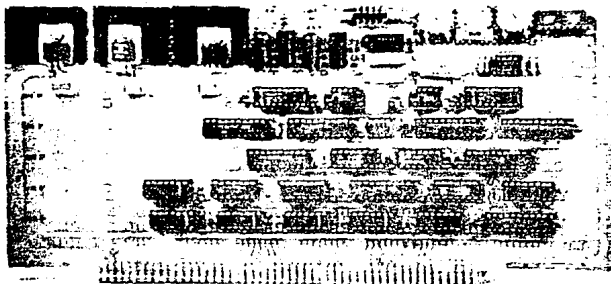
Para su implementación, se utilizaron tres diferentes reguladores: 7805 [12], 7812 [12] y 7912 [12], fabricados por National. El primero para el voltaje de +5V, el segundo para el de +12V y el tercero para -12V. Cada uno de éstos fué montado sobre un disipador de calor construido de aluminio.

Los capacitores utilizados realizan la función de filtraje y estabilización, mientras los diodos emisores de luz son utilizados como testigos.

La disposición física de estas fuentes de alimentación dentro de la tarjeta, se puede observar en la figura 1.9.

## 1.8 Construcción.

Dado que la interfaz D/A forma parte del sistema graficador implementado como dispositivo periférico de la microcomputadora FAP-100, para su construcción física se cumplió con los requerimientos necesarios para su compatibilidad con la misma.



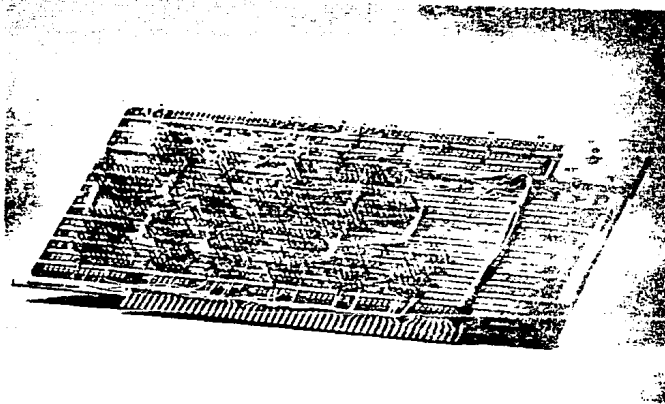
FOTOGRAFIA 1.1.

## Sistema Convertidor digital/analógico.

Así, la tarjeta de circuito impreso en la que se interconectaron sus elementos sigue el estandar para el bus S-100. La fotografia 1.1 muestra a la interfaz D/A, y se observa la tarjeta estandar para este tipo de bus, mientras que la figura 1.9 muestra la disposición de los circuitos

integrados en ella, etiquetandolos de acuerdo al número que les corresponde en el diagrama 1.1.

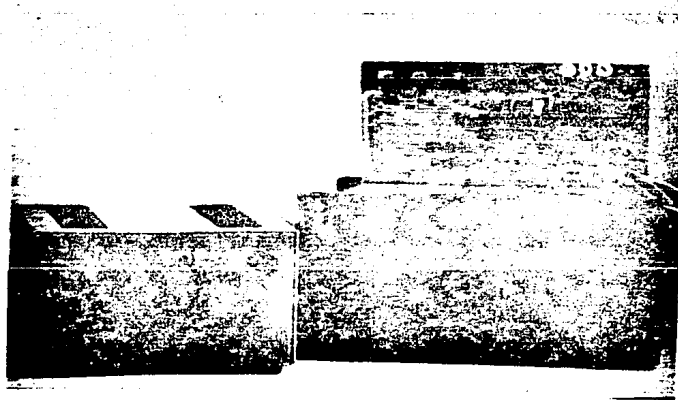
Los dispositivos utilizados se interconectaron siguiendo la técnica de "alambre enrollado" (wire-wrap), la cual se muestra en la fotografía 1.2. Cabe hacer notar que los elementos pasivos fueron soldados directamente a las pistas de la tarjeta.



FOTOGRAFIA 1.2.

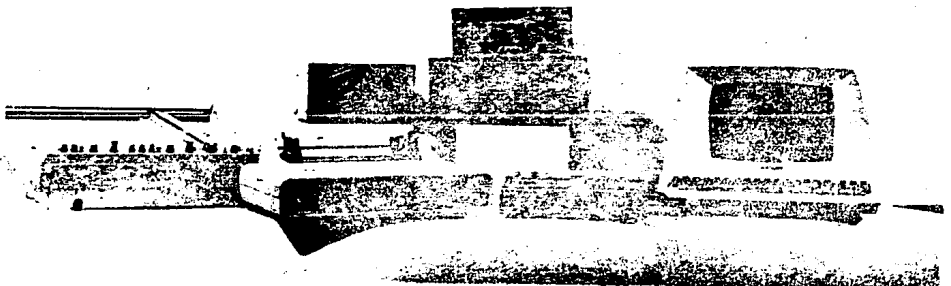
Vista posterior tarjeta interfaz D/A.

Finalmente, la fotografía 1.3 muestra la tarjeta convertidora D/A, montada sobre la microcomputadora FAP-100, mientras que la 1.4 muestra la totalidad del sistema graficador.



FOTOGRAFIA 1.3.

Microcomputadora FAP-100 con fuente de alimentación  
e interfaz D/A.



FOTOGRAFIA 1.4.

Sistema Graficador de 12 bits FAP-100.

En esta última se distinguen el graficador analógico, la terminal entrada/salida, la unidad de almacenamiento, la impresora y la microcomputadora, sobre la cual descansa la interfaz, con su fuente de alimentación.

LISTA DE CIRCUITOS INTEGRADOS CORRESPONDIENTE  
A DIAGRAMA 1.1 Y FIGURA 1.9.

CI 1	MC 1403
CI 2,1	DAC 1222
CI 4,5	TL 070
CI 6	TL 072
CI 7,8,9,10,11,12	74 LS 273
CI 13	74 74
CI 14	74 42
CI 15	74 LS 02
CI 16	74 08
CI 17,18	74 LS 273
CI 19,20,21,22,23,24	74 00
CI 25	74 32
CI 26	74 42



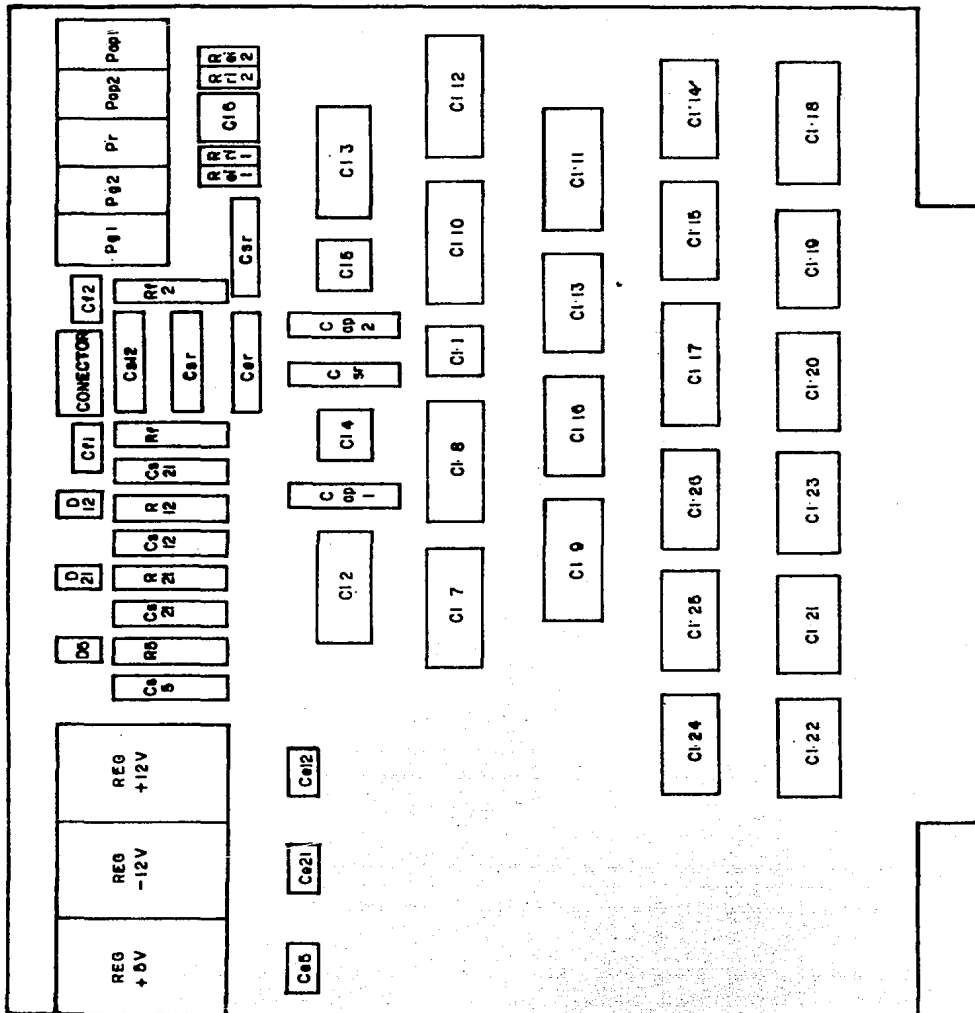


FIGURA 1.9.

Disposición circuitos sobre tarjeta interfaz D/A.

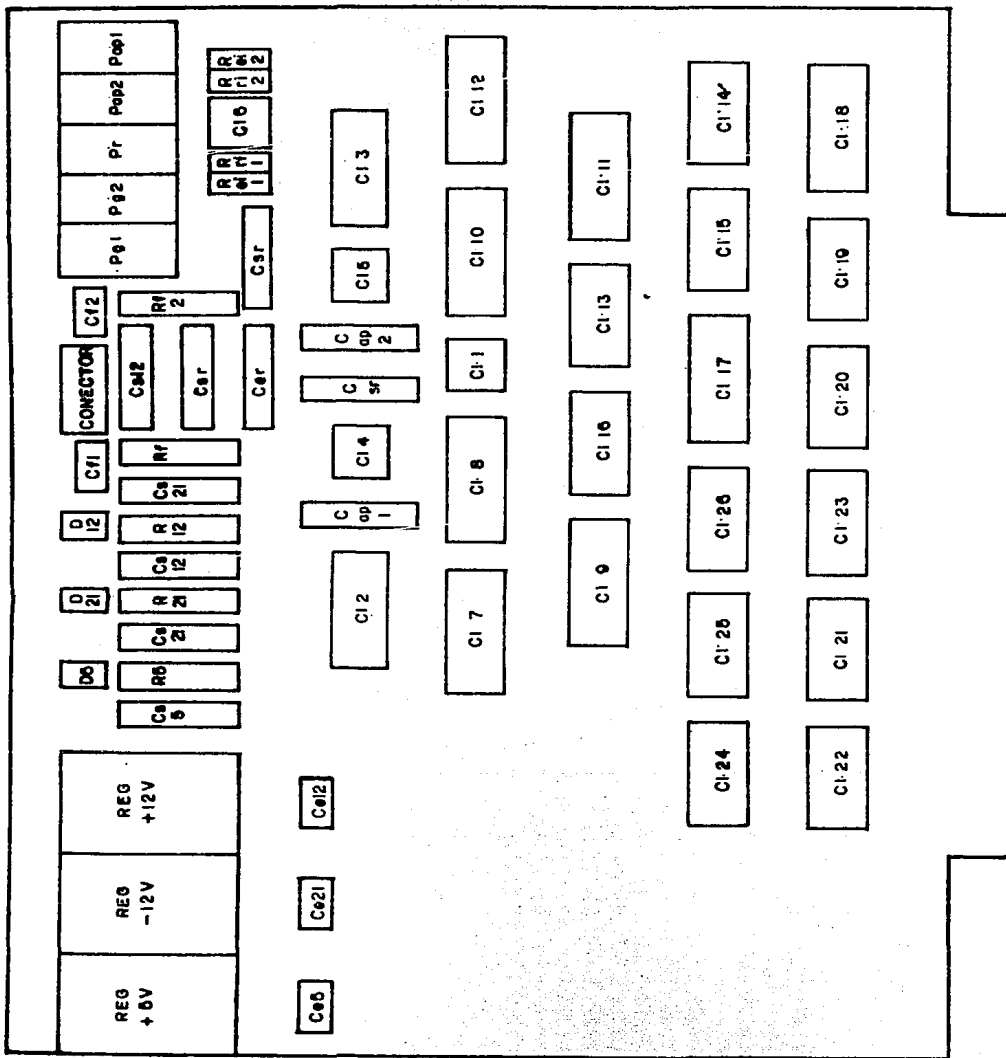


FIGURA 1.9.

Disposición circuitos sobre tarjeta interfaz D/A.

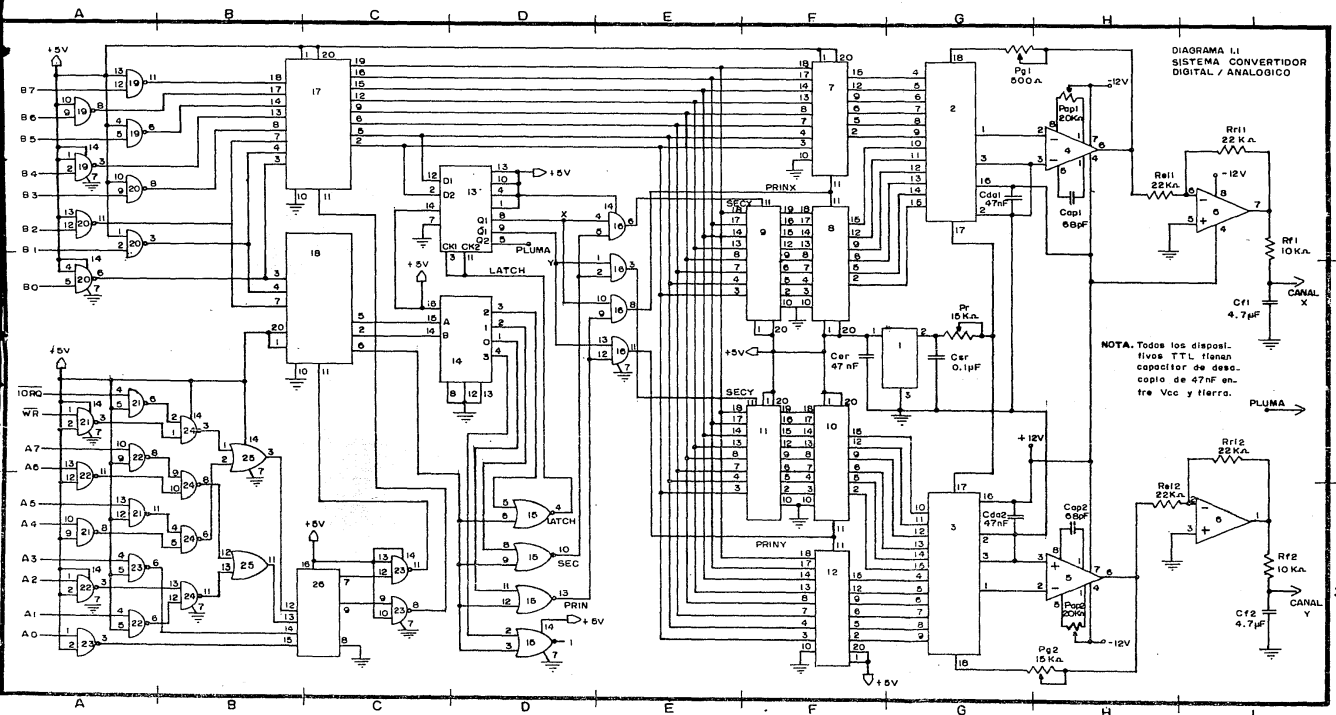


DIAGRAMA I.1  
SISTEMA CONVERTIDOR  
DIGITAL / ANALOGICO

NOTA: Todos los dispositivos TTL. Hacer capacitor de desacoplo de 47nF entre Vcc y tierra.

PLUMA →

CANAL Y

CANAL X

## CAPITULO DOS

### Programación Sistema de graficado 12 bits FAP-100.

Una de las principales características de un sistema graficador semejante al desarrollado en el presente trabajo, es que la computadora, elemento principal del mismo, es la que maneja todos los recursos disponibles en él. Es por esto que para realizar esta función, se requiere además de la circuitería necesaria, de un segundo elemento, sin el cual el sistema sería por demás inútil: la programación.

En este capítulo, se hace un análisis detallado de los programas desarrollados para que la microcomputadora FAP-100 fuera capaz tanto de manejar los diferentes elementos del sistema (convertidor D/A, interfaz máquina-usuario), como de manipular la información suministrada por el usuario para su posterior presentación gráfica.

En general, este conjunto de programas se divide en cuatro diferentes tipos, agrupados de acuerdo a la función que realizan:

- de graficación.
- de generación de datos para graficado.

- de transformacion.
- de comunicacion con usuario.

los cuales fueron realizados en su totalidad en lenguaje ensamblador para el microprocesador Z-80.

A continuacion se exponen los programas agrupados en los tres primeros tipos, dejando el estudio del cuarto como tema del siguiente capitulo.

## 2.1 Programas de graficacion.

Los programas que conforman este grupo, son los encargados de establecer la comunicacion entre la microcomputadora y la interfaz D/A. Es por medio de estos, que se logra la transferencia de informacion para su procesamiento y finalmente su presentacion grafica.

La estructura de cada uno de ellos, esta estrechamente ligada al funcionamiento del sistema convertidor D/A, pues llevan a cabo la generacion de la serie de palabras de control requeridas por la interfaz (vease seccion 1.6) en el orden adecuado, para el buen funcionamiento de esta. Adicionalmente, realizan tareas como la de actualizar el bit que controla la posicion de la pluma del graficador (arriba o abajo).

En general, se basan para su funcionamiento en los datos que reciben de los algoritmos de graficacion (que se estudian en la siguiente seccion), aunque tambien toman directamente

de la tabla de datos proporcionada por el usuario, algunos parámetros.

Con respecto a esta última, bastara decir ahora, ya que el estudio detallado de ella se realiza posteriormente, que se trata de una serie de datos con una estructura específica y en la cual residen las coordenadas de los puntos extremos de los elementos a graficarse, así como palabras de control que establecen acciones como: tipo de trazo (recta o círculo), posición de la pluma (arriba o abajo) y fin de archivo.

Para la recepción de la información a transmitirse, los programas hacen uso de dos localidades de servicio (CBUFY y BUFY), en las cuales, los programas generadores de datos depositan las coordenadas de punto próximo a graficarse. Cabe hacer notar que para la explicación del funcionamiento de estos programas, se partirá del hecho de que los datos listos a graficarse están almacenados con anterioridad en las localidades de servicio, y que estos solo tendrán que ser accedados. El mecanismo de almacenamiento corresponde a los programas generadores de datos, y por lo tanto, su estudio se realiza posteriormente.

Las tareas que realizan este grupo de programas, son tres fundamentalmente:

- Accesa las coordenadas del punto inicial del archivo de datos, y transmite los parámetros necesarios para que la pluma del graficador se posicione en ellas sin imprimir el trazo.

- Recibe las coordenadas de los puntos necesarios para la construcción de los elementos gráficos y los transmite al sistema convertidor D/A para que la pluma del graficador se posicione en ellos conforme le son entregados.

- Cuando el archivo es finalizado, transmite los parámetros necesarios para que se levante la pluma del graficador y se posicione en el extremo superior derecho del área de graficado (CASA).

Adicionalmente, los programas deben tomar en cuenta otros tipos de factores como son: la inercia que presenta el graficador para su posicionamiento, por lo que genera retardos en la transmisión de la información para hacer compatible la velocidad del sistema con la del dispositivo. Y el formato en que los datos deben ser entregados a la interfaz, esto es, en complemento a 1 y en dos secciones (véase sección 1.3 ), por lo que deben ser capaces de realizar esta transformación.

A continuación se hace una explicación detallada de cada una de las tareas que realizan estos programas. Además, se expone la forma en que las llevan a cabo y se cita la programación implementada para este fin.

#### 2.1.1 Posicionamiento en punto inicial.

Para la realización de esta tarea, se hace uso de la subrutina diseñada específicamente para cubrir este objetivo,

y cuyo nombre es POS (POSiciona).

Los pasos que sigue ésta durante su funcionamiento son:

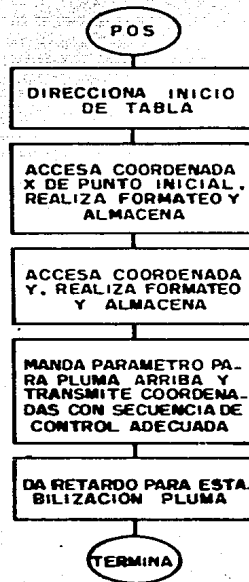
Direcciona primeramente el inicio de la tabla de datos suministrada por el usuario, con lo que automáticamente tiene acceso a las coordenadas del punto primero del trazo requerido. Por medio de otro apuntador, direcciona la localidad de servicio utilizada para almacenar la coordenada X (BUFX) del punto próximo a graficarse, y con esto toma este dato del archivo, lo procesa para acondicionarlo al formato adecuado para su compatibilidad con el sistema D/A y lo guarda en el buffer antes citado. Con el mismo apuntador que utilizó para la coordenada X, apunta a la localidad que almacenará la Y del punto a graficarse (BUFY), toma la información de la tabla de datos y realiza el formateo requerido. El dato con nueva estructura es guardado en el buffer correspondiente.

Ya que cuenta con las coordenadas iniciales en el formato adecuado y almacenadas apropiadamente, entonces se procede a su transferencia hacia la interfaz D/A. En esta etapa, la primera acción es transmitir el parámetro requerido para que se levante la pluma, para posteriormente mandar la secuencia de palabras de control, junto con los datos, en el orden apropiado. Es necesario aclarar, que dentro de este proceso, se encuentran intercalados los retardos necesarios para que la inercia del graficador no afecte el posicionamiento de la pluma. Estos, están calculados de tal forma, que la acción se lleva a cabo sin contratiempos.



Después de que se finalizó con la transmisión de los datos, se genera un retardo adicional para permitir a la pluma del graficador estabilizarse en la posición adecuada, con lo que el proceso termina.

El diagrama a bloques de esta subrutina se muestra a continuación:



Para su implementación, se desarrollaron las subrutinas necesarias para la realización de tareas independientes, las cuales conjuntadas, logran el objetivo planteado. Las características principales de cada una de éstas se listan a continuación, mientras que para un estudio más detallado de

ellas, se puede consultar el apéndice A de este trabajo, en donde se expone el listado completo. La dirección y función específica de las localidades de servicio utilizadas, se encuentran en el apéndice B.

\* NOMBRE: POS.

FUNCION: Posiciona la pluma del graficador en el punto indicado por las coordenadas iniciales de la tabla de datos. Esta acción la lleva a cabo sin imprimir el trazo.

DIRECCION: 13A0H.

SUBROUTINAS QUE UTILIZA: PDATO, TRAPO, RPOS.

DATOS ENTRADA: Dirección inicial del archivo en localidad DIR1. Tabla de datos a partir de esa dirección.

DATOS SALIDA: Coordenadas de punto inicial en formato adecuado en localidades BUFX y BUFY. Transmisión de éstos a sistema D/A.

REGISTROS QUE MODIFICA: A, HL, BC, IX.

\* NOMBRE: PDATO.

FUNCION: Procesa dato para configurarlo en formato adecuado para su compatibilidad con sistema D/A (ver sección 1.6) y lo almacena.

DIRECCION: 147CH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a formatearse apuntado por HL.

Localidad donde se almacena dato con estructura nueva apuntada por IX. \

DATOS SALIDA: Dato en formato adecuado almacenado en localidades apuntadas por IX+0 y IX-1.

REGISTROS QUE MODIFICA: A, HL.

\* NOMBRE: TRAPO.

FUNCION: Transmite las coordenadas del punto inicial del trazo requerido, en formato adecuado, al sistema convertidor D/A, dando los retardos necesarios para que no influya la inercia del graficador.

DIRECCION: 13FAH.

SUBROUTINAS QUE UTILIZA: SACAL, RTDO.

DATOS ENTRADA: Datos a transferirse en localidades BUFX y BUFY.

DATOS SALIDA: Datos por puerto requerido por interfaz D/A.

REGISTROS QUE MODIFICA: A, DE, BC, IX.

\* NOMBRE: SACAL.

FUNCION: Proporciona a la interfaz D/A las palabras de control necesarias para que realice el procesamiento de datos. Esto lo lleva a cabo en el orden adecuado y mandando la información por el puerto correcto. Además transmite el parámetro de control de pluma y los datos mismos. Genera también un retardo que se ocupa para que la pluma del graficador se estabilice en la posición

marcada.

DIRECCION: 1417H.

SUBROUTINAS QUE UTILIZA: RPOS, MANDA.

DATOS ENTRADA: Bit de control de pluma en el espacio menos significativo del acumulador. Dato a transmitirse apuntado por IX.

DATOS SALIDA: Palabras de control y datos hacia sistema convertidor D/A.

REGISTROS QUE MODIFICA: A, D, E.

\* NOMBRE: RPOS.

FUNCION: Genera un retardo cuya duración depende de un parámetro de entrada. Puede variar desde 22.8 microsegundos hasta 17.7 Horas.

DIRECCION: 14CCH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Parámetro que define la duración del retardo en HL.

DATOS SALIDA: Ninguno.

REGISTROS QUE MODIFICA: A, BC, HL.

\* NOMBRE: MANDA.

FUNCION: Es encargado en parte de generar las palabras de control para la interfaz D/A. Manda los datos a procesarse y chequea si éstos se salen de los límites establecidos por el área de graficado. No se puede utilizar si no se llama a continuación de

SACA o SACAl.

DIRECCION: 143BH.

SUBROUTINAS QUE UTILIZA: RANGO, RTDO.

DATOS ENTRADA: Dato a procesarse apuntado por IX.

DATOS SALIDA: Palabras de control y dato hacia interfaz  
D/A.

REGISTROS QUE MODIFICA: A, D.

\* NOMBRE: RANGO.

FUNCION: Revisa si el dato listo a ser procesado rebasa los limites establecidos por el Area de graficado, y si esto sucede en que sentido se da. Esto lo realiza en ambos ejes. Si el número es menor que el limite inferior, manda el valor de este limite sin modificar al dato. Si es mayor que el superior, manda el valor de este último sin modificar al dato.

DIRECCION: 14A6H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a ser procesado apuntado por IX.

DATOS SALIDA: Dato válido, o limite, transferido a la interfaz D/A.

REGISTROS QUE MODIFICA: A.

\* NOMBRE: RTDO.

FUNCION: Genera un retardo cuya magnitud depende de un parámetro de entrada. Puede variar desde 15.2

microsegundos hasta 1.5 milisegundos.

DIRECCION: 14EFH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Parámetro que define la duración del retardo en A.

DATOS SALIDA: Ninguno.

REGISTROS QUE MODIFICA: B: Afecta sus bits los datos de entrada

### 2.1.2 Transmisión de datos.

La segunda función que realizan los programas presentados en esta sección, es la referente a la transmisión de datos para la construcción de los elementos gráficos. Para realizar esto, los programas no tienen capacidad alguna para analizar a que tipo de trazo pertenece la información suministrada, sino que su trabajo se restringe únicamente a transportarlos hacia la interfaz D/A, en el formato adecuado y sin modificarlos, a menos que el valor de éstos caiga fuera de los límites establecidos por el área de graficado, caso en que mandara el valor límite de ésta sin modificar el dato en cuestión.

Para llevar a cabo esta tarea, se hace uso de la subrutina TRANS. (TRANSMite), la cual fué desarrollada específicamente para este fin, y cuya estructura es básicamente la misma que la de POS, aunque a diferencia de ésta, los datos que transfiere no necesitan ser formateados,

sino que los recibe ya con la construcción requerida. Estos se almacenan en las localidades BUFX y BUFY, donde son depositados por los algoritmos de graficación, y de donde deberán ser tomados para su transferencia al sistema convertidor.

Otras diferencias con respecto a POS son: TRANS toma en cuenta el estado del bit que controla la posición de la pluma (arriba o abajo) en cada transferencia que realiza, y los retardos para compensar la inercia del graficador son menores.

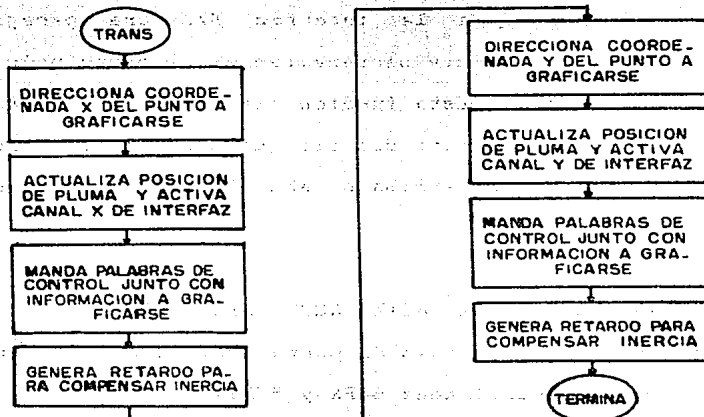
En general, los pasos que sigue entonces esta subrutina son:

Direcciona por medio de un apuntador, la localidad que almacena la coordenada X del punto próximo a graficarse (el cual debe estar en el formato adecuado). Checa el bit de control de pluma, el cual es el menos significativo de la palabra de control almacenada en la localidad PLUMA, con lo que estructura el byte que será mandado a la interfaz D/A y cuyo efecto será el de activar el canal X y simultáneamente actualizar la posición de la pluma. Hecho esto, transmite hacia el sistema convertidor, la serie de comandos que son requeridos por este último para el procesamiento de la información, la cual es transferida intercalada en estos. Da un retardo para compensar la inercia del graficador y direcciona con el mismo apuntador que utilizó para la coordenada X, la Y del punto en cuestión.

Teniendo este dato disponible, actualiza la posición de

la pluma y al mismo tiempo activa el canal Y del sistema D/A. Transmite la serie de palabras de control junto con la información a graficarse y finalmente genera un nuevo retardo como compensación a la inercia del graficador.

Con esto, el diagrama a bloques de esta subrutina, es el que se muestra a continuación



Esta subrutina es utilizada fundamentalmente por los algoritmos de graficación, y puede verse como un submódulo de éstos, ya que al ser recursivos (como se verá adelante), requieren transferir un dato cada vez que éste es calculado, para lo que esta subrutina es invocada.

Para un estudio detallado de la estructura de TRANS, se debe consultar el apéndice A de este trabajo, donde se expone el listado completo de ella y otras subrutinas en las que se basa para llevar a cabo su tarea. A continuación se muestran



solamente las características principales de las mismas. Adicionalmente, el apéndice B muestra la dirección y función específica de cada una de las localidades de servicio utilizadas.

\* NOMBRE: TRANS.

FUNCION: Transmite a la interfaz D/A una pareja de coordenadas x,y pertenecientes al punto próximo a graficarse. Esta función la realiza tomando en cuenta el estado del bit que controla la posición de la pluma (arriba o abajo) y la inercia del graficador.

DIRECCION: 13DBH.

SUBROUTINAS QUE UTILIZA: ACTX, ACTY, SACA, RTD01.

DATOS ENTRADA: Coordenadas de punto próximo a graficarse en localidades BUFx y BUFy.

DATOS SALIDA: Información y palabras de control hacia interfaz D/A.

REGISTROS QUE MODIFICA: A, B, D, IX.

\* NOMBRE: ACTX.

FUNCION: Construye la palabra de control con la que activa el canal X de la interfaz D/A y se actualiza la posición de la pluma del graficador (arriba o abajo).

DIRECCION: 1462H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Palabra cuyo bit menos significativo determina la posición de la pluma en localidad PLUMA.

DATOS SALIDA: Palabra que activa canal X y contiene información de pluma en A.

REGISTROS QUE MODIFICA: A.

\* NOMBRE: ACTY.

FUNCION: Construye la palabra de control con la que activa el canal Y de la interfaz D/A y se actualiza la posición de la pluma (arriba o abajo).

DIRECCION: 146FH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Palabra cuyo bit menos significativo determina la posición de la pluma en localidad PLUMA.

DATOS SALIDA: Palabra que activa canal Y y contiene información de pluma en A.

REGISTROS QUE MODIFICA: A.

\* NOMBRE: SACA.

FUNCION: Proporciona a la interfaz D/A las palabras de control requeridas para el procesamiento de datos y los datos mismos. Adicionalmente, entrega el parámetro que controla la posición de la pluma. Para su utilización se debe invocar primero a ACTX o ACTY.

DIRECCION: 142DH.

SUBROUTINAS QUE UTILIZA: MANDA.

DATOS ENTRADA: Bit de control de pluma en espacio menos significativo de A. Dato a transmitirse apuntado por IX.

DATOS SALIDA: Palabras de control y datos hacia interfaz D/A.

REGISTROS QUE MODIFICA: A, B, D.

\* NOMBRE: MANDA (explicada anteriormente).

DIRECCION: 143BH.

\* NOMBRE: RTDO1.

FUNCION: Genera un retardo cuya duración depende del parámetro de entrada. Puede variar entre 12 microsegundos hasta 2.5 mS.

DIRECCION: 14E3H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Parámetro que define duración en A.

DATOS SALIDA: Ninguno.

REGISTROS QUE MODIFICA: BC.

### 2.1.3 Posicionamiento final.

La tercera y última función que realizan este tipo de programas, es la del posicionamiento que se realiza cuando se

alcanza el final de un archivo.

Para realizar esto, se invoca a la subrutina desarrollada específicamente para llevar a cabo esta acción, y cuyo nombre es CASA. El objetivo de ésta, es posicionar la pluma del graficador en el extremo superior derecho del Área disponible de graficado sin realizar trazo alguno, esto es, con la pluma levantada. Este punto es denominado como casa, de donde proviene el nombre de la subrutina.

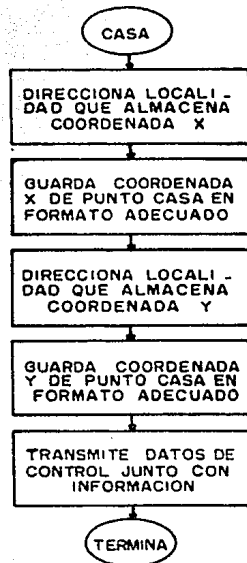
Para llevar a cabo esta tarea, la subrutina direcciona la localidad que almacena la coordenada X del punto próximo a graficarse y la carga con el valor máximo que se puede alcanzar en el eje correspondiente, con el formato requerido por el sistema D/A. Después de esto, hace lo mismo con la coordenada Y para finalmente transmitir los parámetros requeridos para el procesamiento de la información, tomando en cuenta que el bit de control de pluma se fuerza al estado que determina la posición "arriba" de ésta.

La ilustración siguiente muestra el diagrama a bloques de esta subrutina.

Para su implementación, se hizo uso de la subrutina TRAPO desarrollada para POS, de la cual se expuso con anterioridad sus características principales. Por medio de ésta, se logra el posicionamiento en el punto deseado y sin que se realice trazo alguno.

Para un estudio detallado de la subrutina CASA, se puede consultar el apéndice A de este trabajo, donde se muestra el listado completo de ésta, mientras que en el B se expone la

función y dirección de cada una de las localidades de servicio utilizadas.



A continuación se proporcionan las características principales de esta subrutina.

\* NOMBRE: CASA.

FUNCION: Transmite a la interfaz D/A los parámetros necesarios para posicionar a la pluma del graficador en el extremo superior derecho del Área de graficado sin realizar el trazo.

DIRECCION: 13BFH.

SUBROUTINAS QUE UTILIZA: TRAPO.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Datos y palabras de control hacia interfaz D/A.

REGISTROS QUE MODIFICA: A, B, D, IX.

\* NOMBRE: TRAPO ( explicada anteriormente).

ADIRECCION: 13FAH.

## 2.2 Programas de generación de datos para graficado.

Los programas incluidos en este grupo, son los encargados de generar los datos que serán transportados, via el conjunto de programas de graficación, a la interfaz D/A para su procesamiento y finalmente su presentación gráfica. Es por esto, que de ellos depende la capacidad del sistema para la producción de figuras geométricas, objetivo final del mismo.

Para la construcción de estas figuras, los programas generadores de datos se basan fundamentalmente en dos algoritmos matemáticos diseñados especialmente para fines de graficación:

- El algoritmo de Bresenham para segmentos de recta
  - y el algoritmo de Bresenham para círculos
- los cuales, debido a su naturaleza; son completamente compatibles con el área de graficado disponible.

El objetivo de estos algoritmos, es el de generar las coordenadas de los puntos necesarios para la construcción ya sea de un segmento de recta o de un círculo, accedendo únicamente los que se tienen disponibles en la red que conforma el espacio destinado al trazado de figuras. Esto lo realizan partiendo de las coordenadas de los puntos extremos del segmento de recta, o de las del centro y las de un punto inicial del círculo, de donde en forma recursiva e interpolando, calculan el resto de la información.

X 1
Y 1
CONTROL 1
X 2
Y 2
CONTROL 2

FIGURA 2.1.

Sección base de tabla de datos.

Para la adquisición de los datos iniciales, se hace uso de la interfaz máquina-usuario, que es tema de estudio del siguiente capítulo, por medio de la cual se genera una tabla o archivo en donde se almacenan los parámetros requeridos por los algoritmos. La figura 2.1 muestra una sección de ésta, la cual es repetitiva a lo largo de toda su extensión, y en

donde se observa la disposición que deben de guardar los datos. Así, las coordenadas X1, Y1 corresponden al extremo inicial de una recta o al punto inicial de un círculo, mientras que X2, Y2 pertenecen al extremo final de una recta o al centro del círculo, dependiendo del trazo deseado.

La palabra control define precisamente el tipo de elemento a graficarse, además de que controla la posición de la pluma (arriba o abajo) y determina el fin de archivo. La figura 2.2 muestra los códigos de control válidos que pueden ser utilizados en la generación de la tabla.

CONTROL	ACCION REALIZADA
00	Traza recta con pluma abajo
01	Traza recta con pluma arriba
02	Traza círculo con pluma abajo
FF	Fin de archivo

FIGURA 2.2.

Códigos válidos palabra de control.

Existe dentro de este grupo un tercer programa, llamado GRAF12, que es el encargado de acceder directamente al archivo generado, con lo que, en base a los datos en él almacenados, proporciona los parámetros requeridos por los algoritmos de graficado para su funcionamiento.

A continuación, se hace un análisis detallado de los



tres programas que conforman este grupo, exponiendo su funcionamiento así como las subrutinas desarrolladas para su implementación.

### 2.2.1 Algoritmo de Bresenham para segmentos de recta.

Este algoritmo forma parte de los que se han dado a llamar como "incrementales". El desarrollo de éstos nace del hecho de contar como área de graficado, con una red de puntos uniformemente distribuidos, como la mostrada en la figura 1.2(a).

Para su implementación, este tipo de algoritmos toman el área como un plano cartesiano en donde a cada uno de los puntos le corresponde un par de coordenadas  $x$ ,  $y$ . Su nombre proviene del hecho de que la metodología que siguen es incrementar una de las coordenadas un determinado intervalo, e ir encontrando, en base a la ecuación que rige a la gráfica que se desea representar, el valor de la otra. Presentan además la ventaja de que generalmente son recursivos, esto es, que a partir de una cantidad mínima de datos de entrada, generan toda la información restante requerida para la construcción del trazo.

Existen varias condiciones que deben cumplir estos algoritmos: las líneas deben de tener la menor cantidad de identaciones posibles, deben de terminar precisamente (puntos extremos), deben de tener una densidad de puntos constante,

ésta debe ser independiente de la longitud y ángulo de la pendiente de la recta, y deben de ser veloces.

El algoritmo que se estudia en esta sección, además de poseer estas características, presenta otras, las cuales fueron determinantes para su elección como elemento del sistema graficador. Estas son:

- Para la generación de puntos, las operaciones que se requieren son únicamente sumas, restas y corrimientos.
- Estas operaciones son llevadas a cabo utilizando solamente aritmética entera.

Con esto, la velocidad de cálculo que presenta es considerablemente más alta que otros, además de que su implementación en lenguaje ensamblador ( y en general cualquier lenguaje computacional) es muy sencilla.

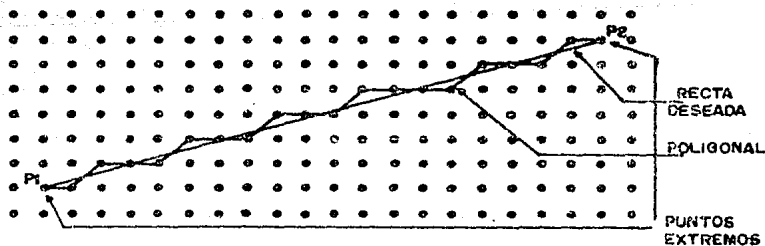


FIGURA 2.3.

Poligonal cercana a una recta deseada.

La finalidad del algoritmo es entonces, la de generar

una poligonal lo más cercana posible a una recta deseada accedendo unicamente a los puntos disponibles en la red de graficado. Esto se muestra en al figura 2.3.

Así, el algoritmo debe de ser capaz de seleccionar los puntos más cercanos a la recta cada vez que alguna de las coordenadas se incremente.

El algoritmo de Bresenham, para su funcionamiento incrementa la coordenada X, y en base a una variable de error encuentra la coordenada Y más cercana a la real. El intervalo en que se incrementa X es la unidad, y la variable de error depende de la distancia, sobre el eje Y, de los dos puntos más cercanos a la recta real para un determinado valor de X.

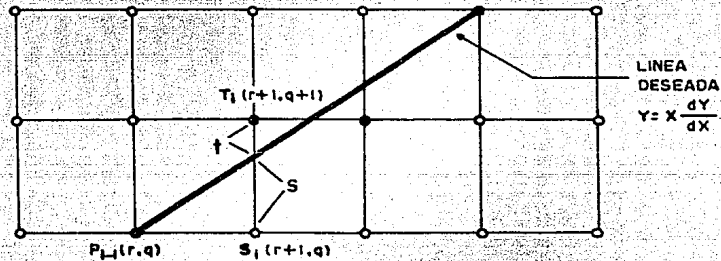


FIGURA 2.4.

Etapa 1-ésima de la generación de poligonal.

La figura 2.4 muestra la 1-ésima etapa de la generación de la poligonal por medio de este algoritmo, donde  $P_{i-1}$  ha sido elegido como el punto más cercano a la recta real en

la etapa 1-1, y T1 y S1 son las dos alternativas disponibles en esta etapa. Las distancias s y t sobre el eje Y, son en las que se basa la construcción de la variable de error citada anteriormente.

El algoritmo toma como variable de decisión la diferencia entre estas distancias, por lo que

si  $s < t$  se elige S1  
y si  $s > t$  se elige T1.

Con esto, en realidad lo que evalúa el algoritmo es la expresión s-t

si  $s-t < 0$  se elige S1  
y si  $s-t > 0$  se elige T1.

A continuación se derivan las expresiones que utiliza el algoritmo para su funcionamiento teniendo como condición que el ángulo que forma la recta con respecto al eje X, esté en el rango de 0 a 45 grados. Posteriormente se deducen las utilizadas generalizando el uso de éste en todo el plano.

Ya que la recta se traza desde  $(X1, Y1)$  a  $(X2, Y2)$ , si se traslada de manera que su punto inicial coincida con el origen, o muy cercano a él, entonces las coordenadas de los puntos extremos serán ahora  $(0, 0)$  y  $(dX, dY)$ , donde

$$dX = X2 - X1$$

$$\text{y } dY = Y2 - Y1$$

por lo que la ecuación de la recta será

$$Y = (dY/dX) X \dots (2.1)$$

La figura 2.5 muestra la disposición de la recta trasladada y la de los puntos  $P_{i-1}$ ,  $T_i$  y  $S_i$  en el plano.

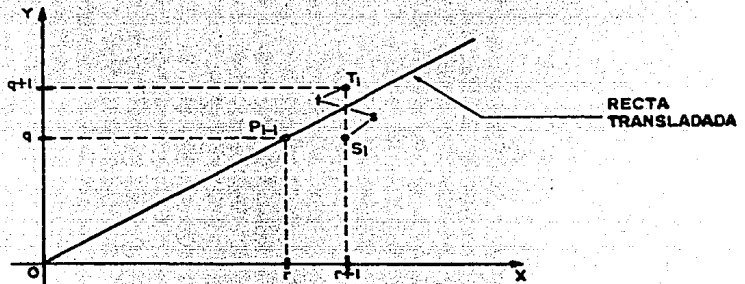


FIGURA 2.5.

Recta trasladada al origen.

Con esta nueva disposición, entonces las distancias  $s$  y  $t$  serán

$$s = (dY/dX)(r+1) - q \dots (2.2)$$

$$y \quad t = q+1 - (dY/dX)(r+1) \dots (2.3)$$

por lo que la variable de decisión tendrá la siguiente estructura

$$s-t = (dY/dX)(r+1) - q - q-1 + (dY/dX)(r+1)$$

$$s-t = 2(dY/dX)(r+1) - 2q-1 \dots (2.4)$$

donde si  $s-t < 0$  se elige  $S_i$  y en caso contrario se elige  $T_i$ .

Si multiplicamos los dos términos de (2.4) por  $dX$ , entonces

$$\begin{aligned} dX(s-t) &= 2dY(r+1) - 2qdX - dX \\ &= 2rdY + 2dY - 2qdX - dX \\ &= 2(rdY - qdX) + 2dY - dX \dots (2.5) \end{aligned}$$

pero dado que se está cumpliendo con la condición de que la recta tiene pendientes entre 0 y 1, entonces  $dX$  siempre es positivo, por lo que la variable de decisión se puede sustituir de la siguiente forma

$$d_i = dX(s-t)$$

donde si  $d_i < 0$  se elige  $S_i$ .

Ya que  $r$  y  $q$  son las coordenadas del punto  $P_{i-1}$ , entonces en (2.5) se puede sustituir también  $r$  por  $X_{i-1}$  y  $q$  por  $Y_{i-1}$ , con lo que

$$d_i = 2X_{i-1}dY - 2Y_{i-1}dX + 2dY - dX \dots (2.6)$$

y adicionando 1 a los subíndices

$$d_{i+1} = 2X_idY - 2Y_idX + 2dY - dX \dots (2.7)$$

que es el discriminante para la etapa  $i+1$  de la construcción de la recta.

Para que el método sea recursivo, entonces  $d_{i+1}$  debe estar en función de su valor anterior, por lo que si se resta (2.6) a (2.7), entonces se tendrá que

$$\begin{aligned}
 d_{i+1} - d_i &= 2X_i dY - 2Y_i dX + 2dY - dX - 2(X_{i-1})dY + 2(Y_{i-1})dX - 2dY + dX \\
 &= 2X_i dY - 2Y_i dX - 2(X_{i-1})dY + 2(Y_{i-1})dX \\
 &= 2(X_i - (X_{i-1}))dY - 2(Y_i - (Y_{i-1}))dX
 \end{aligned}$$

pero dado que el intervalo en que se incrementa la coordenada X, para encontrar

la Y, es la unidad, entonces  $X_i - (X_{i-1}) = 1$ , por lo que con esta consideración la expresión anterior toma la siguiente estructura

$$d_{i+1} - d_i = 2dY - 2dX(Y_i - (Y_{i-1}))$$

y sumando  $d_i$  en ambos términos se obtiene

$$d_{i+1} = d_i + 2dY - 2dX(Y_i - (Y_{i-1}))$$

que es la expresión general para encontrar el valor del nuevo discriminante en función del valor previo.

Para la elección del punto más cercano a la recta real, se tiene que si  $d_i > 0$  se elige  $T_1$ , por lo que  $Y_i = (Y_{i-1}) + 1$ , y entonces

$$d_{i+1} = d_i + 2dY - 2dX((Y_i - 1) + 1 - (Y_{i-1}))$$

$$d_{i+1} = d_i + 2dY - 2dX \dots (2.8)$$

De otra forma, si  $d_i < 0$  entonces  $S_1$  es elegido, con lo que  $Y_i = Y_{i-1}$  y sustituyendo en la expresión general

$$d_{i+1} = d_i - 2dX((Y_i - 1) - (Y_{i-1}))$$

$$d_{i+1} = d_i + 2dY \dots (2.9)$$

Las expresiones (2.8) y (2.9) son las que conforman la estructura recursiva del algoritmo para una determinada etapa intermedia en la construcción de un segmento de recta. El valor inicial de  $d_i$  se encuentra evaluando la ecuación (2.6) para  $i=1$ , donde sabemos que  $(X_0, Y_0) = (0, 0)$ , por lo que

$$d_1 = 2X_0dY - 2Y_0dX + 2dY - dX$$

$$d_1 = 2dY - dX \dots (2.10)$$

con lo que el algoritmo es completado.

Si se toma en cuenta que los movimientos que se realizan para acceder los puntos de la red de graficado son unicamente horizontales y diagonales, como se muestra en la figura 2.6, y estos son etiquetados como  $P_p$  y  $P_s$  respectivamente, entonces la estructura del algoritmo será:

1. Evaluar  $dX = X_2 - X_1$

$$dY = Y_2 - Y_1$$

2. Encontrar discriminante inicial

$$D = 2dY - dX$$

3. Repetir hasta  $dX = 0$

Si  $D > 0$ : Elegir  $T_1$

( movimiento  $P_s$  )

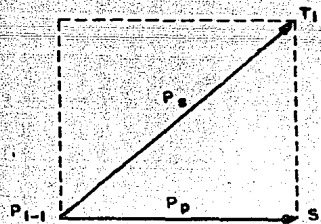


FIGURA 2.6

Movimientos en trazado de rectas.



Reevaluar

$D \leftarrow D + 2dY - 2dX$

$dX \leftarrow dX - 1$

$dY \leftarrow dY - 1$

De otra forma, si  $D \leq 0$ : Elegir Si

(movimiento Pp)

Reevaluar

$D \leftarrow D + 2dY$

$dX \leftarrow dX - 1$

$dY \leftarrow dY$

La figura 2.7 muestra un ejemplo numérico para este algoritmo con la condición  $dX \leq dY$ .

dX	dY	D	mov
9	3	-3	
8	3	-3	Pp
7	2	-7	Ps
6	2	-3	Pp
5	2	1	Pp
4	1	-5	Ps
3	1	-3	Pp
2	1	-1	Pp
1	1	1	Pp
0	0	-1	Ps

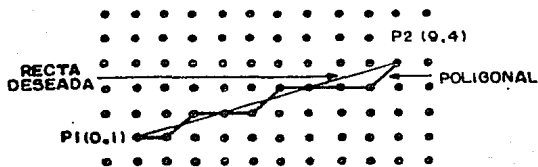


FIGURA 2.7.

Algoritmo de Bresenham para condición  $dX < dY$ .

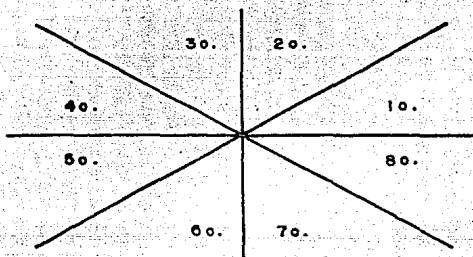


FIGURA 2.8.

Plano cartesiano dividido en octantes.

Para la generalización del algoritmo, se divide el plano en 8 partes iguales, los cuales se denotan como octantes. Graficamente se muestran en la figura 2.8.

Cada una de estas secciones está delimitada de acuerdo a

la relación que guardan las magnitudes de las coordenadas de un determinado punto situado en alguna de ellas. Así, los puntos situados en el 3er. octante, por ejemplo, cumplen con la condición de que el valor absoluto de la coordenada Y es mayor o igual que el absoluto de  $-X$ . La figura 2.9 muestra la tabla en donde se establecen las fronteras para cada uno de los octantes.

octante	fronteras	
1	$ X  \geq  Y $	$ Y  \geq 0$
2	$ Y  \geq  X $	$ X  \geq 0$
3	$ Y  \geq  -X $	$ X  \geq 0$
4	$ -X  \geq  Y $	$ Y  \geq 0$
5	$ -X  \geq  -Y $	$ Y  \geq 0$
6	$ -Y  \geq  -X $	$ X  \geq 0$
7	$ -Y  \geq  X $	$ X  \geq 0$
8	$ X  \geq  -Y $	$ Y  \geq 0$

FIGURA 2.9.

## Fronteras de octantes.

En la tabla anterior se puede observar que, tomando en cuenta esta división, el algoritmo de Bresenham para segmentos de recta es válido únicamente en el primer octante, ya que es solo en éste donde se cumple la condición bajo la cual se desarrolló ( $dX \geq dY$ ).

Para que la implementación del algoritmo sea posible en todos los octantes, es necesario entonces, hacer que las

condiciones que se cumplen en el primero de los octantes se cumplan también en los demás. Para lograr esto, se define a X como la variable a la cual se le asigna el valor de la coordenada mayor, mientras que a Y se le asigna la de menor magnitud.

octante	sustituciones	
1	$X =  X $	$Y =  Y $
2	$X =  Y $	$Y =  X $
3	$X =  Y $	$Y =  -X $
4	$X =  -X $	$Y =  Y $
5	$X =  -X $	$Y =  -Y $
6	$X =  -Y $	$Y =  -X $
7	$X =  -Y $	$Y =  X $
8	$X =  X $	$Y =  -Y $

FIGURA 2.10.

Sustituciones para generalización de algoritmo de rectas.

Con esto, y tomando como base la tabla de la figura 2.9, se realizan las sustituciones necesarias para que en cada uno de los octantes se cumpla la condición  $X \Rightarrow Y$ , con lo que el algoritmo es aplicable, cuando menos en este sentido, en todo el plano. La figura 2.10 muestra esta serie de sustituciones.

Ya que con las sustituciones anteriores se cumple con la condición básica para la generalización del algoritmo, es necesario atender también otro tipo de problema que se

presenta para alcanzar este fin, y es el relacionado al tipo de movimientos.

La figura 2.11 muestra el plano dividido en octantes, y en cada uno de estos los movimientos que se requieren para la construcción de un segmento de recta en alguno de ellos.

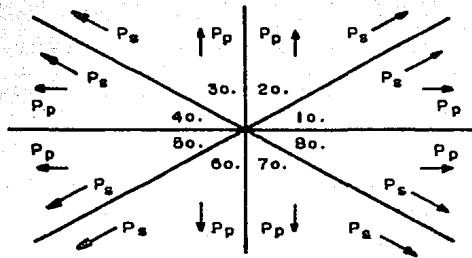


FIGURA 2.11.

Movimientos en 8 octantes para construcción de rectas.

Con esto, el algoritmo debe ser capaz de generar las acciones necesarias (modificar el sentido de los movimientos Pp y Ps) para el trazo de una línea dependiendo del octante en que se encuentre situada. La figura 2.12 muestra estas acciones para cada uno de los 8 diferentes octantes, en donde se puede observar dos cuestiones fundamentales que deben ser tomadas en cuenta y con las cuales se determinan completamente los requerimientos en cada uno de ellos.

Estas son:

- 1) Determinar la dirección del movimiento Pp.

a. Horizontal.

b. Vertical.

2) Determinar las modificaciones que deben sufrir las coordenadas X, Y para producir el movimiento Ps.

a. Incremento en X y Y.

b. Decremento en X e incremento en Y.

c. Decremento en X y Y.

d. Incremento en X y decremento en Y.

octante	Pp	Ps	octante	Pp	Ps
1	X=X+1 Y=Y	X=X+1 Y=Y+1	5	X=X-1 Y=Y	X=X-1 Y=Y-1
2	X=X Y=Y+1	X=X+1 Y=Y+1	6	X=X Y=Y-1	X=X-1 Y=Y-1
3	X=X Y=Y+1	X=X-1 Y=Y+1	7	X=X Y=Y-1	X=X+1 Y=Y-1
4	X=X-1 Y=Y	X=X-1 Y=Y+1	8	X=X+1 Y=Y	X=X+1 Y=Y-1

FIGURA 2.12.

Acciones necesarias para movimientos en octantes.

El sentido del movimiento Pp se reconoce facilmente ya que el sentido de Ps lo engloba, como se puede observar en la tabla 2.12.

Para que el algoritmo logre generar estos movimientos correctamente, se construye una palabra de control, la cual es tomada como referencia para el trazado de la gráfica. Esta consta de tres bits, ya que con este número es suficiente para definir las características del movimiento, y cada uno de ellos contiene información específica, la cual en conjunto, produce la necesaria para que el algoritmo reconozca el octante en que está situado.

La estructura de esta palabra es tal que dos bits se utilizan para definir el sentido del movimiento  $P_s$ , y con esto el de  $P_p$ , y el tercero para la dirección de este último. Así, dependiendo del estado de cada uno de estos, el algoritmo realiza las siguientes funciones:

bit0 = 0 ---  $P_p$  vertical.

bit0 = 1 ---  $P_p$  horizontal.

bit1 = 0 --- decremento en X.

bit1 = 1 --- incremento en X.

bit2 = 0 --- decremento en Y.

bit2 = 1 --- incremento en Y.

Para la generación de esta palabra de control, el algoritmo debe contemplar primero tres casos especiales:

\* Movimiento cero. Este se presenta cuando las diferencias en cada uno de los ejes es cero, esto es, cuando se trata de un

mismo punto. En este caso, no se realiza ninguna otra función mas que la de acceder la siguiente pareja de puntos de la tabla de datos.

\* Movimiento puramente horizontal. Este es cuando la diferencia en el eje Y es igual a cero, pero en el eje X no. Esta condición solo se presenta en el 4o. y 8o. octantes, y solo se reconoce si  $X > 0$  o no para generar la palabra de control, la cual será X01 y X11 respectivamente para cada uno de los casos (X = no importa).

\* Movimiento puramente vertical. Es inverso al anterior. Se tiene que la diferencia en X es cero, pero en Y no. Esto se presenta solamente en el 2o. y 6o. octantes, donde la palabra de control es 1X0 y 0X0 respectivamente, dependiendo si Y es mayor o menor que cero.

Si ninguno de estos casos se presenta, entonces la metodología que sigue es:

a. Reconoce si la diferencia en X es positiva o negativa, con lo que elimina la mitad del plano y sitúa el segmento de recta a generarse en la región formada por el primer y cuarto cuadrantes o en la que constituyen el segundo y el tercero. Con esto, si  $X > 0$  entonces el bit1 es 1, ya que en esta región se requiere siempre de un incremento en X para  $P_s$  (ver fig. 2.12, octantes 1, 2, 7 y 8), mientras que si  $X < 0$  entonces el valor del bit1 es cero (octantes 3, 4, 5 y 6).

b. Define en que cuadrante se encuentra la gráfica a trazarse



considerando si la diferencia en Y es mayor o menor que cero, con lo que elimina uno de los dos cuadrantes de la región en que situó en el paso anterior. Si  $Y > 0$  entonces el segmento de recta se encuentra en el primer o segundo cuadrante, con lo que el bit2 es 1, ya que los octantes que forman a cualquiera de estos dos (1,2,3 y 4), requieren siempre de un incremento en Y. Inversamente, si  $Y < 0$ , entonces el bit2 es cero (para octantes 5,6,7 y 8).

c. Finalmente define octante eliminando alguno de los dos del cuadrante seleccionado. Esto lo realiza evaluando la relación que existe entre las diferencias de ambos ejes, con lo que determina el tercer bit de la palabra de control. Si la diferencia en el eje X es mayor que la del Y, entonces la dirección de Pp es horizontal (octantes 1,4,5 u 8), con lo que el bit0 es 1. En caso contrario Pp es vertical (octantes 2,3,6 o 7), por lo que el bit0 es cero. En esta etapa, el algoritmo debe de considerar que las diferencias en los dos ejes pueden ser iguales, lo cual se presenta en los octantes 1,3,5 y 7 de acuerdo a la tabla de la figura 2.9. Para este caso, se puede observar que el bit que define el sentido del movimiento principal en esos octantes, está dado por la operación lógica NOR Exclusiva entre el bit1 y el 2 de la palabra de control, los cuales son previamente definidos.

En base al desarrollo anterior, se puede estructurar entonces una tabla con los diferentes valores que puede tomar la palabra de control necesarios para cada uno de los

octantes, y a partir de los cuales el algoritmo genera los movimientos requeridos en cada uno de ellos. Esta tabla se muestra en la figura 2.13.

octante	palabra de control	byte de control
1	111	07
2	110	06
3	100	04
4	101	05
5	001	01
6	000	00
7	010	02
8	011	03

FIGURA 2.13.

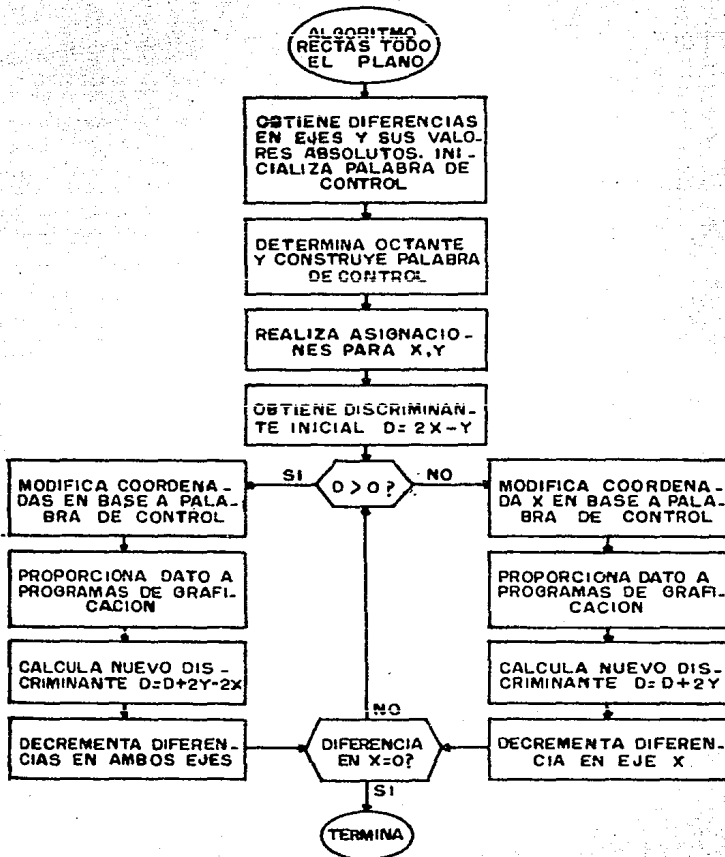
Palabra de control para los 8 octantes.

Con la construcción de esta palabra, el algoritmo de Bresenham para segmentos de recta es generalizado de tal forma que se hace posible su aplicación en todo el plano.

La siguiente ilustración muestra en forma de diagrama de flujo la totalidad del procedimiento que sigue el método para la generación de un segmento de línea en cualquier región del plano.

Para la implementación del algoritmo como parte del sistema graficador, se desarrollaron las subrutinas necesarias para realizar cada una de las acciones requeridas en el proceso de construcción. Estas fueron realizadas en su totalidad en lenguaje ensamblador para el microprocesador

Z-80, y para un estudio detallado, en el apéndice A de este trabajo se muestra el listado total de ellas.



A continuación se presentan las características principales de cada una de estas subrutinas, haciendo notar, que la función específica, así como la dirección de las

localidades de servicio utilizadas se expone en el apéndice B.

\* NOMBRE: DIF1.

FUNCION: Direcciona las coordenadas Y's de los puntos extremos y calcula la diferencia sobre este eje. Direcciona las coordenadas X's de los mismos puntos y calcula la diferencia en este eje también. Obtiene el valor absoluto de estos resultados e inicializa el valor de la palabra de control con OOH.

DIRECCION: 158EH.

SUBROUTINAS QUE UTILIZA: ABS.

DATOS ENTRADA: Registro IY apuntando a coordenada XI.

DATOS SALIDA: Diferencia eje X en localidad DIFX. Diferencia eje Y en localidad DIFY. Valor absoluto localidad DIFX en ABSX. Valor absoluto localidad DIFY en ABSY. Acumulador cargado con OOH.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: ABS.

FUNCION: Recibe un dato signado de 16 bits y obtiene su valor absoluto.

DIRECCION: 1617H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato signado en registro HL.

DATOS SALIDA: Valor absoluto en registro HL.

REGISTROS QUE MODIFICA: DE, HL.

\* NOMBRE: DETO.

FUNCION: Determina el octante en que se encuentra situado al segmento de recta a generarse. Con esto, construye la palabra de control requerida para identificar los tipos de movimientos a realizarse para la construcción de la gráfica.

DIRECCION: 15C4H.

SUBROUTINAS QUE UTILIZA: EVAY, EVAX, AXAY.

DATOS ENTRADA: Diferencia eje X en localidad DIFX.

Diferencia eje Y en localidad DIFY.

DATOS SALIDA: Palabra de control en localidad OCT.

REGISTROS QUE MODIFICA: A, HL, DE.

\* NOMBRE: EVAY.

FUNCION: Determina si la diferencia entre las coordenadas Y de los puntos extremos es positiva, negativa o cero.

DIRECCION: 1636H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Diferencia eje Y en localidad DIFY.

DATOS SALIDA: Resultado en registro de banderas.

REGISTROS QUE MODIFICA: HL.

\* NOMBRE: EVAX.

FUNCION: Determina si la diferencia entre las coordenadas X de los puntos extremos es positiva, negativa o cero.

DIRECCION: 162FH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Diferencia eje X en localidad DIFX.

DATOS SALIDA: Resultado en registro de banderas.

REGISTROS QUE MODIFICA: HL.

\* NOMBRE: AXAY.

FUNCION: Evalua la relación que existe entre los valores absolutos de las diferencias en ambos ejes. Esto es, cual de los dos es mayor o si son iguales.

DIRECCION: 1624H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Valor absoluto de la diferencia en eje X en localidad ABSX. Valor absoluto de la diferencia en eje Y en localidad ABSY.

DATOS SALIDA: Resultado en registro de banderas.

REGISTROS QUE MODIFICA: DE, HL.

\* NOMBRE: ASIG.

FUNCION: En base a la palabra de control generada por DETO realiza las asignaciones necesarias para que se cumpla la condición  $X \rightarrow Y$ .

DIRECCION: 152EH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Palabra de control en localidad OCT. Valor absoluto diferencia eje X en localidad ABSX. Valor absoluto diferencia eje Y en localidad ABSY.

DATOS SALIDA: Valor absoluto de mayor magnitud en localidad BRX. Valor absoluto de menor magnitud en localidad BRY.

REGISTROS QUE MODIFICA: A, HL.

\* NOMBRE: BRELI2.

FUNCION: Une dos puntos por medio de una poligonal lo más cercana posible a la recta delimitada por éstos. Es la implementación del algoritmo de Bresenham para segmentos de recta.

DIRECCION: 17C9H.

SUBROUTINAS QUE UTILIZA: DIS+, DIS-, TRANS.

DATOS ENTRADA: Valor absoluto de diferencia de mayor magnitud en localidad BRX. Valor absoluto de menor magnitud en localidad BRY. Coordenada X de punto inicial en localidad BUFX. Coordenada Y de punto inicial en localidad BUFY. Palabra de control en localidad OCT.

DATOS SALIDA: Coordenadas de puntos intermedios para generación de gráfica en localidades BUFX y BUFY (Ya que el método es recursivo, los datos almacenados en estas localidades cambiarán cada vez que se genere uno nuevo).

REGISTROS QUE MODIFICA: A, BC, DE, HL, IX.

\* NOMBRE: DIS+.

FUNCION: Modifica las coordenadas del punto a graficarse, en base a la palabra de control, para cuando se requiere realizar un movimiento del tipo Ps.

DIRECCION: 1839H.

SUBROUTINAS QUE UTILIZA: DECD, INCD.

DATOS ENTRADA: Palabra de control en localidad OCT.  
Coordenadas del último punto graficado en localidades BUFX y BUFY.

DATOS SALIDA: Coordenadas del punto próximo a graficarse en localidades BUFX y BUFY.

REGISTROS QUE MODIFICA: A, IX.

\* NOMBRE: DIS-.

FUNCION: Modifica en base a la palabra de control las coordenadas del punto a graficarse para cuando se requiere realizar un movimiento del tipo Pp.

DIRECCION: 185CH.

SUBROUTINAS QUE UTILIZA: DECD, INCD.

DATOS ENTRADA: Palabra de control en localidad OCT.  
Coordenadas del último punto graficado en localidades BUFX y BUFY.

DATOS SALIDA: Coordenadas del punto próximo a graficarse en localidades BUFX y BUFY.

REGISTROS QUE MODIFICA: A, IX.



\* NOMBRE: DECD.

FUNCION: Decrementa el dato almacenado en la localidad  
apuntada por IX.

DIRECCION: 1888H.

SUBROUTINAS QUE UTILIZA: RDATA, PPDA, PDATA.

DATOS ENTRADA: Registro IX apuntando a la localidad que  
contiene el dato a decrementarse.

DATOS SALIDA: Dato decrementado en localidad apuntada por  
IX.

REGISTROS QUE MODIFICA: DE, HL.

\* NOMBRE: INCD.

FUNCION: Incrementa el dato almacenado en la localidad  
apuntada por IX.

DIRECCION: 187BH.

SUBROUTINAS QUE UTILIZA: RDATA, PPDA, PDATA.

DATOS ENTRADA: Registro IX apuntando a la localidad que  
contiene el dato a incrementarse.

DATOS SALIDA: Dato incrementado en localidad apuntada por  
IX.

REGISTROS QUE MODIFICA: DE, HL.

\* NOMBRE: RDATA.

FUNCION: Toma la coordenada del último punto graficado y la  
transforma del formato requerido por la  
interfaz D/A (ver figura 1.6) al formato necesario

para su manipulación dentro del programa. Además, invierte cada uno de sus bits para obtener el valor real.

DIRECCION: 189DH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: IX apuntando a la coordenada próxima a modificarse.

DATOS SALIDA: Dato en formato requerido para su manipulación en registro DE.

REGISTROS QUE MODIFICA: A, DE.

\* NOMBRE: PPDA.

FUNCION: Almacena el dato próximo a transformarse para tenerlo en formato adecuado para su graficación, en localidad BUF, y apunta con el registro HL a ésta.

DIRECCION: 1895H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a formatearse en registro DE.

DATOS SALIDA: Dato a formatearse en localidad BUF y registro HL apuntando a

ésta.

REGISTROS QUE MODIFICA: HL.

\* NOMBRE: PDATA.

FUNCION: Formatea una coordenadas del punto próximo a graficarse para que su estructura sea compatible

con la interfaz D/A. Esta coordenada es la apuntada por el registro IX.

DIRECCION: 147CH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a formatearse en localidad apuntada por HL. Registro IX apuntando a la localidad en la que se almacena el dato formateado.

DATOS SALIDA: Dato formateado en localidad apuntada por IX.

REGISTROS QUE MODIFICA: A, HL.

### 2.2.2 Algoritmo de Bresenham para círculos.

Al igual que el algoritmo de rectas, éste pertenece al grupo denominado como incrementales. Con esto, su compatibilidad con el sistema es total, ya que su funcionamiento se apega a las características del área de graficado disponible.

El algoritmo tiene como objetivo, el de generar un círculo en base a una poligonal, la cual accesa únicamente a los puntos disponibles en la red, y trata de acercarse lo más posible al círculo verdadero. La figura 2.14 muestra este procedimiento.

Para la realización de esta tarea, el método cumple con las condiciones generales demandadas a un algoritmo incremental, pero adicionalmente las expresiones de las que

hace uso basan su construcción en sumas, restas y corrimientos, las cuales son evaluadas con aritmética entera, lo que determinó su inclusión dentro del sistema.

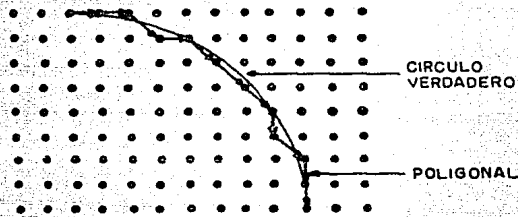


FIGURA 2.14.

Funcionamiento algoritmo para círculos.

Para su estudio, la metodología que se seguirá es semejante a la utilizada con el algoritmo de rectas. Primero se desarrollará para su aplicación en un octante, para posteriormente acondicionarlo en un cuadrante y finalmente generalizarlo en todo el plano. Esto se realiza para un círculo centrado en el origen inicialmente, y se presentará su implementación con centro situado en cualquier punto del plano en segundo término.

El algoritmo de Bresenham para círculos, al igual que todos los de su tipo, basa su funcionamiento en la ecuación que describe el comportamiento de la gráfica que se desea trazar. En este caso, la expresión básica del método es

$$X^{*2} + Y^{*2} = R^{*2} \dots (2.11)$$

que es la función de un círculo con centro en el origen, y donde  $X, Y$  son las coordenadas de un punto en el plano cartesiano y  $R$  es el radio del círculo.

En base a esta expresión, se establece que la distancia cuadrada del origen a cualquier punto perteneciente al círculo, está dada por la suma de los cuadrados de las coordenadas de éste, lo cual constituye la consideración inicial que hace el algoritmo para su desarrollo.

La figura 2.15 muestra una sección de la red de puntos accesibles en el área de graficado, donde  $P_{i-1}$  ha sido seleccionado como el punto más cercano al círculo en una etapa  $i-1$ . Las trayectorias etiquetadas con las letras A a la E, son las que puede tomar el comportamiento del círculo en el primer cuadrante. Si,  $T_i$  y  $R_i$  son las posibles elecciones para la etapa  $i$  en la determinación de la poligonal.

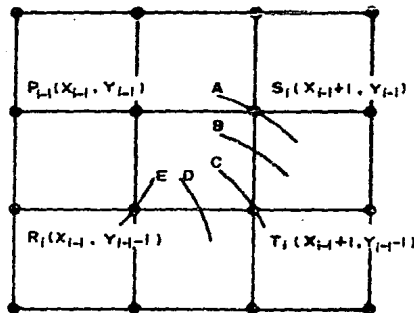


FIGURA 2.15.

Posibles trayectorias de la poligonal  
en primer cuadrante.

La derivación de las expresiones que son utilizadas por este algoritmo, se realizará primeramente para el segundo octante, esto es, para el intervalo  $0 < X < R/2$ , por lo que el estudio se centrará por ahora en los puntos  $S_1$  y  $T_1$ , y las trayectorias A, B y C.

Ya que la elección debe de realizarse entonces, entre los puntos  $S_1$  y  $T_1$ , se evalúa la distancia de cada uno de ellos hacia el origen, y comparandolas con la real, se selecciona la más cercana a ésta.

Para realizar esta comparación, el algoritmo define una variable de error fundamentada en la ecuación (2.11), la cual esta dada por

$$D(P_i) = (X^{*2} + Y^{*2}) - R^{*2} \dots (2.12)$$

la cual se aplica a las posibles selecciones, resultando elegida la que genere un resultado más cercano a cero (ya que para puntos pertenecientes al círculo, ese es el resultado).

Implementando la expresión (2.12) para los puntos  $S_1$  y  $T_1$ , se obtiene que

$$D(S_1) = [(X_1-1)+1]^{*2} + (Y_1-1)^{*2} - R^{*2} \quad (2.13)$$

$$D(T_1) = [(X_1-1)+1]^{*2} + ((Y_1-1)-1)^{*2} - R^{*2} \quad (2.14)$$

a partir de lo cual, si  $D(S_1) > D(T_1)$  se elige  $T_1$  como el punto más cercano al círculo verdadero. Inversamente, si

$D(S_1) < D(T_1)$ , entonces  $S_1$  es el punto elegido.

Como resultado de esto, se puede definir un discriminante para la selección del punto más cercano al círculo real, el cual tiene la siguiente estructura

$$d_1 = D(S_1) - D(T_1) \dots (2.15)$$

de donde

si  $d_1 > 0$  se elige  $T_1$

y si  $d_1 < 0$  se elige  $S_1$ .

Basandose en la figura 2.15, se observa que para la trayectoria B, se tiene que  $D(S_1)$  es mayor que cero, ya que el punto está situado fuera del círculo, mientras que  $D(T_1)$  es menor, ya que la posición de éste cae dentro del círculo. Tomando en cuenta estas condiciones, se puede establecer un nuevo discriminante

$$d_1 = D(S_1) + D(T_1) \dots (2.16)$$

donde

si  $d_1 > 0$  se elige  $T_1$

y si  $d_1 < 0$  se elige  $S_1$

el cual, aunque se generó a partir de la trayectoria B, es válido también para la A y la C, como a continuación se muestra.

Para el caso de A, se tiene que  $D(S_1)$  es igual a cero, ya que su posición coincide exactamente con la trayectoria

del círculo, mientras que el punto  $T_1$  sigue estando situado dentro de éste, por lo que  $D(T_1)$  es menor que cero. Si la comparación entre éstos se realiza por medio de (2.16), entonces se obtiene que  $d_1 < 0$ , por lo que con la elección que se lleva a cabo, el punto  $S_1$ , se cumple con el objetivo del algoritmo.

Para el caso de  $C$ , se tiene que  $D(S_1)$  es mayor que cero, mientras  $D(T_1)$  es cero, y sustituyendo esto en (2.16) se obtiene que  $d_1$  es mayor que cero, por lo que se elige  $T_1$ , cuestión que no se contrapone al propósito del método.

Con esto se concluye que (2.16) es aplicable para cualquier caso que se presente en el segundo octante.

La estructura detallada de esta expresión es entonces

$$d_1 = [(X_1-1)+1]^{**2} + (Y_1-1)^{**2} - R^{**2} \\ + [(X_1-1)+1]^{**2} + [(Y_1-1)-1]^{**2} - R^{**2} \\ d_1 = 2[(X_1-1)+1]^{**2} + [(Y_1-1)-1]^{**2} + (Y_1-1) - 2R^{**2}$$

$$d_1 = 2(X_1-1)^{**2} + 4(X_1-1) + 2 + (Y_1-1)^{**2} \\ - 2(Y_1-1) + 1 + (Y_1-1)^{**2} - 2R^{**2} \dots (2.17)$$

donde adicionando 1 a los subíndices se obtiene

$$d_{i+1} = 2(X_i)^{**2} + 4(X_i) + 2(Y_i)^{**2} \\ - 2(Y_i) - 2R^{**2} + 3 \dots (2.18)$$

que es el discriminante para la etapa  $i+1$  del desarrollo del algoritmo, el cual obtiene una estructura recursiva si se le resta (2.17), esto es



$$\begin{aligned}
 d_{i+1} - d_i &= 2(X_i)^2 + 4(X_i) + 2(Y_i)^2 - 2(Y_i) \\
 &\quad - 2R^2 + 3 - 2(X_{i-1})^2 - 4(X_{i-1}) \\
 &\quad - 2(Y_{i-1})^2 + 2(Y_{i-1}) + 2R^2 - 3
 \end{aligned}$$

y adicionando  $d_i$  en ambos términos

$$\begin{aligned}
 d_{i+1} &= d_i + 2[X_i^2 - (X_{i-1})^2] + 4[X_i - (X_{i-1})] \\
 &\quad + 2[Y_i^2 - (Y_{i-1})^2] - 2[Y_i - (Y_{i-1})] \dots (2.19)
 \end{aligned}$$

que es la expresión para obtener el discriminante en cualquier etapa del desarrollo en forma recursiva.

Para la elección del punto más cercano al círculo deseado tenemos que, si  $d_i > 0$  se elige  $T_i$ , por lo que  $X_i = (X_{i-1}) + 1$  y  $Y_i = (Y_{i-1}) - 1$ , y sustituyendo en (2.19)

$$\begin{aligned}
 d_{i+1} &= d_i + 2[(X_{i-1}) + 1]^2 - 2(X_{i-1})^2 + 4[(X_{i-1}) + 1] \\
 &\quad - 4(X_{i-1}) + 2[(Y_{i-1}) - 1]^2 - 2(Y_{i-1})^2 \\
 &\quad - 2[(Y_{i-1}) - 1] + 2(Y_{i-1})
 \end{aligned}$$

$$\begin{aligned}
 d_{i+1} &= d_i + 2(X_{i-1})^2 + 4(X_{i-1}) + 2 - 2(X_{i-1})^2 \\
 &\quad + 4(X_{i-1}) + 4 - 4(X_{i-1}) + 2(Y_{i-1})^2 - 4(Y_{i-1}) \\
 &\quad + 2 - 2(Y_{i-1})^2 - 2(Y_{i-1}) + 2 + 2(Y_{i-1})
 \end{aligned}$$

$$d_{i+1} = d_i + 4(X_{i-1}) - 4(Y_{i-1}) + 10$$

$$d_{i+1} = d_i + 4[(X_{i-1}) - (Y_{i-1})] + 10 \dots (2.20)$$

mientras que si  $d_i$  es menor que cero se elige  $S_i$ , por lo que  $X_i = (X_{i-1}) + 1$  y  $Y_i = (Y_{i-1})$ , lo que genera lo siguiente

$$\begin{aligned}
 d_{i+1} &= d_i + 2[(X_{i-1})+1]**2 - 2(X_{i-1})**2 + 4[(X_{i-1})+1] \\
 &\quad - 4(X_{i-1}) + 2(Y_{i-1})**2 - 2(Y_{i-1})**2 - 2(Y_{i-1}) + 2(Y_{i-1}) \\
 d_{i+1} &= d_i + 2(X_{i-1})**2 + 4(X_{i-1}) + 2 - 2(X_{i-1}) \\
 &\quad + 4(X_{i-1}) + 4 - 4(X_{i-1}) \\
 \\
 d_{i+1} &= d_i + 4(X_{i-1}) + 6 \dots (2.21)
 \end{aligned}$$

que son las ecuaciones en las que se basa el algoritmo para obtener la aproximación al círculo en forma recursiva. El valor inicial de la variable de error se obtiene haciendo la consideración de que el punto inicial para el trazo es el que tiene el par de coordenadas (0,R), las cuales, si se sustituyen en la ecuación (2.17) para  $i=1$ , producen el siguiente resultado:

$$\begin{aligned}
 d_1 &= 2(X_0)**2 + 4(X_0) + 2(Y_0)**2 - 2(Y_0) - 2R**2 + 3 \\
 &= 2(0)**2 + 4(0) + 2(R)**2 - 2(R) - 2R**2 + 3 \\
 \\
 d_1 &= 3 - 2R \dots (2.22)
 \end{aligned}$$

Con lo que el algoritmo es completado para su funcionamiento en el intervalo  $0 \leq X \leq R/2$  (segundo octante).

Si se toma en cuenta que los movimientos necesarios para el trazo del círculo son los mostrados en la figura 2.16, entonces la estructura total del algoritmo será:

1. Asignar:  $R = Y_0$

$$dY = Y_0$$

$$dX = 0$$

2. Calcular discriminante

inicial:

$$d = 3 - 2R$$

3. Repetir hasta  $dX \Rightarrow dY$

Si  $d > 0$ : Elegir  $T_1$   
(movimiento  $P_s$ ).

reevaluar

$$d \leftarrow d + 4(dX - dY) + 10$$

$$dX \leftarrow dX + 1$$

$$dY \leftarrow dY - 1$$

De otra forma, si  $d < 0$ : Elegir  $S_1$

(movimiento  $P_p$ ).

reevaluar

$$d \leftarrow d + 4(dX) + 6$$

$$dX \leftarrow dX + 1$$

$$dY \leftarrow dY$$

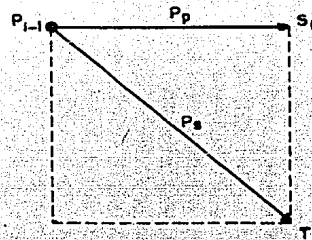


FIGURA 2.16.

Movimientos generación  
círculo segundo octante

La figura 2.17 muestra un ejemplo numérico del algoritmo para la condición de trabajo en el intervalo  $0 \leq X \leq R/2$ .

En base al desarrollo hasta esta etapa logrado, para la implementación del algoritmo en el primer cuadrante, se requiere solamente acondicionarlo para que después del trazo

en el segundo octante, sea capaz de realizar las elecciones necesarias para el trazo en el primer octante.

De acuerdo a la figura 2.15, para el primer octante las trayectorias que puede seguir el trazo del círculo, son las etiquetadas con las letras C, D y E, por lo que la elección, si se toma a  $P_{i-1}$  como el punto elegido en la etapa  $i-1$ , tiene que realizarse entre  $T_i$  y  $R_i$ , para la etapa  $i$ .

dX	dY	d	mov
0	10	-17	
1	10	-11	Pp
2	10	-1	Pp
3	10	13	Pp
4	9	-5	Ps
5	9	17	Pp
6	8	11	Ps
7	7	13	Ps

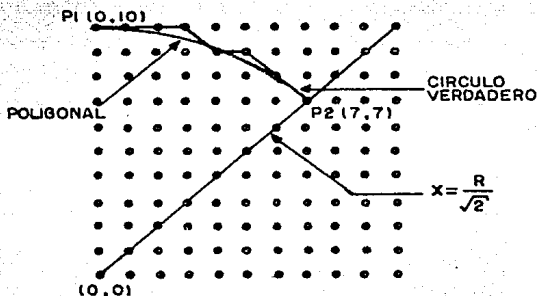


FIGURA 2.17.

Algoritmo de círculos segundo octante.

Así, aplicando la ecuación (2.12) a estos dos, se tiene que

$$D(T_1) = [(X_1-1)+1]^2 + [(Y_1-1)-1]^2 - R^2 \quad (2.23)$$

$$D(R_1) = [X_1-1]^2 + [(Y_1-1)-1]^2 - R^2 \quad \dots (2.24)$$

a partir de las cuales se puede demostrar, de manera semejante a la seguida en el segundo octante, que

$$d_i = D(T_1) + D(R_1) \quad \dots (2.25)$$

donde

si  $d_i > 0$  se elige  $R_i$   
y si  $d_i < 0$  se elige  $T_i$ .

Con esto, la estructura de la variable de error es

$$d_i = [(X_{i-1})+1]**2 + 2[(Y_{i-1})-1]**2 + (X_{i-1})**2 - 2R**2 \quad \dots (2.26)$$

y adicionando uno a los subindices, se obtiene que

$$d_{i+1} = [(X_i)+1]**2 + 2[(Y_i)-1]**2 + (X_i)**2 - 2R**2$$

la cual adquirirá estructura recursiva si se le resta (2.26)

$$d_{i+1} - d_i = [(X_i)+1]**2 + 2[(Y_i)-1]**2 + (X_i)**2 - [(X_{i-1})+1]**2 - 2[(Y_{i-1})-1]**2 - (X_{i-1})**2$$

de donde, adicionando  $d_i$  en ambos términos

$$d_{i+1} = d_i + [(X_i)+1]**2 + 2[(Y_i)-1]**2 + (X_i)**2 - [(X_{i-1})+1]**2 - 2[(Y_{i-1})-1]**2 - (X_{i-1})**2 \quad (2.28)$$

que es la expresión para calcular el discriminante en una etapa  $i+1$  en la generación de la poligonal en el primer octante, en función de su valor previo.

Para la elección, si  $d_i > 0$  se elige  $T_i$  y con esto  $X_i = (X_{i-1})+1$  y  $Y_i = (Y_{i-1})-1$ , por lo que (2.28) adquiere la siguiente estructura

$$d_{i+1} = d_i + [(X_{i-1})+2]**2 + 2[(Y_{i-1})-2]**2 + [(X_{i-1})+1]**2$$

$$\begin{aligned}
 & -[(X_1-1)+1]**2-2[(Y_1-1)-1]**2-(X_1-1)**2 \\
 d_{i+1} = & d_i+(X_1-1)**2+4(X_1-1)+4+2(Y_1-1)**2-8(Y_1-1)+8 \\
 & +(X_1-1)**2+2(X_1-1)+1-(X_1-1)**2-2(X_1-1)-1 \\
 & -2(Y_1-1)**2+4(Y_1-1)-2-(X_1-1)**2 \\
 d_{i+1} = & d_i + 4(X_1-1) + 10 - 4(Y_1-1)
 \end{aligned}$$

$$d_{i+1} = d_i + 4[(X_1-1)-(Y_1-1)] + 10 \quad \dots (2.29)$$

mientras que si  $d_i < 0$ , entonces se elige  $R_1$ , con lo que  $X_1=(X_1-1)$  y  $Y_1=(Y_1-1)-1$ , las cuales si se sustituyen en (2.28) se obtiene

$$\begin{aligned}
 d_{i+1} = & d_i+[(X_1-1)+1]**2+2[(Y_1-1)-2]**2+(X_1-1)**2 \\
 & -[(X_1-1)+1]**2-2[(Y_1-1)-1]**2-(X_1-1)**2 \\
 d_{i+1} = & d_i+2(Y_1-1)**2-8(Y_1-1)+8-2(Y_1-1)**2+4(Y_1-1)-2
 \end{aligned}$$

$$d_{i+1} = d_i - 4(Y_1-1) + 6 \quad \dots (2.30)$$

que son las expresiones para evaluar el discriminante en una etapa intermedia de la generación de la poligonal en el primer octante.

El discriminante inicial de esta etapa, se obtiene a partir de la ecuación (2.26), para la cual se hace la consideración de que el punto inicial de esta etapa es cuando  $X=Y$ , por lo que

$$\begin{aligned}
 d_1 = & (X+1)**2 + 2(Y-1)**2 + X**2 - 2R**2 \\
 d_1 = & X**2 + 2X + 1 + 2Y**2 - 4Y + 2 + X**2 - 2R**2 \\
 d_1 = & 2(X**2+Y**2) + 2X - 4Y + 3 - 2R**2
 \end{aligned}$$

pero  $X^2+Y^2=R^2$  y  $X=Y$ , lo que genera, si se sustituye en la expresión anterior, el siguiente resultado

$$d1 = 2R^2 + 2X - 4X + 3 - 2R^2$$

$$d1 = 3 - 2X \dots (2.31)$$

con lo que el algoritmo en el primer octante es completado.

Tomando en consideración los movimientos requeridos en la generación de la poligonal para el primer octante, mostrados en la figura 2.18, entonces el algoritmo total para el primer cuadrante será:

1. Asignar  $R=Y0$

$$dY = R$$

$$dX = 0$$

2. Calcular discriminante

inicial

$$d = 3 - 2R$$

3. Repetir hasta  $dX \rightarrow dY$

Si  $d > 0$ : Elegir  $T1$

(movimiento  $P_s$ )

reevaluar

$$d \leftarrow d + 4(dX - dY) + 10$$

$$dX \leftarrow dX + 1$$

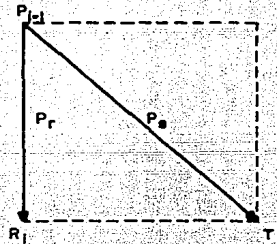


FIGURA 2.18.

Movimientos círculo  
primer octante.

$dY \leftarrow dY-1$

De otra forma, si  $d < 0$ : Elegir S1

(movimiento Pp)

reevaluar

$d \leftarrow d+4(dX)+6$

$dX \leftarrow dX+1$

$dY \leftarrow dY$

4. Calcular discriminante

primer octante

$d = 3 - 2(dX)$

5. Repetir hasta  $dY=0$

Si  $d > 0$ : Elegir R1

(movimiento Pr)

reevaluar

$d \leftarrow d - 4(dY) + 6$

$dX \leftarrow dX$

$dY \leftarrow dY-1$

De otra forma: Si  $d \leq 0$ : Elegir T1

(movimiento Ps)

reevaluar

$d \leftarrow d+4(dX-dY)+10$

$dX \leftarrow dX+1$

$dY \leftarrow dY-1$

La figura 2.19 muestra un ejemplo numérico para esta implementación, el cual es continuación del ejemplo de la figura 2.17.



dx	dy	d	mov
0	10	-17	
1	10	-11	Pp
2	10	-1	Pp
3	10	13	Pp
4	9	-5	Ps
5	9	17	Pp
6	8	11	Ps
7	7	13	Ps
7	7	-11	
8	6	-1	Ps
9	5	11	Ps
9	4	-3	Pr
10	3	27	Ps
10	2	21	Pr
10	1	19	Pr
10	0	17	Pr

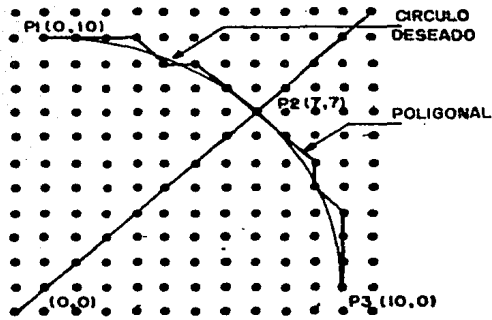


FIGURA 2.19.

Algoritmo círculos primer cuadrante.

Para la generalización del algoritmo, se parte del hecho de que si el trazo generado en el primer cuadrante se refleja en cada uno de los tres restantes, se obtiene como resultado final la gráfica completa del círculo.

Esto se logra tomando en cuenta que para la construcción de la poligonal en el trazo inicial, se requieren dos tipos de movimientos:

- El debido a la actualización de una sola coordenada. (Pp para el segundo octante y Pr para el primero).
- El producido por la actualización de las dos

coordenadas. (Ps en ambos casos).

los cuales, si se modifican apropiadamente tanto en sentido como en dirección, pueden ser aplicados en los tres cuadrantes restantes, con lo que se alcanza el objetivo planteado.

Si se denota al primero como movimiento simple ( $M_s$ ) y al segundo como conjugado ( $M_c$ ), entonces el sentido y dirección que deben tener para la construcción total de la gráfica es el mostrado en la figura 2.20, en la que se observa el comportamiento de éstos en cada uno de los octantes.

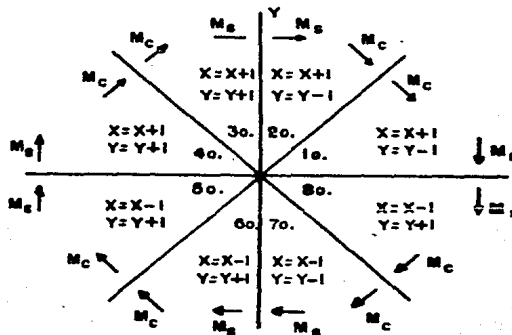


FIGURA 2.20.

Movimiento simple y conjugado en todo el plano.

De la figura anterior, si se toma en cuenta que el movimiento conjugado es la suma del simple más uno auxiliar ( $M_a$ ), como se muestra en la figura 2.21, se observa que el algoritmo debe determinar las siguientes características:

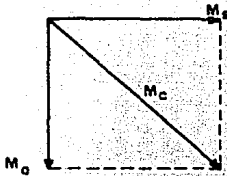


FIGURA 2.21.

Componentes  $M_c$ .

1. Si el movimiento simple se da en el eje X o en el Y.

2. Si el movimiento auxiliar se da en el eje X o Y.

3. Si el movimiento en el eje X requiere incremento o decremento.

4. Si el movimiento en el eje Y requiere incremento o decremento.

lo cual lo lleva a cabo por medio de una palabra de control que esta constituida por cuatro bits, uno para cada característica, los cuales establecen las siguientes condiciones:

bit0. Determina si el movimiento en el eje Y se lleva a cabo en base a un incremento o un decremento.

bit0=1 - incremento

bit0=0 - decremento

bit1. Determina si el movimiento en el eje X se lleva a cabo en base a un incremento o un decremento.

bit1=1 - incremento

bit1=0 - decremento

bit2. Determina si el movimiento simple se realiza sobre el eje X o el Y.

bit2=1 - eje X

bit2=0 - eje Y

bit3. Determina si el movimiento auxiliar se realiza sobre el eje X o el Y.

bit3=1 - eje X

bit3=0 - eje Y

En base a esto, y tomando en cuenta los incrementos y decrementos requeridos en cada octante (mostrados en la figura 2.18), además de las direcciones de Ms y Mc, la palabra de control es la mostrada en la tabla de la figura 2.22 para cada sección del plano.

octante	control	octante	control
1	1010 (0A)	5	1001 (09)
2	0110 (06)	6	0101 (05)
3	0111 (07)	7	0100 (04)
4	1011 (0B)	8	1000 (08)

FIGURA 2.22.

Palabra de control movimientos en todo el plano.

Con esto, el algoritmo debe inicializar ésta con 06H para realizar el trazo en el segundo octante, modificarla a 0AH para el primero y seguir reevaluandola para lograr la gráfica total con el siguiente orden: 8o., 7o., 6o.,

5o., 4o. y 3er. octantes.

Aunque con la implementación de esta palabra de control se cumple con los requerimientos para llevar a cabo las acciones necesarias del trazado de la gráfica total, existe otro problema que debe ser atacado, y es el referente a la actualización de las coordenadas.

Cuando se inicia el trazo de un círculo, las coordenadas del punto de arranque son  $(0, R)$ , mientras que las del punto final son  $(R/2, R/2)$ , para el segundo octante, y  $(R, 0)$  para el primero. Estos valores coinciden con las variables  $dX$  y  $dY$ , en las que se basa el algoritmo para su funcionamiento, pero se debe poner especial atención a su magnitud cuando se generaliza el algoritmo.

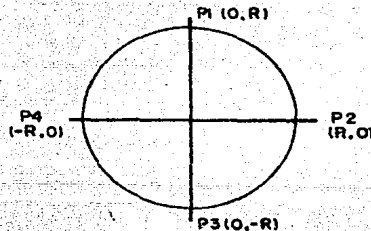


FIGURA 2.23.

Puntos extremos en trazo círculo.

La figura 2.23 muestra los puntos inicial y final en el trazo de cada cuadrante. Si se sigue el algoritmo básico, el valor de las variables  $dX$  y  $dY$  al final del primer cuadrante serán las coordenadas del punto  $P2$ . En esta etapa del trazo, es el momento en que se inicia la reflexión del

método para el cuarto cuadrante, por lo que se define la palabra de control adecuada y, en base a la estructura del método, se asigna  $dX=0$  y  $dY=R$ . Con esto, el trazo presenta el problema de regresar al punto P1 y adicionalmente realizar movimientos erróneos debidos a la nueva palabra de control.

Este problema se resuelve si se implementan dos variables adicionales, a las cuales se les asigna el valor real de las coordenadas del punto graficado, y las que no se ven afectadas (en su magnitud), cuando se realiza la reflexión en los cuadrantes 2o., 3o. y 4o.

Si se denota a estas variables como COX y COY, entonces el algoritmo debe inicializar las variables dX y COX con cero, así como a dY y COY con el valor del radio para el primer cuadrante. Con esto, cuando finaliza el trazo inicial, el valor de dX y COX es R, mientras que el de dY y COY es cero. En ese momento, se implementa el método para el cuarto cuadrante, definiendo la palabra de control con 08H y asignando  $dX=0$  y  $dY=R$ , pero manteniendo intacto el valor de COX y COY, con lo que, cuando se realizan las evaluaciones para este nuevo segmento de gráfica, la magnitud de éstas se ve modificada de tal forma que su valor al final del trazo en este cuadrante coincide con las coordenadas del punto P3 de la figura 2.23, en donde se realizan las mismas asignaciones que en la etapa anterior para el trazo del 3er. cuadrante, y de igual forma para el segundo.

Un entendimiento más fácil de la implementación de estas nuevas variables se logra observando el ejemplo numérico de

la figura 2.24, en donde se muestra la generalización del método para la construcción de la poligonal en todo el plano, con la condición de que el centro del círculo sea el origen.

cont	dX	COX	dY	COY	d	mov
06	0	0	10	10	-17	
06	1	1	10	10	-11	Ms
06	2	2	10	10	-1	Ms
06	3	3	10	10	13	Ms
06	4	4	9	9	-5	Mc
06	5	5	9	9	17	Ms
06	6	6	8	8	11	Mc
06	7	7	7	7	13	Mc
0A	7	7	7	7	-11	
0A	8	8	6	6	-1	Mc
0A	9	9	5	5	11	Mc
0A	9	9	4	4	-3	Ms
0A	10	10	3	3	27	Mc
0A	10	10	2	2	21	Ms
0A	10	10	1	1	19	Ms
0A	10	10	0	0	17	Ms
08	0	10	10	0	-17	
08	1	10	10	-1	-11	Ms
08	2	10	10	-2	-1	Ms
08	3	10	10	-3	13	Ms
08	4	9	9	-4	-5	Mc
08	5	9	9	-5	17	Ms
08	6	8	8	-6	11	Mc
08	7	7	7	-7	13	Mc
04	7	7	7	-7	-11	
04	8	6	6	-8	-1	Mc
04	9	5	5	-9	11	Mc
04	9	4	4	-9	-3	Ms
04	10	3	3	-10	27	Mc
04	10	2	2	-10	21	Ms
04	10	1	1	-10	19	Ms
04	10	0	0	-10	17	Ms
05	0	0	10	-10	-17	
05	1	-1	10	-10	-11	Ms
05	2	-2	10	-10	-1	Ms
05	3	-3	10	-10	13	Ms
05	4	-4	9	-9	-5	Mc

cont	dX	COX	dY	COY	d	mov
05	5	-5	9	-9	17	Ms
05	6	-6	8	-8	11	Mc
05	7	-7	7	-7	13	Mc
09	7	-7	7	-7	-11	
09	8	-8	6	-6	-1	Mc
09	9	-9	5	-5	11	Mc
09	9	-9	4	-4	-3	Ms
09	10	-10	3	-3	27	Mc
09	10	-10	2	-2	21	Ms
09	10	-10	1	-1	19	Ms
09	10	-10	0	0	17	Ms
0B	0	-10	10	0	-17	
0B	1	-10	10	1	-11	Ms
0B	2	-10	10	2	-1	Ms
0B	3	-10	10	3	13	Ms
0B	4	-9	9	4	-5	Mc
0B	5	-9	9	5	17	Ms
0B	6	-8	8	6	11	Mc
0B	7	-7	7	7	13	Mc
07	7	-7	7	7	-11	
07	8	-6	6	8	-1	Mc
07	9	-5	5	9	11	Mc
07	9	-4	4	9	-3	Ms
07	10	-3	3	10	27	Mc
07	10	-2	2	10	21	Ms
07	10	-1	1	10	19	Ms
07	10	0	0	10	17	Ms

FIGURA 2.24(a).

Generalización algoritmo de círculos.



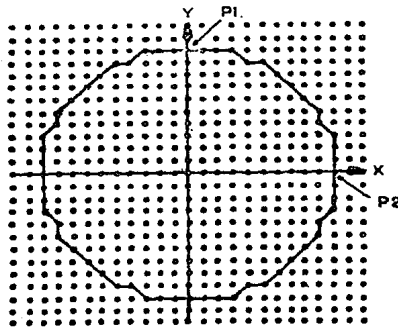


FIGURA 2.24(b).

Resultado gráfico tabla figura 2.24(b).

Dentro del sistema graficador, la implementación de estos dos diferentes tipos de variable no representa problema alguno, pues como se puede observar en la exposición de las características de las subrutinas construidas para el algoritmo de rectas, existen localidades de servicio independientes para el almacenamiento de éstas, BRX y BRY para  $dX$  y  $dY$  respectivamente, y BUFX, BUFY para las coordenadas del punto a graficarse, que en este caso son utilizadas para las variables COX y COY, por lo que el algoritmo puede ser desarrollado fácilmente.

Es también en base a esta estructura, como se logra la generación de un círculo con centro en un punto arbitrario, pues por medio de las diferentes localidades de servicio disponibles, se simula la construcción de la gráfica centrada en el origen, estableciéndose las condiciones requeridas para

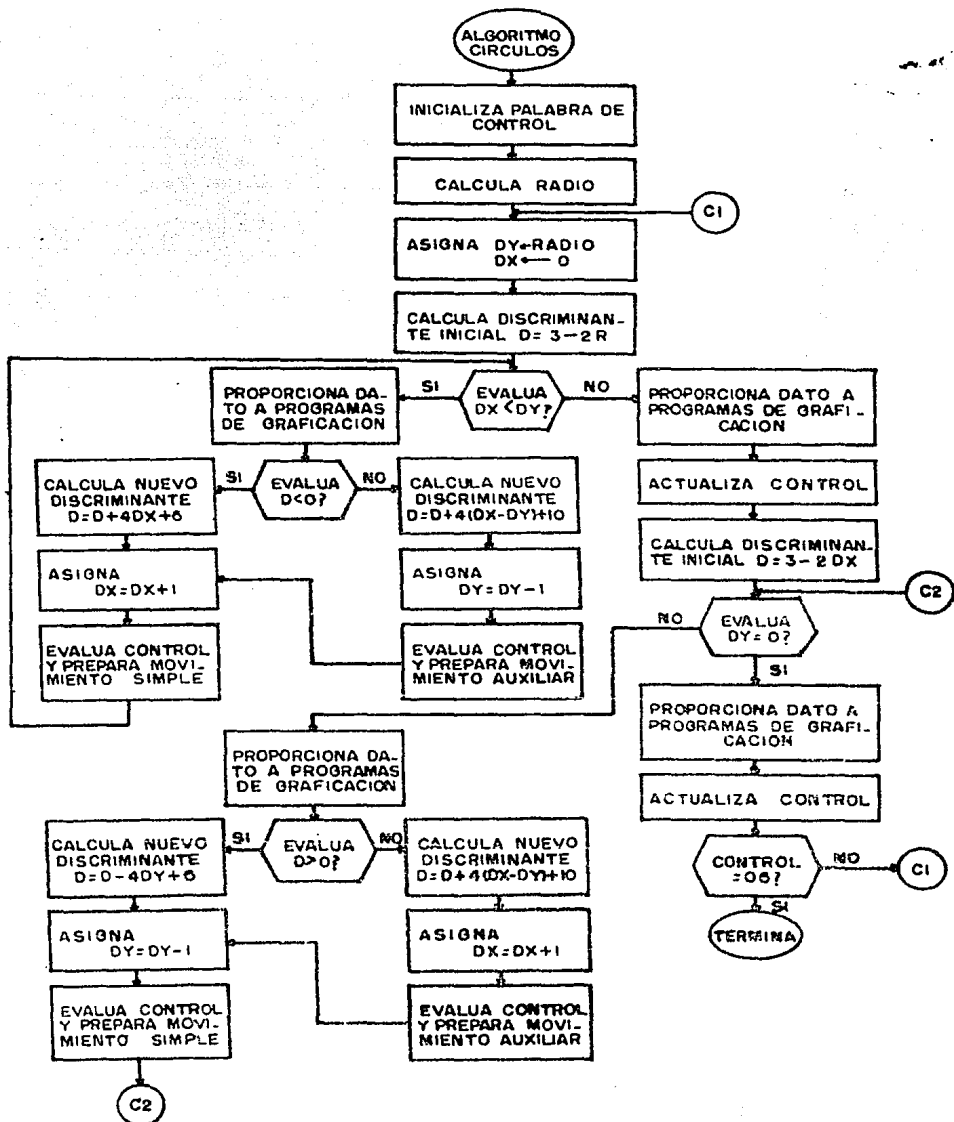
esta tarea, pero asignandole a las variables que contienen las coordenadas del punto a graficarse los valores necesarios para el trazo en cualquier otro lugar del plano.

Los datos requeridos por el algoritmo son entonces: el punto inicial del trazo y el centro del círculo. El suministro de éstos es de tal forma que las coordenadas X de ambos deben ser iguales, mientras la diferencia entre las Y el radio del círculo.

En la página siguiente se muestra el diagrama a bloques del algoritmo total.

Cabe hacer notar, que para el funcionamiento del algoritmo, se da por hecho que las coordenadas del punto inicial del trazo están almacenadas con anterioridad en las localidades destinadas para COX y COY, es decir BUFX y BUFY. Además, el proceso de actualización de la palabra de control, debe ser un ciclo cerrado, esto es, que después de definir la palabra para el tercer octante (último en el trazo), se reinicia el proceso con la requerida para el segundo (06H). Esto último se debe a que, como se observa en el diagrama anterior, al algoritmo toma como señal de salida esta palabra.

Es necesario aclarar también, que el algoritmo nunca trabaja con números negativos, debido a que el convertidor D/A en la interfaz hacia el graficador se implementó como unipolar, aunque el usuario puede contar con esta alternativa como se verá en el capítulo dedicado a la comunicación máquina-usuario.



A continuación se exponen las características principales de cada una de las subrutinas desarrolladas para la implementación del algoritmo, de las cuales puede ser consultado su listado en el apéndice A de este trabajo para un estudio profundo de ellas. La función específica de las localidades de servicio utilizadas, así como su dirección, se listan en el B.

\* NOMBRE: BREC12

FUNCION: Genera una poligonal lo más cercana posible a un círculo deseado dado, a partir de un punto inicial y el centro del mismo. Se basa fundamentalmente en el algoritmo de Bresenham para círculos.

DIRECCION: 163DH.

SUBROUTINAS QUE UTILIZA: X1MY1, SUMA4, Y-1, MOV2, X+1, MOV1, DEFCON, YICERO.

DATOS ENTRADA: Localidad coordinada X punto inicial apuntada por IY. Coordenadas de punto inicial almacenadas en BUFX y BUFY.

DATOS SALIDA: Coordenadas puntos intermedios para generación de gráfica, en localidades BUFX y BUFY. (Ya que el método es recursivo, los datos almacenados en estas localidades cambian cada vez que se genera uno nuevo).

REGISTROS QUE MODIFICA: A, DE, BC, HL, IX.

\* NOMBRE: X1MY1.

FUNCION: Establece la relación existente entre las variables  $dX$  y  $dY$  en base a la resta del valor almacenado en BRX y BRY.

DIRECCION: 1752H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Magnitudes de variables  $dX$  y  $dY$  en localidades BRX y BRY respectivamente.

DATOS SALIDA: Relación entre magnitudes en registro de banderas.

REGISTROS QUE MODIFICA: DE, HL.

\* NOMBRE: EVADIS.

FUNCION: Determina si la magnitud del discriminante es mayor, menor o igual a cero.

DIRECCION: 1764H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Magnitud del discriminante en localidad DIS1.

DATOS SALIDA: Condición que se cumple en registro de banderas.

REGISTROS QUE MODIFICA: BC, HL.

\* NOMBRE: SUMA4.

FUNCION: El contenido de HL lo multiplica por cuatro y el resultado lo adiciona a la magnitud del discriminante.

DIRECCION: 176BH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a multiplicarse en HL. Magnitud discriminante en DIS1.

DATOS SALIDA: Resultado de la suma entre discriminante y dato multiplicado en HL.

REGISTROS QUE MODIFICA: HL.

\* NOMBRE: Y-1.

FUNCION: Decrementa en uno la magnitud de la variable dY.

DIRECCION: 1773H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Magnitud variable dY en localidad BRY.

DATOS SALIDA: Magnitud decrementada en localidad BRY.

REGISTROS QUE MODIFICA: HL.

\* NOMBRE: MOV2.

FUNCION: Evalúa en base a la palabra de control, si el movimiento auxiliar se realiza en el eje X o en el Y. Después de esto, realiza incremento o decremento en la variable asignada al eje seleccionado.

DIRECCION: 1783H.

SUBROUTINAS QUE UTILIZA: BIT0, BIT1.

DATOS ENTRADA: Palabra de control en localidad CONT.  
Coordenadas de punto a graficarse en BUFx y BUFy.

DATOS SALIDA: Coordenadas del punto a graficarse

modificadas para realizar movimiento auxiliar en BUFX y BUFY.

REGISTROS QUE MODIFICA: A, DE, HL, IX.

\* NOMBRE: BIT0.

FUNCION: Evalua el bit 0 de la palabra de control para realizar un incremento o decremento en la variable asignada al eje Y.

DIRECCION: 17B1H.

SUBROUTINAS QUE UTILIZA: INCD, DECD.

DATOS ENTRADA: Palabra de control en localidad CONT. Coordinada Y de punto a graficarse en BUFY y apuntada por HL.

DATOS SALIDA: Coordinada Y del punto a graficarse modificada en localidad BUFY.

REGISTROS QUE MODIFICA: DE, HL.

\* NOMBRE: INCD (Explicada anteriormente).

DIRECCION: 187BH.

\* NOMBRE: DECD (Explicada anteriormente).

DIRECCION: 1888H.

\* NOMBRE: BIT1.

FUNCION: Evalua el bit 1 de la palabra de control para realizar un incremento o un decremento en la variable asignada al eje X.

DIRECCION: 17BDH.

SUBROUTINAS QUE UTILIZA: INCD, DECD.

DATOS ENTRADA: Palabra de control en localidad CONT.  
Coordenada X del punto a graficarse en  
localidad BUFX y apuntada por IX.

DATOS SALIDA: Coordenada X del punto a graficarse  
modificada en localidad BUFX.

REGISTROS QUE MODIFICA: DE, HL.

\* NOMBRE: X+1.

FUNCION: Incrementa en uno la variable dX almacenada en  
localidad BRX.

DIRECCION: 177BH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Magnitud de dX en localidad BRX.

DATOS SALIDA: Magnitud incrementada de dX en localidad BRX.

REGISTROS QUE MODIFICA: HL.

\* NOMBRE: MOV1.

FUNCION: Evalua en base a la palabra de control, si el  
movimiento simple se realiza sobre el eje X o el  
Y. Después de esto, realiza incremento o  
decremento en la variable asignada al eje  
seleccionado.

DIRECCION: 179AH.

SUBROUTINAS QUE UTILIZA: BITO, BIT1.

DATOS ENTRADA: Palabra de control en localidad CONT.



Coordenadas del punto a graficarse en localidades BUFX y BUFY.

DATOS SALIDA: Coordenadas del punto a graficarse modificadas para realizar movimiento simple, en BUFX y BUFY.

REGISTROS QUE MODIFICA: A, DE, HL, IX.

\* NOMBRE: DEFCON.

FUNCION: Actualiza la palabra de control dependiendo de la etapa en que se encuentra la construcción de la poligonal. Realiza el siguiente ciclo de valores, tomando en cuenta que la palabra se inicializa con 06H: 08H, 04H, 05H, 09H, 0BH, 07H y reinicia con 06H.

DIRECCION: 170AH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Palabra de control en localidad CONT.

DATOS SALIDA: Palabra de control actualizada en CONT.

REGISTROS QUE MODIFICA: A.

\* NOMBRE: YICERO.

FUNCION: Evalua si la variable dY es mayor, menor o igual que cero.

DIRECCION: 175DH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Magnitud de la variable dY en localidad BRY.

DATOS SALIDA: Resultado de la evaluación en registro de

banderas.

REGISTROS QUE MODIFICA: HL.

### 2.2.3 Programa generador de rectas y círculos.

La función principal de este programa es la de suministrar a los algoritmos de graficado los parámetros necesarios para su funcionamiento. Esto lo realiza a partir de la tabla de datos proporcionada por el usuario, de la cual, obtiene también información adicional para realizar las siguientes tareas: inicializar las localidades que contienen las coordenadas del punto a graficarse con los valores iniciales de la tabla, evaluar la palabra de control para la invocación del algoritmo requerido, el control de pluma y el reconocimiento de fin de archivo.

Para la exposición del funcionamiento del programa, se tomará como base la estructura detallada de la tabla de datos mostrada en la figura 2.25, ya que es de ella de la que depende fundamentalmente. En esta, se muestran todos los posibles casos que se pueden presentar en la construcción de la misma, y los cuales deben de ser tomados en cuenta por el programa para el adecuado suministro de parámetros.

En general éstos son:

1. Generación de un segmento de línea seguido por otro segmento de línea.

2. Generación de un segmento de línea seguido por un círculo.
3. Generación de un círculo seguido por otro círculo.
4. Generación de un círculo seguido por un segmento de recta.

P1X bajo P1X alto	P4X bajo P4X alto	P7X bajo P7X alto
P1Y bajo P1Y alto	P4Y bajo P4Y alto	P7Y bajo P7Y alto
control línea (C1)	control círculo (C5)	control línea (C8)
P2X bajo P2X alto	P5X bajo P5X alto	P8X bajo P8X alto
P2Y bajo P2Y alto	P5Y bajo P5Y alto	P8Y bajo P8Y alto
control línea (C2)	control línea (C6)	control línea (C9)
P3X bajo P3X alto	P6X bajo P6X alto	P9X bajo P9X alto
P3Y bajo P3Y alto	P6Y bajo P6Y alto	P9Y bajo P9Y alto
control línea (C3)	control círculo (C7)	fin archivo (C10)

FIGURA 2.25.

Posibles alternativas en construcción tabla de datos.

Cabe hacer notar que cada coordenada de cada punto se divide en dos partes (alto y bajo), debido a que la longitud de éstas es de 12 bits, por lo que la parte baja

contiene los 8 primeros y la alta los 4 restantes.

Tomando esto en cuenta, las tareas que realiza el programa, así como el orden de éstas, son las siguientes:

Primeramente toma el valor de la coordenada X del punto 1 y la almacena en la localidad BUFX, para después hacer lo mismo con la coordenada Y de este punto, almacenandola en la localidad BUFY. Con esto, los programas de graficación tienen disponible el dato con el que posicionan a la pluma en el punto inicial de la gráfica a trazarse. En esta etapa, no importa si los primeros datos corresponden a un círculo o a una recta, ya que en ambos casos la información contenida en éstos, marcara el punto inicial de un elemento a graficarse.

Después de esto, almacena la palabra de control (C1) para su posterior utilización y direcciona nuevamente el inicio de la tabla (PIX bajo), almacenando también la dirección de C1.

Establece un apuntador situado en el primer punto de las dos parejas de coordenadas que definen al elemento a trazarse, y chequea el estado del bit1 de la palabra de control C1, el cual, como se trata de un segmento de recta, está en un estado bajo.

Con esto, invoca al algoritmo de rectas, proporcionandole primero todos los parámetros que requiere, esto es, encuentra las diferencias

$$dX = P2X - P1X$$

$$dY = P2Y - P1Y$$

determina la palabra de control para definir octante y realiza las asignaciones necesarias de datos, para que con esto, el algoritmo trabaje apropiadamente.

Ya que se generaron las coordenadas de los puntos pertenecientes a la recta, y se exhibieron vía los programas de graficación, el programa retoma la dirección de la localidad que almacena a C1, a la cual la incrementa cinco veces para direccionar a C2.

Teniendo direccionada esta palabra, chequea si el bit7 es uno o no. En caso afirmativo se trata de fin de archivo, pero como no es el caso, entonces repite el procedimiento anterior desde la etapa en que almacena la palabra de control (C2 en este caso).

Después de almacenar C2 y la dirección que a ésta almacena, entonces direcciona el punto P2X bajo y chequea el estado del bit1 de la palabra de control (C2), el cual, por tratarse de un segmento de recta, es cero. Con esto, invoca el algoritmo de rectas proporcionándole los parámetros requeridos a partir de las diferencias

$$dX = P3X - P2X$$

$$dY = P3Y - P2Y$$

Ya construido este nuevo segmento de recta, el programa retoma la dirección que almacena a la palabra de control C2, a la cual le adiciona cinco unidades y accesa a C3.

En esta etapa, evalúa el bit7 de esta palabra para determinar si se trata de fin de tabla. Como el estado de

este bit es bajo, entonces repite el proceso guardando la palabra de control así como la dirección en que está almacenada. Direcciona el punto P3X bajo y evalúa el estado del bit1 de C3.

Ya que el valor de éste es uno, se requiere el trazo de un círculo, por lo que invoca al algoritmo implementado para este fin, el cual se encarga de generar los puntos necesarios para la construcción de la gráfica a partir de los datos

RADIO = P4Y - P5Y

y origen en P5(x,y).

Después de esto, retoma la dirección que almacena la palabra de control C3 y la incrementa cinco veces, con lo que direcciona C6.

En esta etapa, cabe hacer mención que siempre que se genere un círculo, la siguiente palabra de control (C6 en este caso), deberá ser forzosamente la especificada para un segmento de recta (o fin de archivo). Esto se debe a que la acción que debe realizar el programa, es unir el punto final del círculo con el punto inicial de otro círculo o con el de un nuevo segmento de línea, acción que lleva a cabo por medio de una recta.

Por lo tanto, ya que acceso a C6, evalúa si el bit7 de ésta es cero o uno. Ya que no se trata de fin de archivo, éste no es cero, por lo que almacena el valor de la palabra de control, para su uso posterior, y también la dirección en la que está almacenada. Debido a que forzosamente grafica un

segmento de recta, no es necesario checar el bit1 de esta palabra, por lo que pasa directamente a suministrar los parámetros requeridos por el algoritmo de línea.

Para la generación de parámetros, el programa toma en cuenta que las coordenadas de los puntos extremos del segmento de recta a construirse son el final (y a la vez inicial) del círculo ya generado, y el inicial del siguiente elemento (otro círculo en este caso), por lo que las diferencias son

$$dX = P6X - P4X$$

$$dY = P6Y - P4Y$$

y es en base a éstas que se genera toda la información requerida por el algoritmo.

Generado ya este segmento de recta, retoma la dirección que almacena a C6 y le adiciona cinco unidades, con lo que dispone ahora de C7, de la cual checa el bit7 para reconocer fin de archivo, pero dado que no es el caso entonces almacena esta palabra de control y la dirección que la contiene, para direccionar nuevamente a P6X bajo.

Hecho esto, evalúa el estado del bit1 de C7, y como está en alto, invoca al algoritmo de círculos, proporcionándole el valor del radio en base a la diferencia

$$RADIO = P6Y - P7Y$$

y el origen en el punto con coordenadas P7(x,y). Con esto se generan, y se grafican via los programas de graficación, las

coordenadas de los puntos requeridos para la construcción de la gráfica.

Posteriormente, retoma la dirección que almacena a la palabra de control C7 y la incrementa cinco veces, con lo que accesa a C8. Con esto, checa si se trata del fin de archivo, pero dado que el bit7 de esta palabra es cero, almacena tanto a C8 así como la dirección que le corresponde, y directamente proporciona los parámetros para el algoritmo de línea, en base a las diferencias

$$dX = P8X - P6X$$

$$dY = P8Y - P6Y$$

con lo que se genera este elemento.

Nuevamente, retoma la dirección de la palabra de control (C8 en esta etapa), y la incrementa cinco veces, con lo que direcciona a C9. Como no se trata de fin de archivo, la almacena junto con la dirección que le corresponde y apunta al punto P8X en su parte baja. Evalúa el estado del bit1 de C9 y como está en bajo, invoca al algoritmo de rectas, proporcionando los parámetros requeridos a partir de

$$dX = P9X - P8X$$

$$dY = P9Y - P8Y.$$

Genera este último elemento y retoma la dirección de C9, incrementa a esta para acceder a C10 y checa el estado de su bit7. Como está en estado alto, reconoce el fin de archivo, por lo que termina el proceso.



Para la finalización de su tarea, el programa hace uso de la subrutina de graficación CASA, con lo que posiciona a la pluma en el punto situado en el extremo superior derecho del Área de graficado.

Como se puede observar en el desglose de las acciones que realiza este programa, en general las cuestiones en las que el programa debe poner especial atención son:

- Reconocer si la pareja de puntos accesada son los extremos de un segmento de recta o los puntos básicos de un círculo.

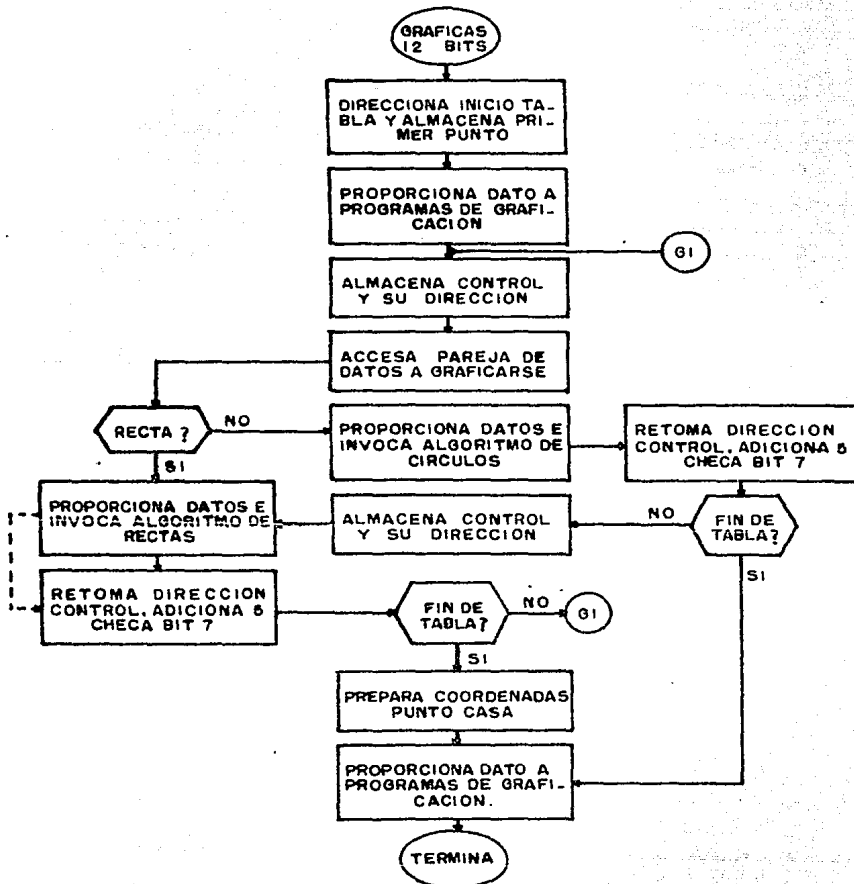
- En caso de ser una recta, después de generarla, direccionar la siguiente pareja de puntos tomando como punto inicial, el final del trazo anterior.

- Para el caso de un círculo, y después del trazo de éste, direccionar la siguiente pareja de puntos tomando como punto inicial el primer dato del trazo anterior, y como final el siguiente del que trabajo como centro del círculo. Adicionalmente, generará un segmento de recta con estos nuevos parámetros, forzosamente.

Otro aspecto que debe ser aclarado, es el uso del bit que controla la posición de la pluma (arriba o abajo). Este es utilizado por los programas de graficación, como se estudio anteriormente, y es debido a esto que cada vez que se accesa una palabra de control, ésta es almacenada en una localidad específica (PLUMA).

Con esto entonces, el diagrama de flujo de este

programa, en el que se contemplan todos los posibles casos que se pueden presentar en la construcción de la tabla de datos, así como la forma de suministrar parámetros a los diferentes algoritmos, es el siguiente:



La línea puntada que aparece en el diagrama anterior, se

refiere al caso cuando los dos puntos de la pareja accesada son iguales, esto es, cuando se tiene que realizar un movimiento nulo. El reconocimiento de esto se da en el período de generación de parámetros, y cuando se da, la acción que se realiza es la de acceder la siguiente pareja de datos sin invocar algoritmo alguno.

A continuación se listan las características principales de cada una de las subrutinas implementadas para la estructuración del programa generador de rectas y círculos (GRAF12), haciendo notar que algunas de éstas fueron ya explicadas, por lo que en estos casos solo se cita el nombre y la dirección de la subrutina en cuestión.

\* NOMBRE: GRAF12.

FUNCION: Es el encargado de proporcionar a los algoritmos de graficado los parámetros requeridos por éstos. Esto la hace en base a la estructura de la tabla de datos proporcionada por el usuario.

DIRECCION: 14F4H.

SUBROUTINAS QUE UTILIZA: POS, BREC12, RE+5B7, DIF2, DETO, ASIG, BREL12, CASA.

DATOS ENTRADA: Dirección inicial de tabla de datos en localidad DIR1.

DATOS SALIDA: Coordenadas puntos para la construcción de elementos geométricos.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY, IX.

\* NOMBRE: POS (explicada anteriormente).

DIRECCION: 13A0H.

\* NOMBRE: BREC12 (explicada anteriormente).

DIRECCION: 163DH.

\* NOMBRE: RE+5B7.

FUNCION: Recupera la dirección almacenada en la pila, le  
adiciona cinco unidades y checa el bit7 del  
contenido de la nueva dirección.

DIRECCION: 154FH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Registro BC cargado con 0000H.

DATOS SALIDA: Estado del bit checado en registro de  
banderas.

REGISTROS QUE MODIFICA: BC, DE, HL.

\* NOMBRE: DIF1 (explicada anteriormente).

DIRECCION: 158EH.

\* NOMBRE: DIF2 (explicada anteriormente).

DIRECCION: 1558H.

\* NOMBRE: DETO (explicada anteriormente).

DIRECCION: 15C4H.

\* NOMBRE: ASIG (explicada anteriormente).

DIRECCION: 152EH.

\* NOMBRE: BREL12 (explicada anteriormente).

DIRECCION: 17C9H.

\* NOMBRE: CASA (explicada anteriormente).

DIRECCION: 13BFH.

Para un estudio más detallado de estas subrutinas, en el apéndice A de este trabajo se pueden consultar los listados de cada una de ellas. Además, la función y la dirección de cada una de las localidades de servicio se muestra en el apéndice B.

### 2.3 Programas de transformación.

El tercer grupo, dentro del conjunto de programas desarrollados para el sistema graficador de 12 bits FAP-100, es el que se refiere al de transformaciones.

Por medio de éstos, es que los sistemas de graficado basados en computadora adquieren una de las características por la que se convierten en herramientas verdaderamente atractivas, pues dan al usuario la posibilidad de modificar una gráfica base sin tener que realizar las tareas usuales de recalcular nuevas coordenadas.

Esto es, el sistema se vuelve capaz de realizar este

trabajo en función solamente de ciertos parámetros de entrada, y con esto tener lista, en un tiempo considerablemente más corto que el utilizado por el humano, la nueva gráfica.

Dentro del sistema graficador del cual se ocupa este trabajo, se tiene acceso a cuatro tipos de transformación: Desplazamiento, Reducción, Amplificación y Rotación; las cuales modifican una determinada gráfica en relación a un sistema de coordenadas cartesianas cuyo origen coincide con el punto situado en el extremo inferior izquierdo del área de graficado (origen absoluto). Así, se le asigna un par de coordenadas  $(x,y)$  a cada punto de los que conforman a ésta.

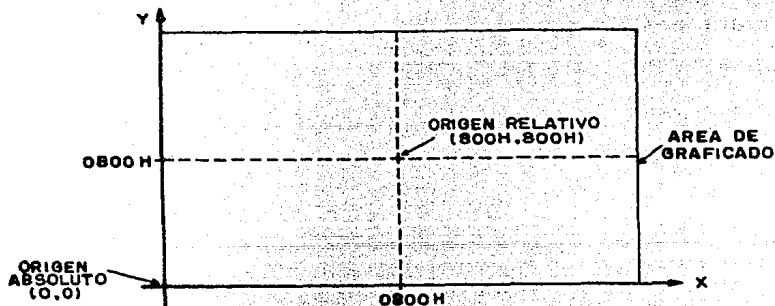


FIGURA 2.26.

Origen absoluto y relativo.

Adicionalmente, y con fines que se aclaran adelante, se define un origen relativo, el cual se puede situar en cualquier posición dentro del espacio disponible. Estos dos se muestran en la figura 2.26, en donde la posición que

guarda el origen relativo es la que el sistema proporciona de entrada, y cuyas coordenadas son  $X=0800H$ ,  $Y=0800H$ .

A continuación, se hace un análisis detallado de cada una de estas alternativas, mostrándose su funcionamiento así como la forma en que se implementa y las subrutinas desarrolladas para este fin.

### 2.3.1 Desplazamiento.

El objetivo de este tipo de transformación, es el de modificar la posición que guarda una determinada gráfica dentro del plano sin modificar su tamaño ni el ángulo que tiene con respecto a los ejes. Esto es, trasladarla dentro del mismo plano.

Para la realización de esta tarea, se toma como base el hecho de que un punto en el plano, puede ser desplazado a cualquier otra posición dentro del mismo plano, si a las coordenadas  $(x,y)$  del punto base  $(P)$ , se les adicionan los factores  $DX$  y  $DY$  respectivamente, lo que generará un nuevo punto  $(P')$  cuyas coordenadas están dadas por

$$X' = X + DX \quad \dots (2.32)$$

$$Y' = Y + DY \quad \dots (2.33)$$

donde  $DX$  y  $DY$  pueden ser cualquier número positivo o negativo.

Graficamente, este fenómeno se observa en la figura

2.27, de donde este hecho se puede generalizar para un conjunto de puntos, a partir de los cuales, si a todos y cada uno de ellos se les afecta por los mismos factores, se generará un nuevo conjunto, en el que sus elementos guardarán las mismas características entre si que las que se tenían en el conjunto origen, pero en otra región del plano.

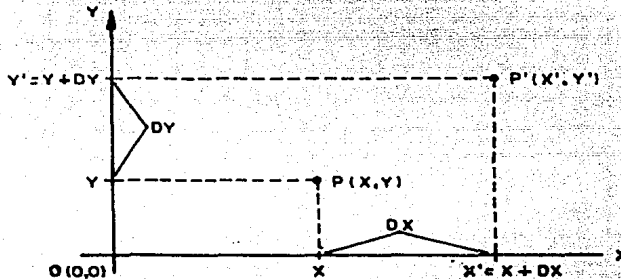


FIGURA 2.27.

Desplazamiento de un punto en el plano.

Es de esto de donde se parte para la implementación de esta alternativa de transformación como parte del sistema graficador, ya que si se toma en cuenta que dentro de éste, una gráfica está definida por los puntos extremos (o iniciales) de los elementos que la conforman, entonces estos se pueden afectar por determinados factores para lograr desplazarlos dentro del área de graficado.

Esto es, si a una determinada tabla de datos, como la mostrada en la figura 2.28(a), se le adicionan los factores  $DX$  y  $DY$  para cada una de las coordenadas respectivas de cada



punto, entonces se generará una nueva tabla, como la mostrada en 2.28(b), en donde el valor de cada punto de ésta, estará dado por (2.32) y (2.33), lo que equivale a un desplazamiento de la gráfica definida por la tabla dentro del plano.

X1		X1' = X1 + DX
Y1		Y1' = Y1 + DY
control		control
X2	+DX	X2' = X2 + DX
Y2		Y2' = Y2 + DY
control	+DY	control
X3		X3' = X3 + DX
Y3		Y3' = Y3 + DY
fin		fin

FIGURA 2.28.

Desplazamiento tabla de datos.

La forma en que se lleva a cabo este procedimiento no conlleva ninguna contradicción con la estructura del sistema, ya que la invocación que se realiza a los algoritmos de graficación para unir los puntos de la primera tabla, se puede realizar para que se unan los de la segunda, obteniéndose como resultado la gráfica definida por la primera con las modificaciones que acarrea la transformación. Esto implica también, que durante la transformación las palabras de control no deben sufrir modificación alguna.

Para una visualización más clara de este proceso de

transformación, se proporciona el siguiente ejemplo:

Se tiene una tabla de datos original cuyos puntos tienen los valores mostrados en la figura 2.29(a). Al graficar estos, por medio del algoritmo adecuado, se genera la figura mostrada con trazo continuo en la figura 2.29(c). Si estos puntos se modifican por los factores  $DX=2$  y  $DY=3$ , como se establece en (2.32) y (2.33), se obtendrá la tabla de datos de la figura 2.29(b), cuya representación gráfica es la mostrada con trazo discontinuo en la figura 2.29(c).

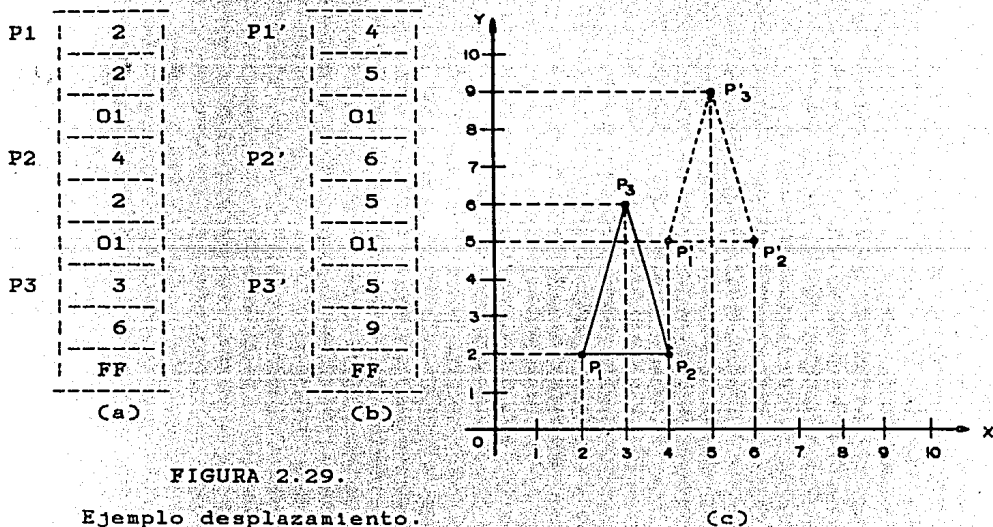


FIGURA 2.29.

Ejemplo desplazamiento.

Como se observa, la gráfica a sido desplazada en el mismo plano sin modificar sus características iniciales, por

lo que se alcanza el objetivo planteado.

Otro hecho que debe ser considerado, es que la translación o desplazamiento se realiza en relación al origen (el absoluto en el área de graficado), lo que nos lleva al problema de como realizar la transformación con respecto a un punto cualquiera (origen relativo), esto es, alguno situado en una posición arbitraria del plano.

Esto es llevado a cabo dividiendo el proceso en tres partes: trasladar la gráfica de manera que el origen relativo coincida con el absoluto, en donde la transformación se lleva a cabo para finalmente trasladar el origen relativo a su posición inicial. En realidad, la división de la tarea de esta forma parece tener poco sentido en esta alternativa, pero como se verá adelante, resulta de gran ayuda para los otros tipos transformación, y el seguirla para todas reditua en una programación más accesible.

Así, si se tuviera la gráfica producto de la tabla de datos de la figura 2.29(a), y se deseara trasladarla dos unidades en ambos ejes ( $DX=DY=2$ ), teniendo como origen relativo al punto con coordenadas  $X=3$ ,  $Y=3$ , los pasos que se deben seguir son:

- Transladar la gráfica de manera que el origen absoluto coincida con el relativo, esto es, restar tres unidades a cada una de las coordenadas de los puntos, con lo que se genera la tabla mostrada en la figura 2.30(a).

- Con esta nueva tabla, se realiza la aplicación de las

formulas de desplazamiento tomando los factores  $DX=2$  y  $DY=2$ , con lo que se generan los resultados de la tabla de la figura 2.30(b).

- Finalmente se desplaza la gráfica de manera que el origen relativo coincida con su posición inicial, es decir, se adicionan tres unidades a cada coordenada de los puntos de la última tabla, con lo que el resultado final es el mostrado en la figura 2.30(c).

-1
-1
01
1
-1
01
0
3
FF

(a)

1
1
01
3
1
01
2
5
FF

(b)

4
4
01
6
4
01
5
8
FF

(c)

FIGURA 2.30.

Desplazamiento con respecto origen relativo.

Graficamente, el proceso se muestra en la figura 2.31, en donde se observa que el desplazamiento ocurre con respecto al punto origen relativo. En esta, el trazo continuo es la gráfica inicial, mientras que el discontinuo corresponde a la

final.

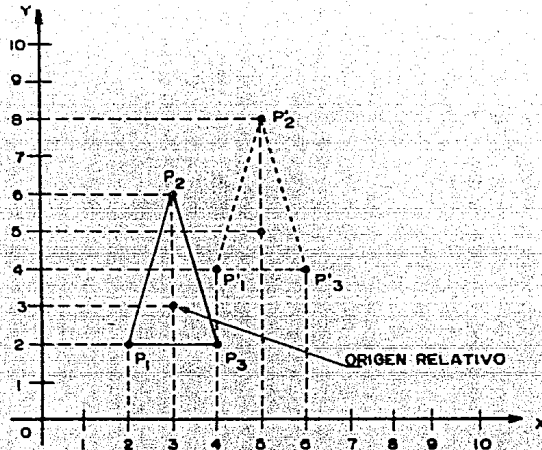
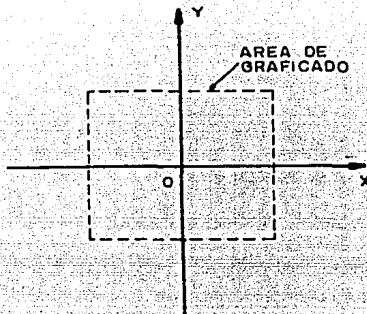


FIGURA 2.31.

Comportamiento gráfico desplazamiento  
con respecto origen relativo.

Dentro del sistema graficador FAP-100, se toma el criterio de que el área de graficado se comporta como una ventana en la que se exhibe parte de una gran gráfica total, esto es, solo una porción del plano, como se ilustra en la figura 2.32, por lo que cuando una gráfica se desplaza, entonces se desplaza también el origen relativo una distancia igual a la que sufre la gráfica, por lo que después de que se aplica el proceso de transformación, se debe actualizar la posición de este último, adicionándole a sus coordenadas los factores  $DX$  y  $DY$ .

FIGURA 2.32.  
Ventana en sistema FAP-100.



Con esto, las tareas que se realizan dentro del sistema para llevar a cabo este tipo de transformación son:

Se define inicialmente un código de identificación, ya que gran parte de la programación implementada, se comparte con otras alternativas de modificación y es por medio de este dato que se reconoce el tipo deseado.

Después de esto, se direcciona el inicio de la tabla de datos y se accesa la coordenada X del primer punto, a la cual se le resta la correspondiente del origen relativo, almacenada en la localidad de servicio ORGX, con lo que se lleva a cabo la translación hacia el origen.

Con este dato listo, se toma el factor DX, almacenado en la localidad FACTX, y por medio del código de identificación se lleva a cabo la transformación de desplazamiento, adicionándole este último al primero.

Hecho esto, se retoma la coordenada X del origen

relativo y se adiciona al dato desplazado, con lo que éste es trasladado hacia la posición inicial, quedando listo para el siguiente paso, que es el de almacenarlo nuevamente.

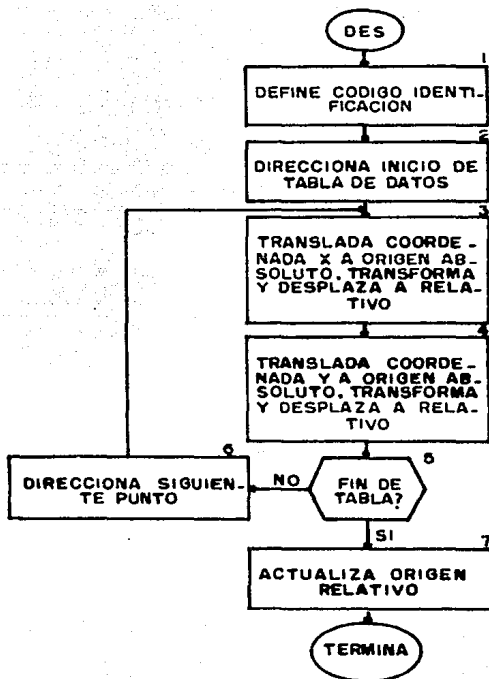
Este proceso es repetido para la coordenada Y del punto en cuestión, teniendo como diferencia que la componente correspondiente del origen relativo se toma de la localidad ORGY y el factor DY de FACTY.

Ya que el punto ha sido transformado en su totalidad, se checa si la palabra de control siguiente a éste es la especificada para el reconocimiento de fin de archivo, a partir de lo cual se define la siguiente acción a realizarse. Si no se trata del dato en cuestión, se direcciona al siguiente punto y lo transforma para llegar nuevamente a esta evaluación. En caso contrario, se da por terminada la tarea y se procede a actualizar la posición del origen relativo, que es la última etapa que se tiene que cubrir. Esto se realiza adicionando los factores DX y DY (FACTX, FACTY) a las coordenadas ORGX y ORGY respectivamente, con lo que el proceso de transformación es finalizado.

Para la implementación de estas tareas, se desarrolló la subrutina DES (DESplaza), la cual es la encargada de manejar otra serie de subprogramas con el fin de cubrir el objetivo planteado, y cuyo diagrama a bloques se muestra en la siguiente ilustración.

A continuación se listan las características principales de cada una de las subrutinas implementadas para llevar a cabo esta alternativa de transformación dentro del sistema

graficador, pero para hacer esto, conviene aclarar ciertas cuestiones.



Aunque con la subrutina DES se cubre el objetivo de trasladar la tabla de datos a otra posición dentro del plano, ésta no es la única función que realiza, sino que lleva a cabo algunas otras estrechamente ligadas al funcionamiento de la interfaz máquina-usuario. Explícitamente, la subrutina ESCALA, invocada por DES, realiza las funciones englobadas en los bloques del 2 al 6,



invocando a su vez a ESMAQ, pero adicionalmente hace uso de la subrutina ESCUSU para ejecutar acciones que necesariamente tienen que ser llevadas a cabo cada vez que se realiza la transformación. Estas otras funciones corresponden al conjunto de programas de comunicación, por lo que su explicación se hace posteriormente. Así, las subrutinas que se listan en esta sección, no son las que conforman en su totalidad a DES, por lo que esto deberá ser tomado en cuenta cuando se cite esta subrutina en secciones posteriores, teniéndose que relacionar la información que ahí se vierta con la que ahora se expone.

Otra cuestión importante es la referente a los factores DX y DY. Dentro de la subrutina DES (y constantemente en ESMAQ), se realiza un proceso de escalamiento de éstos, ya que por razones que en el siguiente capítulo se exponen, su rango de variación (almacenado en FACTX y FACTY) es de -9999 a +9999, lo que los hace incompatibles con el área de graficado (0-4096). Este proceso es llevado a cabo por medio de las subrutinas DIV y +-48, las cuales dentro de esta sección solo serán citadas, pues un estudio detallado de ellas en este momento sería aislado y poco entendible. En su lugar, la subrutina DIV se explica ampliamente en la siguiente sección, y su relación con +-48 se aborda en el estudio de la interfaz máquina-usuario, momento en que se aclarará la acción llevada a cabo y que por ahora solo queda indicada.

Tomando en cuenta lo anterior, las características

principales de las subrutinas utilizadas para llevar a cabo el proceso de desplazamiento son:

\* NOMBRE: DES.

FUNCION: Establece código de identificación para desplazamiento. Realiza transformación de tabla de datos y actualiza posición origen relativo.

DIRECCION: 20D9H.

SUBROUTINAS QUE UTILIZA: ESCALA, DIV, +-48.

DATOS ENTRADA: Coordenadas origen relativo en ORGX y ORGY.  
Dirección inicial de tabla de datos en localidad DIR1.

DATOS SALIDA: Datos desplazados a partir de dirección almacenada en DIR1. Coordenadas actualizadas del origen relativo en ORGX y ORGY.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: ESCALA.

FUNCION: Direcciona tabla de datos, translada a origen absoluto, realiza el desplazamiento requerido y regresa a la posición inicial los datos. Adicionalmente realiza otras funciones relacionadas con la interfaz máquina-usuario, las cuales se estudian en su momento.

DIRECCION: 210FH.

SUBROUTINAS QUE UTILIZA: REQPAR, ESCUSU, ESMAQ.

DATOS ENTRADA: Código de identificación en OP. Coordenadas

origen relativo en ORGX y ORGY. Dirección inicial tabla de datos en DIR1.

DATOS SALIDA: Tabla de datos desplazada a partir de dirección almacenada en DIR1.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

- \* NOMBRE: REQPAR (estrechamente ligada a interfaz máquina - usuario, por lo que su estudio se hace adelante).

DIRECCION: 2134H.

- \* NOMBRE: ESCUSU (estrechamente ligada a interfaz máquina - usuario, por lo que su estudio se hace adelante).

DIRECCION: 2186H.

- \* NOMBRE: ESMAQ.

FUNCION: Translada la tabla de datos hacia el origen absoluto. En base al código de identificación realiza la transformación requerida. Translada la tabla modificada hacia posición inicial.

DIRECCION: 21D9H.

SUBROUTINAS QUE UTILIZA: ESMEN, NUEDAT, NUECON, SUMIY5, DEFOPM, LETEC.

DATOS ENTRADA: Dirección inicial tabla de datos en DIR1. Factores DX y DY en FACTX y FACTY. Coordenadas origen relativo en ORGX y ORGY.

Código de identificación de transformación  
en OP.

DATOS SALIDA: Tabla transformada a partir de dirección  
almacenada en DIR1.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

- \* NOMBRE: NUEDAT (Almacena una coordenada x o y de un punto  
en memoria. Para un entendimiento más claro  
su estudio se hace en el siguiente  
capítulo).

DIRECCION: 1CD7H.

- \* NOMBRE: NUECON (Almacena una palabra de control en memoria.  
Su estudio se realiza en el siguiente  
capítulo).

DIRECCION: 1CE4H.

- \* NOMBRE: LETEC (Recibe un dato proveniente del teclado. Un  
entendimiento más claro se logra con su  
explicación detallada en el siguiente  
capítulo).

DIRECCION: 0136H.

- \* NOMBRE: SUMIY5.

FUNCION: Adiciona cinco unidades al valor almacenado en el  
registro IY.

DIRECCION: 1CEDH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a incrementarse en registro IY.

DATOS SALIDA: Dato incrementado en registro IY.

REGISTROS QUE MODIFICA: DE, IY.

\* NOMBRE: DEFOPM.

FUNCION: Define en base al código de identificación el tipo de transformación a realizarse. Esta puede ser desplazamiento, reducción o amplificación.

DIRECCION: 22D7H.

SUBROUTINAS QUE UTILIZA: En caso de amplificación CONV, en caso de reducción DIV y en caso de desplazamiento DESPM.

DATOS ENTRADA: Código de identificación en localidad OP.  
Dato a transformarse en HL. Factor DX o DY, según el momento en que se invoca, en localidad DIVOR.

DATOS SALIDA: Dato transformado en HL.

REGISTROS QUE MODIFICA: En amplificación BC, DE, HL. En reducción A, DE, BC, HL. En desplazamiento A, DE, BC, HL.

\* NOMBRE: DESPM.

FUNCION: Desplaza el dato almacenado en HL adicionandole el factor almacenado en DIVOR, el cual es escalado previamente. El dato puede ser cualquiera de las coordenadas x o y de un punto dado, dependiendo

del momento en que se invoca, al igual que el factor, el cual puede ser DX o DY.

DIRECCION: 22FDH.

SUBROUTINAS QUE UTILIZA: DIV, +-48.

DATOS ENTRADA: Dato a transformarse en HL. Factor transformador en localidad DIVOR.

DATOS SALIDA: Dato transformado en HL.

REGISTROS QUE MODIFICA: A, DE, BC, HL.

\* NOMBRE: DIV (utilizada fundamentalmente en reducción, por lo que su estudio se hace adelante).

DIRECCION: 1C5BH.

\* NOMBRE: +-48 (utilizada fundamentalmente por la interfaz máquina-usuario, por lo que su estudio se hace en el siguiente capítulo).

DIRECCION: 1CC1H.

### 2.3.2 Reducción.

Otra de las alternativas de transformación a las que se tiene acceso dentro del sistema de graficado FAP-100, es la de reducción. El objetivo de ésta, es la de hacer más pequeño el tamaño de una determinada gráfica, lograndose con esto, que el área de observación que se tiene del plano sea mayor. Esto es, cuando se reduce el tamaño de los elementos

visualizados, entonces el espacio que puede ser desplegado dentro del área de graficado es mayor.

Para la implementación de este fenómeno, se parte del hecho de que las coordenadas de un punto en el plano pueden ser reducidas en magnitud, y con esto la distancia del punto con respecto al origen, si a cada una de ellas se les multiplica por los factores  $R_X$  y  $R_Y$  respectivamente, con lo que se tiene un nuevo par de coordenadas de la forma

$$X' = X * R_X \quad \dots (2.34)$$

$$Y' = Y * R_Y \quad \dots (2.35)$$

donde  $X'$  siempre es menor que  $X$ , y  $Y'$  que  $Y$ , por lo que  $R_X$  y  $R_Y$  deben ser siempre menores que 1, pero mayores que cero, pues si se presentara este último caso, toda la información de la gráfica se perdería.

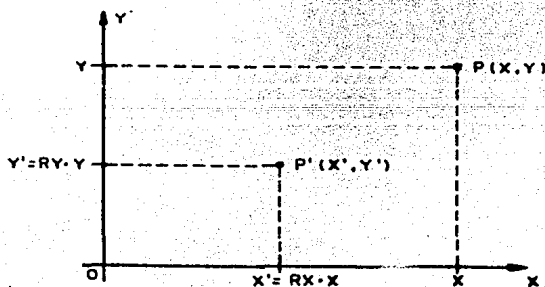


FIGURA 2.33.

Efecto de reducción en un punto del plano.

Graficamente, este proceso se muestra en la figura 2.33, donde el punto inicial  $P(x,y)$  es afectado por  $RX$  y  $RY$ , produciendo el nuevo punto  $P'(x,y)$ .

Este hecho, al igual que con la alternativa anterior, puede ser generalizado a un conjunto de puntos en el plano, los cuales, si se ven afectados por los mismos factores, se reducirán sus magnitudes en la misma proporción, por lo que los puntos del nuevo conjunto mantendrán, proporcionalmente, las mismas características que guardaban en el conjunto original.

Es precisamente del concepto anterior del que se parte para la implementación de esta alternativa dentro del sistema graficador, ya que en éste, una gráfica es representada únicamente por dos puntos extremos de cada elemento. Así, si éstos son afectados por los mismos factores  $RX$  y  $RY$  (variando siempre éstos entre 0 y 1), en la forma que se establece en (2.33) y (2.34), se generarán otros nuevos puntos los cuales pueden ser unidos por los algoritmos adecuados para producir la nueva gráfica transformada.

Para un mejor entendimiento de esto, se proporciona el ejemplo mostrado en la figura 2.34, donde la tabla original (etiquetada con la letra a), es afectada por los factores  $RX=RY= 0.5$ , con lo que se produce la tabla mostrada en 2.34(b), donde se observa que las palabras de control no deben ser modificadas durante el proceso de transformación, ya que ello acarrearía problemas con la invocación de los algoritmos de graficado.



8		4
8		4
01		01
8		4
14		7
01	X*0.5	01
14	Y*0.5	7
14		7
01		01
14		7
8		4
FF		FF
(a)		(b)

FIGURA 2.34.

Ejemplo reducción.

Graficamente, este proceso se muestra en la figura 2.35, donde el dibujo con línea continua muestra los valores de la tabla inicial, mientras que el línea discontinua la final.

En esta figura se puede observar como la relación entre dos puntos primados es, en proporción, la misma que la de los no primados, por lo que al proceso de reducción se le conoce como uniforme, y se da siempre y cuando los factores RX y RY sean iguales. Cuando se da el caso de que RX sea distinto de RY, las características de la figura inicial cambian, y al

proceso se le conoce como reducción diferencial.

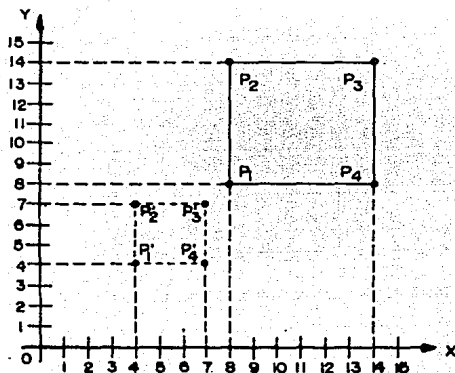


FIGURA 2.35.

Comportamiento gráfico figura 2.34.

Otra cuestión importante que se debe tomar en consideración, es que la transformación se realiza en relación al origen del plano, por lo que se presenta el hecho de como llevar a cabo ésta con respecto a un punto cualquiera, es decir, con respecto a un origen relativo.

Para llevar a cabo esto, se sigue con el mismo procedimiento citado en la sección anterior, es decir: se translada la gráfica de manera que el origen relativo coincida con el absoluto, se realiza la transformación con respecto a este último y finalmente se translada nuevamente la gráfica obtenida haciendo que el origen relativo, retome su posición original. Con esto se alcanza el objetivo planteado y además se tiene la ventaja de seguir los mismos

pasos que en la alternativa de desplazamiento, por lo que mucho de la programación implementada puede ser aprovechada, teniéndose como única diferencia el tipo de operación con que se ven afectados los datos.

-3	-1.5	9.5
-3	-.75	10.25
01	01	01
-3	-1.5	9.5
3	.75	11.75
01	01	01
3	1.5	12.5
3	.75	11.75
01	01	01
3	1.5	12.5
-3	1.5	10.25
FF	FF	FF

(a) (b) (c)

FIGURA 2.36.

Reducción con respecto a un origen relativo.

Este proceso se ejemplifica en la figura 2.36, en donde se toma como tabla base la de la figura 2.34(a), teniéndose como origen relativo el punto con coordenadas  $X=11$ ,  $Y=11$ , y como factores  $R_X=0.5$  Y  $R_Y=0.25$ . Así, siguiendo los pasos citados, se translada la gráfica hacia el origen absoluto,

generandose la tabla de la figura 2.36(a), a la que se le aplican las ecuaciones (2.34) y (2.35) en cada uno de los puntos, lo que por resultado da la tabla trasformada etiquetada con la letra (b) de la misma figura. Con esto, se desplaza nuevamente el origen relativo a su posición inicial, dandose por terminada la transformación al obtener la tabla de la figura 2.36(c).

Graficamente la transformación se muestra en la figura 2.37, donde el trazo con línea continua pertenece a la tabla inicial, mientras el de la línea discontinua a la final. Adicionalmente, se observa claramente el efecto de llevar a cabo una transformación diferencial (RX distinto de RY).

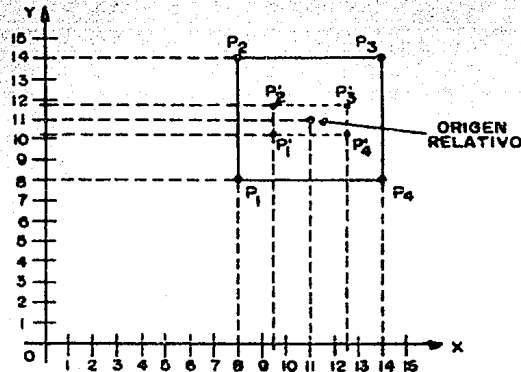


FIGURA 2.37.

Resultado gráfico ejemplo figura 2.36.

Para la implementación de esta alternativa dentro del sistema graficador, se tuvo el problema que la

microcomputadora huesped (FAP-100), no tiene la capacidad de manejar números reales, por lo que el valor de los factores de escalamiento se implementó como el recíproco de cada uno de ellos, es decir, de la forma

$$1/X$$

donde X es un número entero mayor o igual que 1.

Con esto, las ecuaciones (2.34) y (2.35) adquieren la siguiente estructura

$$X' = X / rX \quad \dots (2.36)$$

$$Y' = Y / rY \quad \dots (2.37)$$

en donde rX y rY son los factores de escalamiento, los cuales producen el mismo efecto que en las ecuaciones originales.

El manejar de esta forma la alternativa, creo el problema adicional de tener que ejecutar divisiones, operación que tampoco es capaz de realizar por sí misma la microcomputadora, y para lo cual se implementó un algoritmo especial para llevar a cabo esta tarea.

Este algoritmo es concido como de división secuencial de corrimiento substracción/adición sin realmacenamiento, el cual es capaz de llevar a cabo la división de un número de 24 bits entre otro de 12, lo que genera un tercero de la misma longitud que el segundo. Adicionalmente, la naturaleza de éste, permite realizar esta tarea en base a corrimientos, adiciones y substracciones, con lo que su implementación resulto ser sencilla. Un análisis detallado de éste, se

presenta en el apéndice C de este trabajo.

La implementación del algoritmo de división, se llevo a cabo por medio de la subrutina DIV (utilizada para escalar factores en DES), la cual requiere para su funcionamiento que el dividendo le sea suministrado en el registro HL, y el divisor en la localidad de servicio DIVOR, con lo que entrega el cociente (Número entero de 12 bits) en HL.

Con esto, las tareas que se deben de realizar para llevar a cabo la alternativa de reducción son:

Primero se define el código de identificación para esta alternativa, ya que como se citó anteriormente, gran parte de la programación se comparte con las otras alternativas.

Después de esto, se direcciona el inicio de la tabla de datos a transformarse y se accesa la coordenada X del primer punto, la cual es trasladada hacia el origen absoluto, restandole la coordenada correspondiente del origen relativo, almacenada en la localidad ORGX.

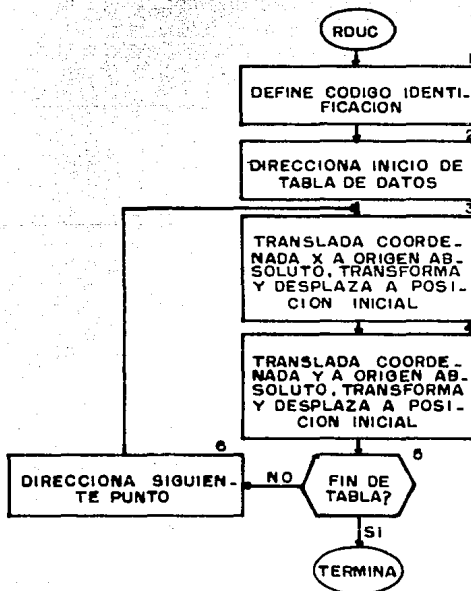
Con este dato listo, se almacena el factor de escalamiento para el eje X en la localidad DIVOR, con lo que se invoca a la subrutina DIV y se lleva a cabo la transformación. Finalmente, se adiciona a este dato transformado la coordenada X del origen relativo y se almacena el resultado final en la posición adecuada de la tabla de valores.

Este mismo proceso es llevado a cabo para la coordenada Y de este punto, con la diferencia de que se

utiliza la coordenada Y del origen relativo, almacenada en ORGY, y el factor de escalamiento determinado para el eje Y.

Después de que se transforma esta última coordenada, se accesa la palabra de control correspondiente a este último punto y se checa si es el fin de archivo. En caso negativo, se accesa al siguiente punto y se repite el proceso. En caso afirmativo se da por terminada la transformación.

Cabe hacer notar, ciertas diferencias que existen en relación a la alternativa anterior: Primero, los factores no necesitan ser escalados, pues el dato proporcionado se puede utilizar sin modificarlo, y segundo, el origen relativo no se modifica.



Dentro del sistema graficador, la subrutina encargada de realizar esta tarea, recibe el nombre de RDUC (ReDUCción), de la cual la ilustración anterior muestra su diagrama a bloques.

Para su funcionamiento, esta subrutina hace uso a su vez de la subrutina ESCALA, la cual es utilizada también por DES, por lo que se hace necesario aclarar el uso de ésta.

Como se anoto anteriormente, muchas de las acciones realizadas en DES, se llevan a cabo en RDUC, y son estas las que realiza ESCALA, pero adicionalmente, esta última debe de reconocer por cual de las dos primeras ha sido invocada, para lo que se vale del código de identificación. A partir de esto, entonces ESMAQ (parte de ESCALA), define la operación a realizarse para la transformación requerida vía DESOPM, y es cuando se llevan a cabo las acciones que diferencian un proceso de otro.

Tomando lo anterior en cuenta, a continuación se listan características principales tanto de las subrutinas que son utilizadas unicamente por esta alternativa de transformación, como las que quedaron pendientes en la sección anterior, y que corresponden a la operación requerida por RDUC. Las subrutinas que corresponden a las acciones comunes en las dos alternativas solo son citadas.

\* NOMBRE: RDUC.

FUNCION: Reduce el tamaño de una determinada gráfica



dividiendo cada una de las coordenadas de los puntos que la conforman por los factores  $r_X$  y  $r_Y$ , según sea el caso. Adicionalmente, define el código de identificación correspondiente a esta alternativa.

DIRECCION: 20C7H.

SUBROUTINAS QUE UTILIZA: ESCALA.

DATOS ENTRADA: Coordenadas origen relativo en localidades  $ORG_X$  y  $ORG_Y$ . Dirección inicial de tabla a transformarse en  $DIR_1$ .

DATOS SALIDA: Tabla de datos transformada a partir de dirección almacenada en  $DIR_1$ .

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: ESCALA (explicada anteriormente y de la cual solo quedaron pendientes

CONV y DIV).

DIRECCION: 210FH.

\* NOMBRE: CONV (utilizada en la alternativa amplificación, por lo que su estudio se realiza posteriormente).

DIRECCION: 1BF8H.

\* NOMBRE: DIV.

FUNCIÓN: Realiza la división de un número binario de 24 bits de longitud entre otro de 12. Genera un

cociente de la misma longitud que el segundo. Trabaja aritmética entera únicamente, por lo que el residuo que se obtiene al llevarse a cabo la operación es desechado.

DIRECCION: 1C5BH.

SUBROUTINAS QUE UTILIZA: COMP2.

DATOS ENTRADA: Dividendo en registro HL, y divisor en localidad DIVOR.

DATOS SALIDA: Cociente en registro HL.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: COMP2.

FUNCION: Obtiene el complemento a dos de un número binario de 16 bits.

DIRECCION: 1B55H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Número a complementarse en HL.

DATOS SALIDA: Número en complemento a dos en HL.

REGISTROS QUE MODIFICA: A, DE, HL.

Para un estudio detallado de estas subrutinas, es necesario consultar el apéndice A de este trabajo, donde se expone el listado completo de cada una de ellas. La función y dirección de las localidades de servicio citadas, se muestran en el B.

## 2.3.3 Amplificación.

El hecho básico de esta tercer alternativa de transformación, es que la magnitud de un punto dentro del plano puede ser aumentada si a su coordenada X se le multiplica por un factor AX y a la Y por uno AY, con lo que se logra la creación de un nuevo punto cuyas coordenadas tienen la siguiente estructura

$$X' = X \cdot AX \quad \dots (2.38)$$

$$Y' = Y \cdot AY \quad \dots (2.39)$$

donde las coordenadas primadas siempre son mayores o iguales que las originales, y los factores de amplificación son mayores o iguales que la unidad.

Graficamente, esto se ilustra en la figura 2.38, donde P es el punto original y P' es el transformado.

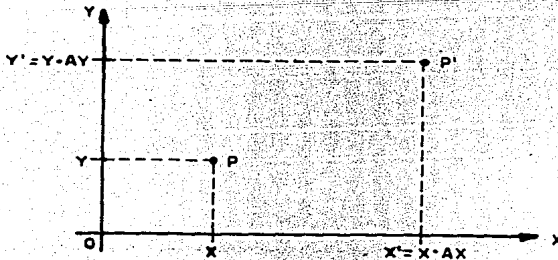


FIGURA 2.38.

Transformación de amplificación sobre un punto.

Si esto se aplica a un conjunto de puntos, en donde a cada uno de ellos se les afecta con los mismos factores, entonces se logrará que las características de las relaciones entre ellos se amplifiquen en la misma proporción.

Así, si a una tabla de datos perteneciente al sistema graficador se le aplican las ecuaciones (2.28) y (2.39) para cada uno de sus puntos, se generará una nueva tabla, la cual puede ser tratada como la original y con lo que se construirá la gráfica ampliada de la primera.

2		8
2		8
01		01
2		8
3		12
01	$X*4$	01
3	$Y*4$	12
4		16
01		01
4		16
2		8
FF		FF
(a)		(b)

FIGURA 2.39.

Ejemplo tabla ampliada.

El proceso que se debe seguir para lograr el objetivo planteado es el mismo que con las alternativas anteriores, y se ejemplifica en la tabla mostrada en la figura 2.39(a), la cual se ve afectada por los factores  $AX=AY=4$ , con lo que se genera la etiquetada con la letra b en la misma figura.

Graficamente, la acción se ilustra en la figura 2.40, donde la gráfica trazada con línea continua corresponde a la tabla original, mientras que la de línea discontinua a la transformada.

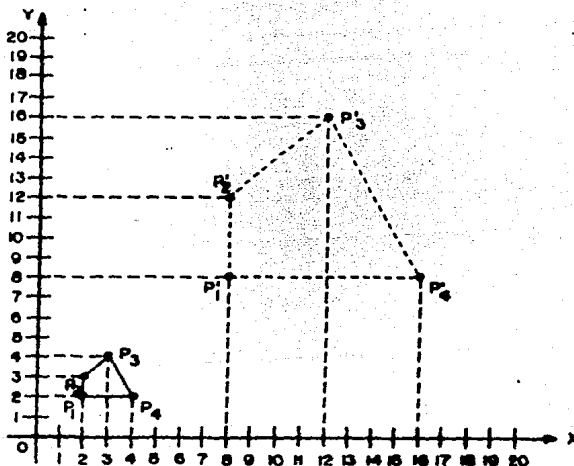


FIGURA 2.40.

Resultado gráfico ejemplo figura 2.39.

En ésta, se puede observar que la ventaja que presenta esta alternativa es la de mostrar más claramente las características de la figura original, ya que las de la

transformada son, en proporción, las mismas que las que regían a la primera, como se anotó anteriormente.

Adicionalmente, se presenta el hecho de que, al igual que las transformaciones anteriores, el cambio se da con respecto al origen absoluto, pudiéndose realizar con respecto a un origen relativo siguiendo los mismos pasos que con las alternativas de desplazamiento y reducción, lo que proporciona la ventaja de poder utilizar gran parte de la programación implementada para estas últimas.

Con esto entonces, las etapas que se deben cumplir para amplificar una figura con respecto a un punto cualquiera son: Transladar la gráfica de manera que el origen relativo coincida con el absoluto, realizar la transformación en esta región del plano y finalmente posicionar la gráfica transformada de manera que el origen relativo retome sus coordenadas iniciales.

Cabe hacer notar, que para esta transformación se pueden dar dos diferentes tipos, al igual que con reducción:

- a. Uniforme. Cuando los factores  $AX$  y  $AY$  son iguales.
- b. Diferencial. Cuando los factores de escalamiento son distintos.

El ejemplo de la figura 2.41 muestra el proceso total que se requiere para llevar a cabo la transformación de amplificación, observándose que se trata del tipo diferencial, ya que  $AX=2$  y  $AY=4$ . En este, la figura 2.41(a) muestra la tabla original, la etiquetada con la letra (b) la trasladada hacia el origen (tomando como origen relativo

el punto  $X=Y=10$ ), en la tercera se exponen los valores transformados por (2.38) y (2.39), y finalmente, la (d) muestra la tabla trasladada a la posición inicial.

Graficamente la transformación se muestra en la figura 2.42, donde la gráfica continua corresponde a la tabla inicial.

8	-2	-4	6
8	-2	-8	2
01	01	01	01
12	2	4	14
8	-2	-8	2
01	01	01	01
10	0	0	10
13	3	12	22
FF	FF	FF	FF
(a)	(b)	(c)	(d)

FIGURA 2.41.

Amplificación con respecto a un origen relativo.

Graficamente la transformación se muestra en la figura 2.42, donde la gráfica continua corresponde a la tabla inicial.

Para la implementación de esta alternativa dentro del sistema graficador, se presento el problema de que la microcomputadora FAP-100 no puede realizar por si misma la

operación multiplicación, requerida para llevar a cabo el escalamiento deseado, para lo cual se desarrollo una subrutina especial para este fin. Esta recibe el nombre de CONV (CONVierte), y en ella se realiza el producto de dos números en base al algoritmo conocido como de adición múltiple. Para esto, el multiplicando se debe entregar en el registro DE, el multiplicador en BC y HL debe ser cargado con cero.

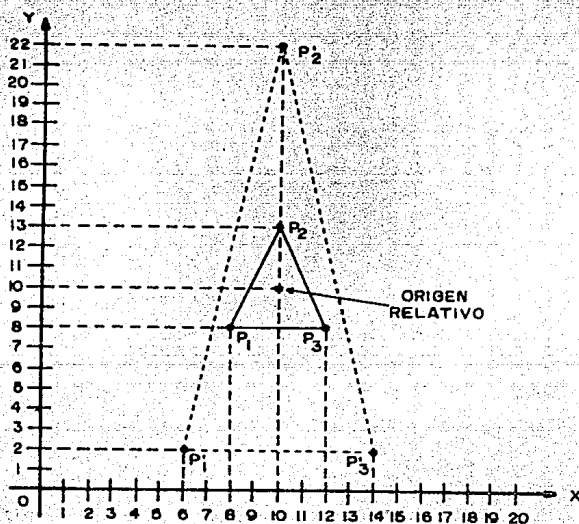


FIGURA 2.42.

Resultado gráfico ejemplo figura 2.41.

Este algoritmo permite obtener el producto en base a adicionar  $n$  veces el multiplicando, donde  $n$  es el multiplicador de la operación. Para un estudio detallado, en



el apéndice D de este trabajo se expone íntegramente este proceso.

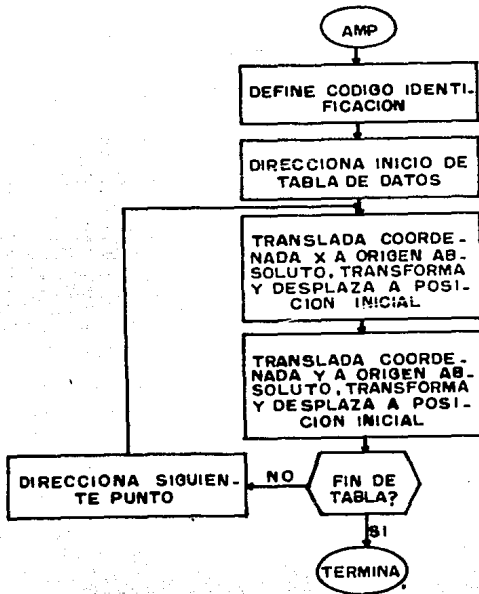
Tomando en cuenta lo anterior, los pasos que se deben de seguir para amplificar una determinada gráfica son: Se define el código de identificación para amplificación, ya que también en este caso la subrutina ESCALA es utilizada. Después de esto se direcciona el inicio de la tabla de datos y se accesa la coordenada X del primer punto. Este es trasladado hacia el origen absoluto, restandole la coordenada X del relativo, almacenada en ORGX. En este momento se establecen los parámetros requeridos por CONV, almacenando en BC el valor del factor AX, residente en FACTX, y se le invoca, con lo que la transformación se realiza. Finalmente, al dato transformado se le adiciona nuevamente el contenido de ORGX y el proceso termina para esta coordenada.

Para la coordenada Y, que es accesada inmediatamente después de realizar las acciones anteriores, se llevan a cabo los mismos pasos, con la diferencia que la coordenada de origen relativo utilizada es la Y (ORGY), y el factor de escalamiento que le afecta es el almacenado en FACTY.

Hecho esto, se accesa la palabra de control correspondiente y se pregunta si se trata de fin de archivo. En caso afirmativo la transformación termina, en caso contrario se accesa al siguiente punto y el proceso se repite.

La subrutina encargada de llevar a cabo esta tarea recibe el nombre de AMP (AMplificación), y a continuación se

muestra su diagrama a bloques:



Como se observa en este diagrama, la estructura seguida para esta alternativa es igual que la utilizada para reducción y desplazamiento, y es en realidad la subrutina ESCALA (via ESMAQ) la que realiza casi todas estas acciones, siendo dentro de ésta donde por medio del código de identificación la subrutina DEFOPM invoca a CONV.

A continuación se listan las características principales de las subrutinas implementadas para esta tarea, aclarando: solo se exponen las subrutinas que no han sido explicadas en secciones anteriores, citandose solamente las restantes. Y el estudio de las subrutinas pertenecientes a ESCALA que

están relacionadas con la interfaz máquina-usuario sigue pendiente hasta su momento.

\* NOMBRE: AMP.

FUNCION: Amplifica el tamaño de una determinada gráfica multiplicando las coordenadas de cada punto por los factores AX y AY, según corresponda. Adicionalmente, define el código de identificación para esta alternativa.

DIRECCION: 20DOH.

SUBROUTINAS QUE UTILIZA: ESCALA.

DATOS ENTRADA: Coordenadas origen relativo en localidades ORGX y ORGY. Dirección inicial de tabla a transformarse almacenada en DIR1.

DATOS SALIDA: Tabla de datos transformada a partir de la dirección almacenada en DIR1.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: CONV.

FUNCION: Realiza el producto de dos números en base al algoritmo de adición múltiple.

DIRECCION: 1BFSH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Registro HL cargado con cero. Multiplicando en registro DE. Multiplicador en registro BC.

DATOS SALIDA: Producto en registro HL.

REGISTROS QUE MODIFICA: BC, HL.

El listado completo de estas subrutinas, se expone en el apéndice A de este trabajo, teniendo que consultarlo para un estudio detallado de ellas. En el apéndice B, se lista la dirección y función específica de cada una de las localidades de servicio utilizadas.

#### 2.3.4 Rotación.

La cuarta y última de las alternativas de transformación a las que se tiene acceso dentro del sistema graficador FAP-100, es la de rotación. El objetivo final de ésta, es la de girar alrededor de un punto una determinada gráfica, sin modificar sus características originales.

Para llevar a cabo esta tarea, se parte del hecho de que si las coordenadas de un punto (P) dentro del plano, se les afecta por los valores del seno y del coseno del ángulo de rotación (cuantificado en sentido antihorario), en la forma en que se muestra en 2.40 y 2.41, se obtendrán las coordenadas de un nuevo punto (P') el cual mantendrá la misma distancia hacia el origen que el primero, pero rotado  $\theta$  grados. Las coordenadas de este último tienen la siguiente estructura:

$$X' = X \cos \theta - Y \sin \theta \quad \dots (2.40)$$

$$Y' = X \operatorname{sen} \Theta + Y \operatorname{cos} \Theta \quad \dots (2.41)$$

Estas ecuaciones se derivan facilmente, si se observa el comportamiento del proceso antes citado, el cual es ilustrado en la figura 2.43, donde P es el punto original, P' el rotado y  $\Theta$  el ángulo de rotación.

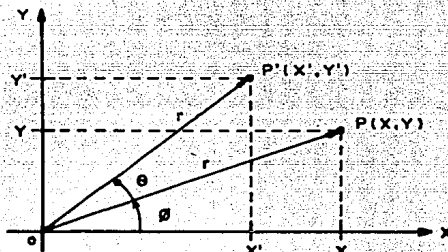


FIGURA 2.43.

Rotación de un punto en el plano.

De esta, se observa que

$$X = r \operatorname{cos} \beta \quad \dots (2.42)$$

$$Y = r \operatorname{sen} \beta \quad \dots (2.43)$$

y que

$$X' = r \operatorname{cos}(\Theta + \beta) \quad \dots (2.44)$$

$$Y' = r \operatorname{sen}(\Theta + \beta) \quad \dots (2.45)$$

pero por identidades trigonométricas se tiene que

$$R \operatorname{cos}(A+B) = R \operatorname{cos}A \operatorname{cos}B - R \operatorname{sen}A \operatorname{sen}B$$

$$R \operatorname{sen}(A+B) = R \operatorname{sen}A \cos B - R \cos A \operatorname{sen}B$$

por lo que las coordenadas primadas adquieren la siguiente forma

$$X' = r \cos \theta \cos \phi - r \operatorname{sen} \theta \operatorname{sen} \phi$$

$$Y' = r \operatorname{sen} \theta \cos \phi + r \cos \theta \operatorname{sen} \phi$$

en donde, si se sustituyen (2.42) y (2.43), se obtienen las ecuaciones básicas de la transformación (2.40) y (2.41).

Si este principio se aplica a un conjunto de puntos distribuidos de alguna manera sobre el plano, modificando las coordenadas de cada uno de éstos de la misma manera, se logrará la generación de un nuevo conjunto distribuido de igual forma que el original pero rotado una cierta cantidad de grados con respecto a un punto.

Así, si todos los elementos que constituyen una gráfica dentro del sistema FAP-100, se ven afectados por las ecuaciones (2.40) y (2.41) en la misma forma se obtiene como resultado la misma gráfica pero rotada  $\theta$  grados en sentido anti-horario y con respecto al origen absoluto.

El ejemplo de la figura 2.44 ilustra este proceso, donde la tabla primera es afectada tomando como ángulo de rotación el de 45 grados, con lo que se obtiene la tabla etiquetada con la letra b de la misma figura.

Graficamente, este proceso se muestra en la figura 2.45, donde la ilustración de trazo continuo corresponde a la tabla inicial, mientras que la punteada a la final.

10		7.071
0		7.071
01		01
15		11.32
0	$X \cdot 0.7071 - Y \cdot 0.7071$	11.32
01	$X \cdot 0.7071 + Y \cdot 0.7071$	01
13		3.535
8		14.85
FF		FF
(a)		(b)

FIGURA 2.44.

Proceso de rotación en el plano ( $\theta = 45$  grados).

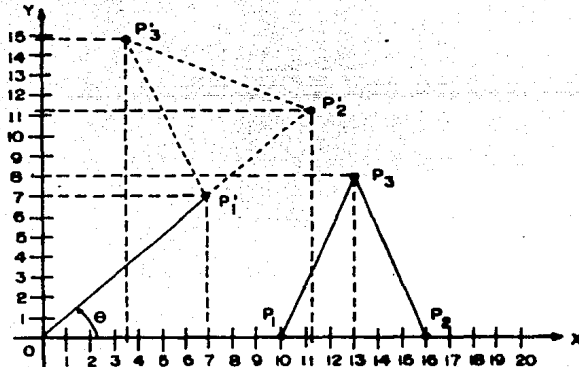


FIGURA 2.45.

Resultado gráfico ejemplo figura 2.44.

Como se puede observar, la restricción que presenta el llevar a cabo la transformación de esta forma, es que el punto alrededor del cual gira la gráfica es el origen absoluto, por lo que para realizar ésta con respecto a uno relativo, es necesario utilizar el mismo procedimiento que con las alternativas anteriores, esto es: Transladar la figura de manera que el origen relativo coincida con el absoluto, realizar la transformación con respecto a este último y finalmente reposicionar al primero de manera que adquiera sus coordenadas originales.

8	-2	2	12
8	-2	-2	8
01	01	01	01
12	2	2	12
8	-2	2	12
01	01	01	01
10	0	-3	7
13	3	0	10
FF	FF	FF	FF
(a)	(b)	(c)	(d)

FIGURA 2.45.

Rotación con respecto a un origen relativo.

Esta serie de etapas se ilustran en la figura 2.46,



donde los parámetros requeridos son: coordenadas origen relativo  $X=10$ ,  $Y=10$ , ángulo de rotación = 90 grados y datos a transformarse en la tabla etiquetada con la letra a. Con esto, se obtiene la tabla (b) que es la correspondiente a la trasladada hacia el origen absoluto, y la (c) en donde se exponen las coordenadas de los puntos transformados. La figura 2.45(d), muestra la tabla final, es decir la trasladada hacia la posición inicial.

La figura 2.47 muestra con trazo continuo la gráfica original, y con discontinuo la final correspondientes a las tabla (a) y (d) del ejemplo anterior. En esta, se observa como la última modificación al algoritmo original produce el efecto deseado.

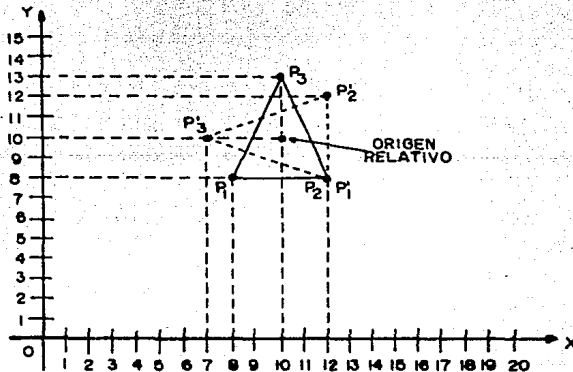


FIGURA 2.47.

Resultado gráfico del ejemplo de la figura 2.46.

Es necesario aclarar que aunque el proceso de transformación con respecto a un origen relativo es exactamente el mismo que el seguido en las alternativas de transformación anteriores, excepto por el tipo de operación requerido en cada una de ellas, no se utiliza el programa disponible para éstas, sino que se optó por desarrollar un conjunto de subrutinas especiales para este fin.

Lo anterior se debe fundamentalmente a que aunque en lo general son iguales los procesos para todas, en lo particular la rotación implica muchos detalles diferentes a las otras, y el principal de estos es que, mientras en las tres primeras no importa si se trata de rectas o círculos, en la última esto es determinante, ya que en el rotado de círculos se debe poner especial atención a la disposición que tienen que guardar el centro de éste y el punto inicial del trazo, es decir, que después de la transformación se debe conservar la misma relación entre ellos para que la figura no pierda sus características.

Para esto, es necesario considerar dos casos generales:

a. Cuando el círculo a rotarse tiene como centro el origen relativo con respecto al cual se lleva a cabo la transformación.

b. Cuando el centro del círculo está alejado una determinada distancia del origen relativo.

En el primer caso, el algoritmo debe ser capaz para no modificar en absoluto la disposición de los parámetros del círculo, ya que un círculo rotado sobre su centro siempre

conserva las mismas características, mientras que si se presenta el segundo caso, se tendrán que seguir varias etapas para llevar a cabo la transformación: Primero se resguarda el valor del radio del círculo almacenándolo temporalmente, con lo que se procederá a la rotación del punto inicial del trazo considerándolo un punto semejante a los extremos de una recta. Después de esto se procede a estructurar el centro del círculo rotado asignándole a su coordenada X el mismo valor que la correspondiente del nuevo punto inicial, y a la Y la resta de la coordenada Y del punto inicial rotado menos el valor del radio previamente almacenado.

Con esto, el rotado de círculos se hará correctamente no importando la posición que tenga en el plano.

En lo que respecta a la rotación de rectas, ésta no presenta ningún problema, ya que se realiza transformando independientemente sus puntos extremos, como se ilustró anteriormente.

Así, el proceso que se sigue para llevar a cabo esta tarea es:

Primero se direcciona el inicio de la tabla de datos a transformarse. Teniendo esto, y por medio de la palabra de control correspondiente al punto que se tiene direccionado, se checa si es el último del conjunto. En caso afirmativo éste se considera como el extremo de una recta y como tal, se traslada hacia el origen absoluto, se transforma y se desplaza a la región original del plano, para con esto terminar la transformación.

Es necesario aclarar que aunque el proceso de transformación con respecto a un origen relativo es exactamente el mismo que el seguido en las alternativas de transformación anteriores, excepto por el tipo de operación requerido en cada una de ellas, no se utiliza el programa disponible para éstas, sino que se optó por desarrollar un conjunto de subrutinas especiales para este fin.

Lo anterior se debe fundamentalmente a que aunque en lo general son iguales los procesos para todas, en lo particular la rotación implica muchos detalles diferentes a las otras, y el principal de estos es que, mientras en las tres primeras no importa si se trata de rectas o círculos, en la última esto es determinante, ya que en el rotado de círculos se debe poner especial atención a la disposición que tienen que guardar el centro de éste y el punto inicial del trazo, es decir, que después de la transformación se debe conservar la misma relación entre ellos para que la figura no pierda sus características.

Para esto, es necesario considerar dos casos generales:

a. Cuando el círculo a rotarse tiene como centro el origen relativo con respecto al cual se lleva a cabo la transformación.

b. Cuando el centro del círculo está alejado una determinada distancia del origen relativo.

En el primer caso, el algoritmo debe ser capaz para no modificar en absoluto la disposición de los parámetros del círculo, ya que un círculo rotado sobre su centro siempre

conserva las mismas características, mientras que si se presenta el segundo caso, se tendrán que seguir varias etapas para llevar a cabo la transformación: Primero se resguarda el valor del radio del círculo almacenándolo temporalmente, con lo que se procederá a la rotación del punto inicial del trazo considerándolo un punto semejante a los extremos de una recta. Después de esto se procede a estructurar el centro del círculo rotado asignándole a su coordenada X el mismo valor que la correspondiente del nuevo punto inicial, y a la Y la resta de la coordenada Y del punto inicial rotado menos el valor del radio previamente almacenado.

Con esto, el rotado de círculos se hará correctamente no importando la posición que tenga en el plano.

En lo que respecta a la rotación de rectas, ésta no presenta ningún problema, ya que se realiza transformando independientemente sus puntos extremos, como se ilustró anteriormente.

Así, el proceso que se sigue para llevar a cabo esta tarea es:

Primero se direcciona el inicio de la tabla de datos a transformarse. Teniendo esto, y por medio de la palabra de control correspondiente al punto que se tiene direccionado, se checa si es el último del conjunto. En caso afirmativo éste se considera como el extremo de una recta y como tal, se translada hacia el origen absoluto, se transforma y se desplaza a la región original del plano, para con esto terminar la transformación.

En caso negativo, se evalua, via la palabra de control, si se trata del extremo de una recta o del punto inicial de un circulo. Si es lo primero, este se transforma como tal y se accesa al siguiente punto, con lo que el proceso se repite. Si se trata de un circulo, entonces se accesa a las coordenadas de su punto central y se comparan con el origen relativo, de donde si resultan iguales no se modifican en absoluto los datos y se accesa al siguiente punto, donde el proceso se repite desde el inicio.

Si el centro del circulo no coincide con el origen, se obtiene el radio del circulo realizando la resta entre las coordenadas Y del punto inicial del trazo y el centro del circulo, para posteriormente transformar su punto inicial de manera semejante a la seguida con los extremos de las rectas, y finalmente obtener las coordenadas del nuevo centro via las siguientes expresiones:

$$X_{\text{centro}} = X_{\text{inicial rotado}}$$

$$Y_{\text{centro}} = Y_{\text{inicial rotado}} - \text{RADIO.}$$

con lo que se accesa al siguiente punto para repetir el proceso.

Para lograr la implementación de este algoritmo dentro del sistema graficador FAP-100, se tuvo que hacer frente a dos problemas graves: el primero es el relacionado a las multiplicaciones requeridas por las ecuaciones (2.40) y (2.41), ya que la microcomputadora huesped no es capaz de realizarlas, y el cual fue solventado via la subrutina CONV,

citada anteriormente.

El segundo, también presente en estas ecuaciones, relacionado con la obtención de las funciones seno y coseno del ángulo de rotación, pues de la misma forma, la máquina es incapaz de generar estos resultados por sí misma.

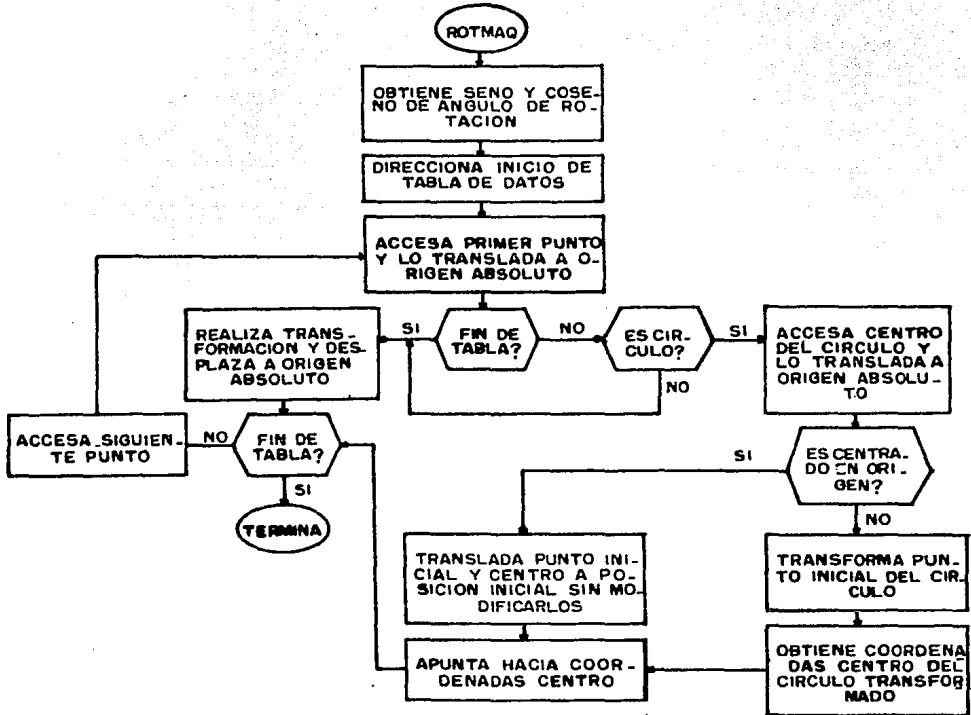
Este último problema se resolvió por medio de la implementación de un algoritmo recursivo con el cual, a partir de los valores  $\text{sen}(0)=0$  y  $\text{cos}(0)=1$ , obtiene el valor de estas funciones para cualquier ángulo. La subrutina SENCOS es la que lo engloba y un estudio detallado de éste se logra consultando en apéndice D de este trabajo, donde se expone completamente.

Cabe mencionar, que dada la naturaleza de éste, el rango de variación de estas funciones es de -1000 a 1000 en lugar de -1 a 1, por lo que para la obtención de resultados correctos, la división entre 1000 deberá ser llevada a cabo, tal y como se hace dentro de la subrutina OPERA, la cual se analiza adelante.

Tomando esto en cuenta, a continuación se muestra el diagrama a bloques de la subrutina ROTMAQ, que es la encargada de realizar la transformación rotación sobre una determinada gráfica.

Es necesario aclarar que cada vez que se realizan las acciones englobadas en esta subrutina, se tienen que realizar otras relacionadas con la interfaz máquina-usurio. Estas últimas son llevadas a cabo por ROTUSUA, y dado que la utilización de ésta se da siempre junto con el uso de ROTMAQ,

estas dos están incluidas dentro de una tercera, llamada ROT (ROTación), y que es la que en realidad se invoca cuando se requiere de la alternativa de transformación rotación.



Debido a que una explicación total de esta subrutina perdería sentido en muchos de sus aspectos, específicamente en lo referente a ROTUSU, dado que no se cuenta hasta el momento con detalles de la interfaz máquina-usuario, durante esta sección solo se citarán las características de los elementos que conforman ROTMAQ, y se dejarán los restantes



para un estudio profundo en el siguiente capítulo, teniéndose que relacionar en ese momento los detalles que se expongan con los que ahora se muestran.

En base a lo anterior, a continuación se listan las características principales de las subrutinas desarrolladas para poder contar dentro del sistema graficador con la alternativa de rotación. Cabe mencionar que para un estudio detallado de ellas, se tiene que consultar el apéndice A de este trabajo, en donde se encuentran los listados completos de las mismas.

\* NOMBRE: ROT.

FUNCION: Realiza la rotación de una gráfica base con respecto a un origen relativo. Esta es en sentido anti-horario y una cantidad de grados definida por el usuario, la cual puede variar entre 0 y 360 grados.

DIRECCION: 2314H.

SUBROUTINAS QUE UTILIZA: REQGRAD, ROTUSU, ROTMAQ.

DATOS ENTRADA: Dirección inicial de tabla de datos a transformarse en localidad DIR1.

DATOS SALIDA: Tabla de datos transformados a partir de dirección DIR1.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: REQGRAD Cestrechamente ligada a interfaz máquina-usuario, por lo que se estudia en el

siguiente capítulo).

DIRECCION: 2336H.

- \* NOMBRE: ROTUSU (estrechamente ligada a interfaz máquina-usuario, por lo que se estudia en el siguiente capítulo).

DIRECCION: 236DH.

- \* NOMBRE: ROTMAQ.

FUNCION: Lleva a cabo la rotación de una gráfica en sentido anti-horario con respecto a un origen relativo. Esto lo hace en base a un ángulo que varía entre 0 y 360 grados.

DIRECCION: 23AAH.

SUBROUTINAS QUE UTILIZA: CRLF, ESMEN, XY-ORG, SUMIYS, OBSECO, NUECON, LETEC.

DATOS ENTRADA: Angulo de rotación en localidad GRAD. Valor del seno del ángulo de rotación en localidad SEN. Valor del coseno en localidad COS. Coordenadas origen relativo en ORGX y ORGY. Dirección inicial tabla de datos en DIR1.

DATOS SALIDA: Tabla de datos transformada a partir de la dirección almacenada en DIR1.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

- \* NOMBRE: CRLF (Manda retorno de carro con alimentación de línea. Está ligada a la interfaz

máquina-usuario, por lo que su estudio se hace en el siguiente capítulo).

DIRECCION: C53AH.

- \* NOMBRE: ESMEN (Despliega mensajes en pantalla. Para su mejor entendimiento su estudio se hace en el siguiente capítulo).

DIRECCION: 01DCH.

- \* NOMBRE: NUECON (Almacena palabra de control en memoria. Se estudia en detalle en el siguiente capítulo).

DIRECCION: 1CE4H.

- \* NOMBRE: LETEC (Lee dato del teclado. Dada su estrecha relación con la interfaz máquina-usuario, su estudio se hace el capítulo siguiente).

DIRECCION: 0136H.

- \* NOMBRE: XY-ORG.

FUNCION: Resta a las coordenadas del punto en proceso, el valor de las correspondientes del origen relativo.

DIRECCION: 2547H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Coordenadas origen relativo en localidades ORGX y ORGY. Registro IY apuntando a la coordenada X del punto en proceso,

DATOS SALIDA: Coordenadas trasladadas a partir de la  
dirección almacenada en IY.

REGISTROS QUE MODIFICA: BC, DE, HL.

\* NOMBRE: XY+ORG.

FUNCION: Adiciona a las coordenadas del punto transformado  
las correspondientes del origen relativo.

DIRECCION: 256EH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Coordenadas origen relativo en localidades  
ORGX y ORGY. Registro IY apuntando a la  
coordenada X del punto transformado.

DATOS SALIDA: Coordenadas trasladadas del punto  
transformado a partir de la dirección  
almacenada en IY.

REGISTROS QUE MODIFICA: BC, DE, HL.

\* NOMBRE: RESIY5.

FUNCION: Resta cinco unidades al número almacenado en el  
registro IY.

DIRECCION: 259IH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a incrementarse en IY.

DATOS SALIDA: Dato incrementado en IY.

REGISTROS QUE MODIFICA: IY.

\* NOMBRE: SUMIY5 (citada anteriormente).

DIRECCION: 1CEDH.

\* NOMBRE: ROTCIR.

FUNCION: Realiza la rotación de un círculo tomando en cuenta si este es centrado en el origen o no.

DIRECCION: 24ACH.

SUBROUTINAS QUE UTILIZA: SUMIY5, OBSECO, NUECON.

DATOS ENTRADA: Valor del seno del Angulo de rotación en SEN. Valor del coseno en COS. Coordenadas trasladadas del centro del círculo hacia origen absoluto en ABSX y ABSY. Coordenadas origen absoluto en DIFX y DIFY. Registro IY apuntando a coordenada X del punto inicial.

DATOS SALIDA: Coordenadas rotadas del punto inicial y centro del círculo, sin trasladarse hacia posición inicial, a partir de dirección almacenada en IY.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: OBSECO.

FUNCION: Realiza las operaciones necesarias para obtener las coordenadas rotadas del punto en proceso en base a las ecuaciones

$$X' = X \cos \theta - Y \sin \theta$$

$$Y' = Y \cos \theta + X \sin \theta$$

Adicionalmente, almacena estos resultados en memoria.

DIRECCION: 2517H.

SUBROUTINAS QUE UTILIZA: YFUN, XFUN, NUEDAT.

DATOS ENTRADA: Valor del seno y coseno del ángulo de rotación almacenados en localidades SEN y COS respectivamente. Registro IY apuntando a la coordenada X del punto a transformarse.

DATOS SALIDA: Coordenadas X' y Y' almacenadas a partir de la dirección almacenada en IY.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

- \* NOMBRE: NUEDAT (almacena dato en memoria. Su estudio se realiza detalladamente en el siguiente capítulo).

DIRECCION: 1CD7H.

- \* NOMBRE: XFUN.

FUNCION: Accesa la coordenada X del punto en proceso y la multiplica ya sea por el seno o el coseno del ángulo de rotación, según sea el caso. Adicionalmente, este resultado lo divide entre 1000 por razones debidas al algoritmo utilizado para la obtención de sen y cos.

DIRECCION: 25BAH.

SUBROUTINAS QUE UTILIZA: OPERA.

DATOS ENTRADA: Valor del seno o coseno del ángulo de rotación en registro DE. Registro IY apuntando a coordenada X del punto en

proceso.

DATOS SALIDA: Resultado de la multiplicación entre X y SEN o COS, dividido entre 1000 en HL.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: YFUN.

FUNCION: Accesa la coordenada Y del punto en proceso y lleva a cabo la multiplicación de ésta con la función seno o coseno del ángulo de rotación, según sea el caso. Adicionalmente, divide este resultado entre 1000 por razones debidas al algoritmo utilizado para la obtención de las funciones SEN y COS.

DIRECCION: 25C4H.

SUBROUTINAS QUE UTILIZA: OPERA.

DATOS ENTRADA: Valor de la función seno o coseno del ángulo de rotación en registro DE. Registro IY apuntando a la coordenada X del punto en proceso.

DATOS SALIDA: Resultado de la multiplicación entre Y y el seno o cos dividido entre 1000 en HL.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: OPERA.

FUNCION: Realiza la multiplicación entre la coordenada X o Y, según sea el caso, del punto en proceso por la función seno o coseno del ángulo de rotación.

Además, este resultado lo divide entre 1000.

DIRECCION: 25CEH.

SUBROUTINAS QUE UTILIZA: MULTI16, PREDIV, ANTDIV.

DATOS ENTRADA: Coordinada en uso del punto en proceso almacenada en registro BC. Función por la cual, se llevará a cabo la multiplicación en DE.

DATOS SALIDA: Resultado de la operación dividido entre 1000 en registro HL.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: MULTI16.

FUNCION: Engloba al algoritmo de corrimiento-adición secuencial (ver apéndice E), el cual multiplica un número de 12 por otro de igual longitud, produciendo un tercero de 24 bits.

DIRECCION: 25DEH.

SUBROUTINAS QUE UTILIZA: COMP2.

DATOS ENTRADA: Multiplicando en registro BC. Multiplicador en registro DE.

DATOS SALIDA: Producto de la multiplicación repartido en registros HL y DE.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: PREDIV.

FUNCION: Realiza los ajustes necesarios para que el resultado de la multiplicación pueda ser utilizado



por la subrutina DIV, para llevar a cabo la división entre 1000 antes citada.

DIRECCION: 2641H.

SUBROUTINAS QUE UTILIZA: COMP16.

DATOS ENTRADA: Producto de la multiplicación en registros HL y DE.

DATOS SALIDA: Dato compatible con DIV en registros HL Y DE.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: COMP16.

FUNCION: Obtiene el complemento a dos del número de 24 bits, producto de la multiplicación, si este es negativo.

DIRECCION: 262BH.

SUBROUTINAS QUE UTILIZA: COMP2.

DATOS ENTRADA: Número de 24 bits en HL y DE.

DATOS SALIDA: Número complemento a 2 en registros HL y DE.

REGISTROS QUE MODIFICA: A, DE, HL.

Para consultar acerca de la función específica, así como la dirección de cada una de las localidades de servicio aquí citadas, se debe acceder el apéndice B de este trabajo, donde se exponen.

## CAPITULO TRES

### Interfaz máquina-usuario.

De entre los elementos que conforman a un sistema graficador basado en computadora, quizá el que juega un papel más importante en lo que se refiere a ofrecer comodidad al usuario, sea aquel que se encarga de interactuar con éste precisamente. Es decir, aquel que lleva a cabo la tarea tanto de recibir información y transportarla a la computadora, como la de suministrar ésta al usuario, pudiendo ser la primera datos o comandos y la segunda alternativas disponibles o indicaciones para llevar a cabo tareas necesarias para el funcionamiento del sistema.

Con lo anterior, no se niega el tremendo potencial que poseen los elementos restantes del sistema, pero es claro que este último está más en contacto con el usuario que los otros, por lo que en él recae más la responsabilidad de hacer atractivo, tanto en apariencia como en funcionamiento, al sistema.

Este elemento, conocido generalmente como interfaz

máquina-usuario, es por lo tanto, el medio por el que se vale el usuario para acceder y manejar todos los recursos contenidos en el sistema, y no importa cual sea su estructura, ésta debe ser tal que su uso presente el máximo de facilidades para el usuario.

Aunque existen muy diversas presentaciones de esta interfaz, en el presente capítulo se centrará la discusión alrededor de la desarrollada específicamente para el sistema graficador FAP-100. En éste, se expondrán tanto sus características detalladas, y que generalmente no son accesibles a vista del usuario, como la apariencia que presenta a éste.

### 3.1 Características generales.

Dentro del sistema graficador FAP-100, la interfaz máquina-usuario está implementada fundamentalmente en base a programas almacenados en memoria, los cuales, para tener un lazo físico de comunicación con el usuario, ya que por sí mismos no lo poseen, hacen uso de dos elementos periféricos de la microcomputadora huésped: el teclado, como dispositivo de entrada y la terminal de video como salida. Estos programas son los que conforman el cuarto tipo de la programación desarrollada para el sistema: los programas de comunicación.

Basicamente, la forma de interacción con que se

relacionan con el mundo externo, son mensajes desplegados en el dispositivo de salida, por medio de los cuales el usuario reconoce tanto alternativas contenidas dentro del sistema, como indicativos para el suministro de información a éste. La forma en que el usuario corresponde a estas acciones, es introduciendo tanto datos como comandos por medio del teclado de la computadora.

Quizá la característica más importante de esta interfaz es la referente al sistema de numeración que utiliza, ya que no se trata del manejado por la microcomputadora (hexadecimal), sino que, tratando de proporcionar la mayor comodidad posible, es el decimal. Con esto, los programas de comunicación adquieren la responsabilidad de realizar la conversión de un sistema a otro.

Adicionalmente, y también buscando mayor facilidad para su uso, la interfaz permite al usuario el manejo de números negativos, los cuales en realidad no son disponibles en el área de graficado, y por lo cual se hace necesario que el proceso de conversión tome en cuenta esta característica. Por otra parte, no se modifica la limitación intrínseca de la máquina de manejar solamente números enteros, por lo que aún dentro del sistema decimal, se tiene que trabajar con estos.

El rango de valores que maneja el sistema es de -9999 a +9999, aunque en realidad la máquina solo trabaje dentro del rango de 0 a 4096. Así, es claro que la interfaz debe de realizar un proceso de escalamiento para ajustar un rango dentro del otro, con lo que se logra una compatibilidad

completa entre la información manejada por el usuario y la manejada por la microcomputadora.

Tomando como base este rango de operación, el área de trabajo disponible al usuario, adquiere valores límites distintos a los manejados por la máquina, los cuales se muestran en la figura 3.1 y funcionan como una ventana del plano cartesiano completo en la misma forma en que lo hacen los límites reales (entre paréntesis en la misma figura), y que se cito en el capítulo anterior.

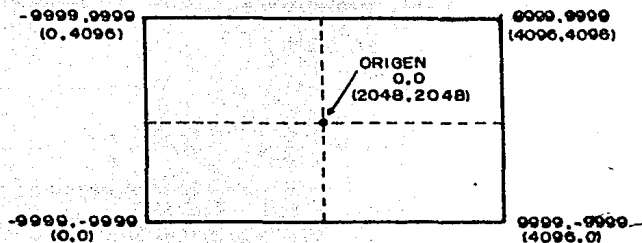


FIGURA 3.1.

Límites para el usuario del área de graficado.

La técnica seguida para acceder a los recursos contenidos en el sistema, es en base a menús de selección, es decir a listas de las alternativas disponibles, las cuales son elegidas en base a un código de identificación que poseen cada una de ellas.

Aunque para cada opción desplegada existe un programa dentro de la interfaz encargado de poner en contacto a ésta con el usuario, existen algunos en los que invariablemente,

todos y cada uno de los pertenecientes a las distintas alternativas, se basan. Estos son denotados como "Programas Básicos de Comunicación".

A continuación se hace un estudio detallado de éstos, para posteriormente abordar las diferentes alternativas disponibles al usuario, tanto en sus detalles generales como en los particulares.

### 3.2 Programas básicos de comunicación.

Dado que la estructura de la interfaz máquina-usuario implementada para el sistema graficador FAP-100 es por demás abstracta, ya que es base a programas, para su interrelación con el usuario se tuvo que hacer uso de un dispositivo físico, el cual fuera manipulable por éste, y además presentara la característica de ser compatible tanto con el sistema como con la microcomputadora misma.

Este puente físico, está compuesto en realidad por dos dispositivos: uno de entrada, el teclado alfanumérico, y otro de salida, la pantalla de video, los cuales pueden ser manejados en base a programas almacenados en memoria, hecho del cual se partió para su inclusión dentro del sistema.

Para el uso de estos dos elementos, se utilizaron dos programas o subrutinas básicas, que permiten: uno adquirir información desde el teclado, y otro transmitir ésta hacia la terminal de video. Cabe mencionar que estos programas, así

como ESMEN (citado adelante en esta sección), no fueron en realidad desarrollados específicamente para el sistema graficador, sino que forman parte del sistema operativo de la microcomputadora FAP-100, la cual utiliza los mismos periféricos para comunicarse con el exterior, por lo que su programación (en este aspecto), pudo ser utilizada para el fin perseguido sin modificación alguna.

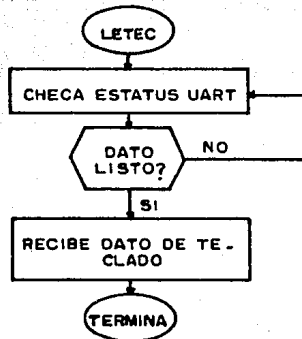
La característica principal de estos dos subprogramas, es que solo se limitan a transportar, ya sea del exterior a la computadora o viceversa, un dato de 8 bits (byte), sin importar la función que este dato realice. Es decir, no importa si forma parte de un valor numérico, o de un mensaje o incluso ser solamente un caracter de control, los programas realizan la tarea de comunicación.

Es por lo anterior, que en la construcción de los diferentes programas de comunicación, la utilización de estas subrutinas básicas se presentó repetidamente, ya que gracias a la versatilidad que ofrece su construcción, el fin que se les dio en cada uno de ellos fue distinto.

El programa que transmite la información desde el teclado hacia la microcomputadora es llamado LETEC (LEe TEClado), y tiene como función la de checar ciclicamente la bandera del UART (Ver capítulo 0, comunicación serie) que indica que se tiene un dato listo hasta que ésta es activa, después de lo cual introduce la información por el puerto de entrada previamente activado (y que puede ser conectado a diferentes tipos de dispositivos), para finalmente

almacenarla en el acumulador.

A continuación se muestran tanto el el diagrama de flujo como las características principales de esta subrutina, pudiendose estudiar el listado completo de ella en el apéndice A de este trabajo.



\* NOMBRE: LETEC.

FUNCION: Introduce un byte desde el teclado alfanumérico.

DIRECCION: 0136H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Dato transmitido desde teclado en A.

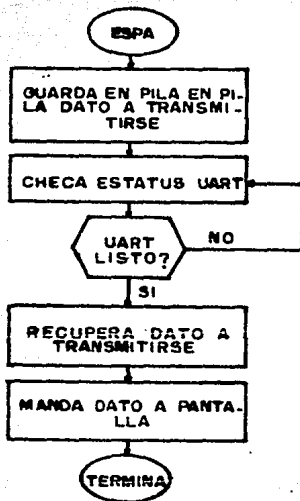
REGISTROS QUE MODIFICA: A.

Por otro lado, la subrutina encargada de transferir información hacia la terminal de video recibe el nombre de ESPA (EScribe en PANTalla). Esta, al igual que LETEC, trasmite un byte cada vez que se invoca, y la tarea la



realiza así: Recibe el dato a transmitirse y pregunta si el UART está listo para recibir la información. Si no está listo repite el proceso hasta que la respuesta es afirmativa. En ese momento manda el contenido del acumulador por el puerto conectado a la terminal con lo que su función termina.

A continuación se muestra el diagrama a bloques de esta subrutina, para inmediatamente listar las características principales de ella. Al igual que toda la programación, para un estudio detallado de ESPA, es necesario consultar el apéndice A de este trabajo.



\* NOMBRE: ESPA.

FUNCION: Transmite un dato de 8 bits hacia la terminal de video.

DIRECCION: 013FH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a transmitirse en acumulador.

DATOS SALIDA: Dato en puerto conectado a terminal de video.

REGISTROS QUE MODIFICA: Ninguno.

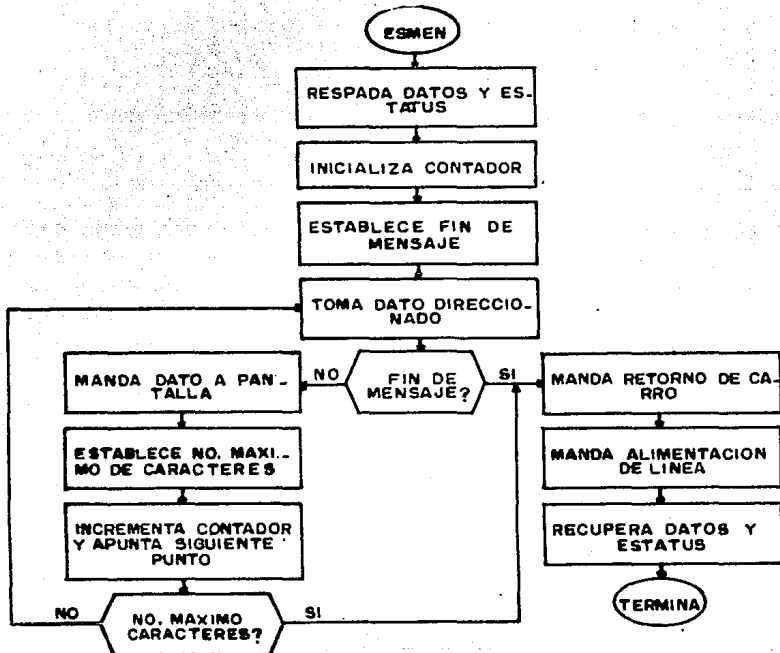
Además de estos dos programas, existen otros que aunque son propiamente una extensión de ESPA, conviene citarlos en esta sección, pues el uso que se les da, es exactamente igual que a los primeros.

La subrutina ESMEN (EScribe MENsaje), tiene como función la de desplegar un mensaje en la pantalla de video de hasta 80 caracteres como máximo, después de lo cual trasmite un comando de retorno de carro, con lo que posiciona al cursor en la columna inicial del renglon, y otro de alimentación de línea, con lo que se posiciona en el renglon siguiente. Para transmitir un mensaje de longitud menor de los 80 caracteres, este debe finalizar con el código de retorno de carro (ODH), con lo que el programa reconoce el fin de la información a transmitirse.

Para su funcionamiento, la subrutina hace uso de ESPA cada vez que transmite un dato a la pantalla, y adicionalmente utiliza dos registros indicadores, uno contador de caracteres transmitidos, y otro como registro comparador para reconocimiento de fin de mensaje.

A continuación se muestra el diagrama a bloques de esta subrutina, en donde se observa el uso que se hace tanto de

ESPA como de los registros indicadores.



Las características principales de este subprograma son las que a continuación se exponen, pudiendose consultar el listado completo en el apéndice A de este trabajo.

\* NOMBRE: ESMEN.

FUNCION: Transmite un mensaje de hasta 80 caracteres hacia la terminal de video, el cual es finalizado con la transmisión de los códigos de retorno de carro y

de alimentación de línea.

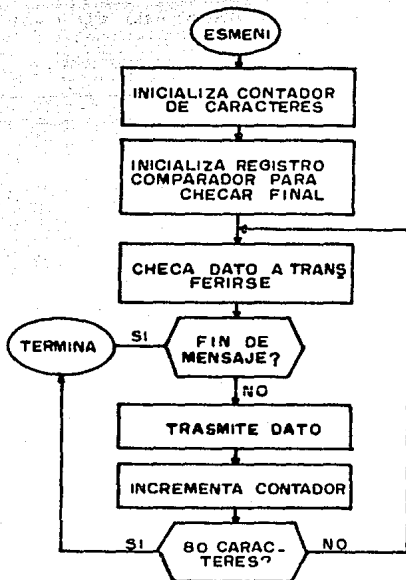
DIRECCION: 01DCH.

SUBROUTINAS QUE UTILIZA: ESPA.

DATOS ENTRADA: Dirección inicial de mensaje en HL.

DATOS SALIDA: Mensaje por puerto conectado a terminal de video.

REGISTROS QUE MODIFICA: HL.



La subrutina ESMENI, realiza basicamente la misma tarea que ESMEN, con la única diferencia de que ésta no transmite ni el código de retorno de carro ni el de alimentación de línea al final del mensaje. Al igual que la anterior, el

máximo de caracteres que puede transmitir es 80 y reconoce el fin de uno menor a este número con el código de retorno de carro.

A continuación se muestran las características principales de esta subrutina, mientras que en la ilustración anterior se puede observar el diagrama a bloques de la misma.

\* NOMBRE: ESMEN1.

FUNCION: Transmite un mensaje de hasta 80 columnas hacia la terminal de video.

DIRECCION: 18BEH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Direccion inicial de mensaje en HL.

DATOS SALIDA: Mensaje por puerto conectado a terminal de video.

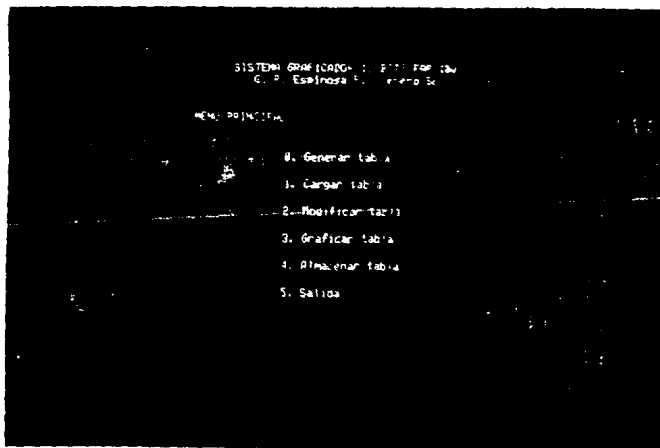
REGISTROS QUE MODIFICA: HL.

Al igual que con la anteriores, el listado completo de esta subrutina se puede consultar en el apéndice A al final de este trabajo.

### 3.3 Alternativas disponibles.

La finalidad principal de la interfaz máquina-usuario es la de ofrecer a este último la facilidad de acceder todas las diferentes funciones que el sistema puede realizar. Esto es,

debe de dar al usuario la capacidad de poder manejar los recursos disponibles tanto para la construcción de una gráfica como para poder manipular a ésta.



FOTOGRAFIA 3.1.

Menú principal sistema graficador FAP-100.

Esta tarea tiene que ser llevada a cabo de tal forma, que para su manejo se presente la menor dificultad posible, además de poseer una apariencia que no produzca rechazo en el usuario, es decir, que este diseñada para que pueda actuar "amigablemente" durante su uso.

El método utilizado para la implementación de esta interfaz dentro del sistema graficador FAP-100, es en base a menús de selección. Esto es, en base a listas de las

diferentes alternativas a las que se puede tener acceso durante una determinada etapa en el manejo del sistema. De estos, se cuenta con dos diferentes:

- \* Menú principal, y
- \* Menú de modificación.

El primero es con el que el sistema inicia su funcionamiento, y el cual presenta el formato mostrado en la fotografía 3.1.

En el, se engloban tanto la presentación del sistema, como las funciones primarias que él posee, entendiéndose éstas como las que se tienen que ejecutar antes de manipular o transformar una determinada gráfica.

En general, la función a la que se accesa por medio de cada una de estas alternativas es:

0. Generar tabla. Se refiere a la habilitación que se le da al usuario para poder construir una tabla de datos la cual podrá ser presentada posteriormente gráficamente.
1. Cargar tabla. Presupone la existencia de una tabla de datos construida con anterioridad y almacenada en el dispositivo destinado a este fin. Con esto ofrece la facilidad de poder recuperarla, realizando la tarea de cargar los datos en la memoria de la máquina para su posible manipulación o presentación

gráfica.

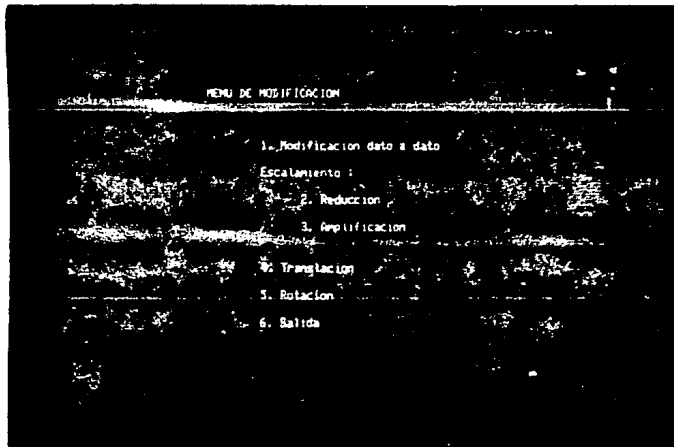
2. Modificar tabla. Presupone la existencia de una tabla de datos alojada en la memoria de la máquina y con esto da acceso al segundo menú de que consta la interfaz: el menú de modificaciones.
3. Graficar Tabla. Ofrece al usuario la posibilidad de desplegar en forma gráfica la información almacenada en memoria. Presupone la correcta conexión hacia el graficador analógico via la interfaz D/A.
4. Almacenar tabla. Da al usuario la posibilidad de almacenar en cinta, la información generada en forma de tabla de datos y almacenada en memoria.
5. Salida. Por medio de la selección de esta alternativa, se deshabilita al sistema graficador y se accesa al sistema operativo de la máquina.

En lo que respecta al menú de modificación, su función es la ofrecer al usuario el posible acceso a las diferentes alternativas de que consta el sistema para poder manipular o transformar una determinada gráfica.

Fundamentalmente se compone de los distintos tipos de transformaciones implementados y los cuales fueron citados en el capítulo anterior, adicionando la posible elección de modificar dato a dato la tabla base.



El formato que presenta a la vista del usuario es el que se muestra en la fotografia 3.2, en donde se puede observar las distintas elecciones que se pueden realizar y que se comentaban en el parrafo anterior.



FOTOGRAFIA 3.2.

Menú de modificación sistema graficador FAP-100.

En general, la función a la que se accesa por medio de cada una de las alternativas listadas es:

1. Modificación dato a dato. Hace capaz al usuario de poder modificar desde uno solo a todos los datos que conforman una determinada tabla de datos. Esto lo realiza accedendo uno a uno cada valor.

2. Reducción. Habilita al usuario para poder reducir el tamaño de una determinada gráfica, requiriendo unicamente los factores de escalamiento para cada uno de los ejes.
3. Amplificación. Ofrece la posibilidad de hacer más grande el tamaño de una gráfica base requiriendo unicamente los factores de escalamiento de cada uno de los ejes.
4. Translación. Por medio de un factor para cada uno de los ejes, habilita al usuario para poder trasladar una gráfica base a una región diferente del plano sin modificar su tamaño.
5. Rotación. Da al usuario la posibilidad de rotar una figura en base al ángulo que este proporciona.
6. Salida. Por medio de esta alternativa se regresa al menú principal del sistema.

A continuación se hace un análisis detallado de cada una de estas alternativas, mostrando tanto sus funcionamiento como las subrutinas implementadas para su inclusión dentro del sistema graficador.

### 3.4 Características particulares de alternativas.

Para la implementación de las alternativas que se ofrecen al usuario en los listados citados en la sección

anterior, se desarrollaron las subrutinas necesarias para lograr la ejecución especificada en cada una de ellas.

La estructura de éstas fue tal, que para la realización de una tarea específica, solo se tiene que invocar a un programa base, el cual se encarga de dirigir a los demás subprogramas para la ejecución de la función requerida, correspondiéndole a cada opción uno de éstos.

Para la invocación de estos programas base, y tratando de brindar la mayor facilidad posible al usuario, el sistema graficador hace uso de un programa principal, el cual se encarga de acceder a cada uno de ellos en función de la alternativa elegida vía los periféricos disponibles para este fin.

La presente sección se dedica al estudio en primer lugar, del programa principal de que consta el sistema, para posteriormente citar las características particulares de cada uno de los programas base que corresponden a las alternativas disponibles, exponiendo además, las subrutinas de que hacen uso cada uno de éstos.

#### 3.4.1 Programa principal.

La función principal de este programa es la de introducir al usuario a la utilización del sistema graficador FAP-100. Es, por lo tanto, el primer programa que se ejecuta cuando se invoca la utilización de este último.

En general, su funcionamiento se puede dividir en dos partes:

1. Hacer del conocimiento del usuario, las diferentes alternativas a las que tiene acceso durante el uso del sistema graficador, y
2. Recibir la elección que de ellas haga para acceder al programa base correspondiente y con ésto llevar a cabo la tarea requerida.

Para llevar a cabo esta tarea, este programa (llamado UNO), se basa fundamentalmente en otras dos subrutinas, MENU para la primera parte, y LETEC para la segunda.

La primera de éstas, es la que se encarga de desplegar en pantalla el menú principal de que consta el sistema, llevando a cabo las siguientes acciones: Primero por medio de ESPA ,transmite el código requerido para borrar la pantalla de la terminal, para inmediatamente después, via la subrutina ESMEN1, desplegar los mensajes correspondientes para cada una de las alternativas disponibles, los cuales se encuentran almacenados en memoria.

Hecho esto, UNO hace uso de la subrutina LETEC, por medio de la cual recibe el código de la opción elegida por el usuario, con lo cual realiza una comparación entre éste y los especificados para cada una de las funciones del sistema, invocando al programa base de la alternativa en donde el resultado de la comparación indique igualdad entre los códigos.

Para llevar a cabo la comparación, se hace uso de la tabla mostrada en la figura 3.2, en donde se expone tanto el código que corresponde a cada una de las alternativas, así como la función que se realiza y el nombre del programa base invocado.

código (hex)	programa base	función
30	GENTA	Generar tabla
31	ACCEA	Cargar tabla
32	MENMOD	Accesa menú de modificación
33	GRAF12	Grafica tabla
34	TRACIN	Almacena tabla
35		Sale a sistema operativo

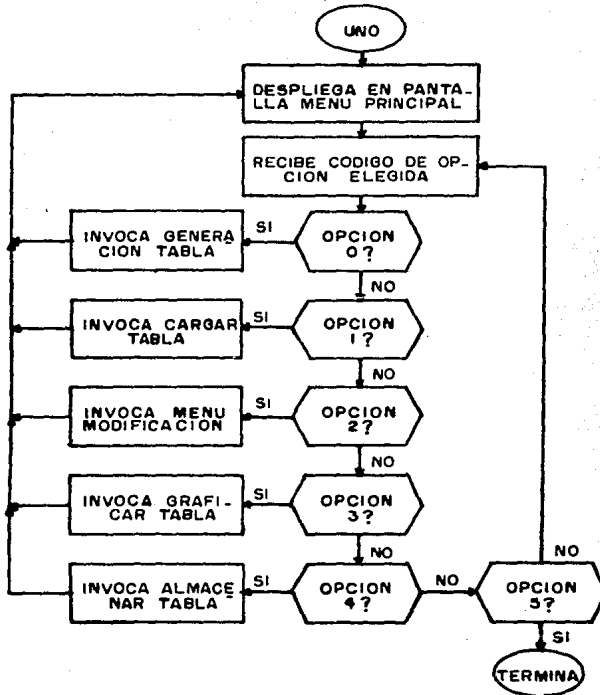
FIGURA 3.2.

Tabla de comparación de programa principal.

Con esto, el diagrama a bloques de este programa es el que se muestra en la siguiente ilustración en donde resalta la facilidad que se goza al seguir la técnica de programación en base a subrutinas, ya que a partir de este programa relativamente corto, se accesa a todo el conjunto de subrutinas de las que está formado el sistema graficador.

Las características principales de las subrutinas de las que hace uso, en parte son las que se listaron en la sección 3.2, esto es, para ESPA, LETEC y ESMENI. Las de las restantes, es decir las de los programas base que invoca, se dejan para secciones posteriores, en donde se abordarán

detalladamente cada una de éstas.



El listado de este programa principal se puede consultar en el apéndice A de este trabajo, mientras que las características principales de el, son las que se exponen a continuación:

\* NOMBRE: UNO.

FUNCION: Despliega las alternativas a las que se tiene acceso durante el uso del sistema graficador y

recibe el código de la elección hecha por el usuario, con lo que invoca al programa base de la alternativa requerida.

DIRECCION: 26BOH.

SUBROUTINAS QUE UTILIZA: ESPA, LETEC, ESMEN1, GENTA, ACCETA, MENMOD, GRAF12, TRACIN.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Ninguno.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY, IX.

a

A continuación, y con fines de una mejor explicación, se exponen las características de la subrutina MENMOD, para después de esto analizar las restantes.

#### 3.4.2 Programa de modificación.

Cuando desde el programa principal se accesa a la alternativa etiquetada con el número 2, que corresponde a MENMOD en la tabla de la figura 3.2, en realidad la acción que se lleva a cabo no es invocar un programa base correspondiente a alguna alternativa dentro del sistema graficador, sino que se ejecuta un programa cuyas características son muy similares a las del primero, y el cual recibe el nombre de programa de modificación.

La función de éste, es la de ofrecer al usuario la oportunidad de hacer uso de otra serie de programas base, los

cuales son los encargados de manejar las subrutinas implementadas para llevar a cabo funciones de transformación, es decir, amplificación, reducción, rotación, etc.

Debido a la estructura que posee, que es exactamente igual a la del principal, este programa divide su funcionamiento en dos partes: la de desplegar en pantalla la información necesaria para que el usuario tenga conocimiento de las opciones que se le ofrecen, y la de recibir el código de identificación de la elección hecha por este último, y con esto invocar el programa base requerido.

código (hex)	programa base	función
31	MODADA	Realiza modificación dato a dato.
32	RDUC	Reduce el tamaño de una gráfica.
33	AMP	Amplifica el tamaño de una gráfica.
34	DES	Desplaza una gráfica en el plano.
35	ROT	Rota una gráfica.
36		Regresa a menú principal.

FIGURA 3.3.

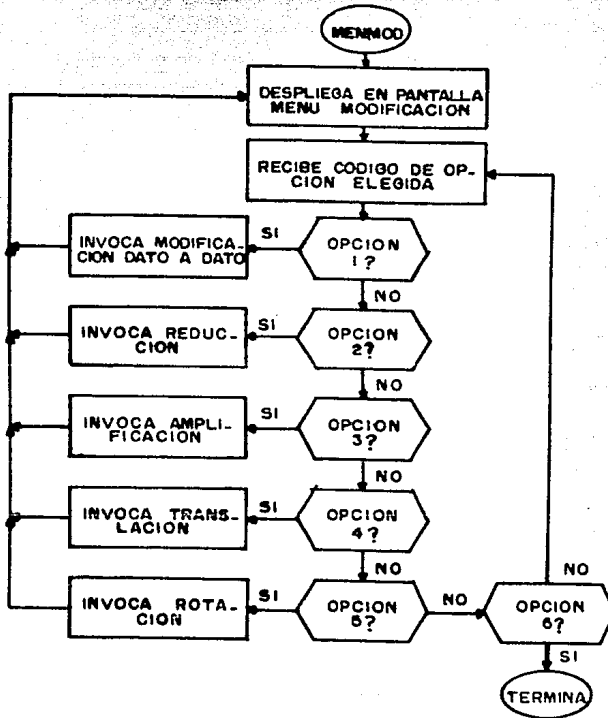
Tabla de comparación para menú de modificación.

La primera la realiza en base a la subrutina ESMEN1, por medio de la cual exhibe los mensajes necesarios para estructurar en pantalla el menú de modificación, que se muestra en la fotografía 3.2. Para la segunda, por medio de LETEC recibe el código de la opción elegida y realiza un



proceso de comparación con los códigos de los programas base disponibles, invocando aquel con el que encuentre igualdad. Esta última función la hace en base a la tabla de comparación que se muestra en la figura 3.3.

Tomando lo anterior en cuenta, el diagrama a bloques de este programa es el que a continuación se presenta, en donde se observa la estructura igual a la del principal, con la diferencia de que el de modificación se comporta como una subrutina de este último.



Las características principales de las subrutinas que utiliza, fueron listadas en la sección 3.2 para LETEC, ESPA Y ESMEN1. Las de los programas base que se invocan se dejan para una posterior explicación, cuando se abordarán detalladamente. A continuación entonces, solo se listan las que corresponden al programa de modificación, teniéndose se consultar el apéndice A de este trabajo para un estudio profundo del mismo, pues es ahí en donde se muestran las características particulares del mismo.

\* NOMBRE: MENMOD.

FUNCION: Da al usuario la oportunidad de acceder los programas base que manejan a las subrutinas de transformación.

DIRECCION: 1E3BH.

SUBROUTINAS QUE UTILIZA: ESPA, ESMEN1, LETEC, MODADA, RDUC, AMP, DES, ROT, POSC.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Ninguno.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY, IX.

\* NOMBRE: POSC (explicada anteriormente).

DIRECCION: C479H.

### 3.4.3 Generación de tabla de datos.

El objetivo de esta alternativa, es la de habilitar al sistema para que pueda ofrecer al usuario la posibilidad de introducir las coordenadas de los puntos que posteriormente serán graficados.

El programa base que corresponde a ésta recibe el nombre de GENTA (GENera TABla), y tiene como función principal la de construir con la información suministrada desde el exterior, la tabla de datos a partir de la cual, los elementos restantes del sistema basarán su funcionamiento (ver figura 2.25 ). Adicionalmente, realiza las tareas de inicializar el valor de las coordenadas del origen relativo, recibir la dirección inicial a partir de la que se estructurará la tabla mencionada y escalar los valores de esta última de manera que los datos en memoria sean compatibles con la máquina misma.

El orden en que lleva a cabo estas funciones es el siguiente:

Primero, en las localidades destinadas a almacenar las coordenadas del origen relativo (ORGUSX,ORGUSY), guarda los valores que el sistema da de entrada para éstas, es decir  $X= Y= 0800H$ . Cabe hacer notar, que el uso de éstas localidades en lugar de las que se han venido utilizando en explicaciones anteriores (ORGX,ORGY), se debe a que en esta etapa de funcionamiento, se genera una tabla denominada como "de usuario", la cual debe de poseer su propio origen con fines

que se aclararán cuando se estudien las alternativas de transformación.

Inmediato a esto, y por medio de la subrutina ESMEN1, exhibe el letrero

DIRECCION INICIAL: OOOO

con lo que permite al usuario introducir la dirección a partir de la que se construirá la tabla de datos.

Después de recibir esta información, borra la pantalla por medio de ESPA, y presenta la plantilla que sirve como guía para el suministro de las coordenadas de los puntos a graficarse. Esta plantilla se muestra en la fotografía 3.3, donde se puede observar que se compone de 7 columnas, la primera para indicar el número de renglon que se está generando y cuya utilización se da en el programa de modificación. Las tres siguientes permiten la introducción de las coordenadas (x,y) de un punto con su correspondiente palabra de control, al igual que las tres últimas.

Para la recepción de estos datos, el programa genera por si mismo el número de renglon (empezando en el uno), e inmediatamente después se posiciona en la columna marcada con COORX, en donde presenta un patron de cuatro ceros y recibe el valor de esta. Ejecuta la misma función para la columna marcada con COORY y hecho ésto accesa la primer columna marcada con la palabra CONTROL.

En esta etapa, el programa es capaz de recibir solamente los códigos válidos para esta (ver figura 2.2), o el código

que indica el fin de tabla para el usuario, CTRL X, con lo que finaliza esta tarea. Si la información recibida no corresponde a este último código, entonces repite el proceso para las tres columnas restantes del renglon en uso, realizando el reconocimiento para identificar un posible fin de tabla al final de este. En caso negativo accesa el siguiente renglon incrementando y desplegando automáticamente el número del anterior, y repitiendo el mismo proceso.

Cada vez que el programa accesa un nuevo renglon, checa si el número de este es mayor que el último que se puede desplegar en la pantalla de la terminal de video (24). En caso negativo prosigue con las tareas encomendadas sin contratiempo alguno, pero si esta condición se presenta, entonces se genera un corrimiento hacia arriba de la información (scrolling) para generar el espacio necesario por el nuevo renglon y se presenta nuevamente la plantilla guía, perdida a causa de esta acción. Con esto, el usuario nunca pierde la información acerca de la columna a la que accesa.

Para finalizar con la construcción de la tabla de datos, el programa reconoce si la palabra de control suministrada por el usuario es la correspondiente a CTRL X, como se cito anteriormente, con lo que da por terminada la generación de la "TABLA DE USUARIO". En este momento, realiza la tarea de calcular ciertos parámetros pertenecientes a ésta, los cuales son utilizados por los programas de comunicación con cinta, como se verá en las secciones correspondientes.

Estos parámetros son: la dirección final que ocupa el

último dato de la tabla y la longitud del buffer. Estos son almacenados en los 6 bytes que anteceden al primer dato de la misma, tomando en cuenta que la dirección inicial también ocupa parte de este espacio. La forma en que son almacenados se muestra en la figura 3.4.

coordenada X origen usuario (2 bytes)
coordenada Y origen usuario (2 bytes)
dirección fi- nal archivo (2 bytes)
longitud archivo (2 bytes)
dirección inicial (2 bytes)
primer byte de archivo

FIGURA 3.4.

Almacenamiento de parámetros tabla usuario.

Ma que calculó esta información y la almacenó debidamente, entonces la despliega en pantalla para que el usuario tenga conocimiento de la misma. El formato en que la presenta se puede observar en la parte inferior de la fotografía 3.3, donde adicionalmente se puede leer el letrero

"NORM."

REF	COORD X	COORD Y	CONTROL	COORD X	COORD Y	CONTROL
0001	0000	9999	0	9999	0000	0
0002	1234	5678	1	8765	4321	1
0003	5478	2345	1	7650	1021	1
0004	4397	1232	0	2320	9891	0
0005	4332	9024	2	8862	2112	2
0006	4376	8624	0	4323	8838	0
0007	9787	8047	0	8217	8887	0
0008	3337	8041	0	8878	4524	0
0009	8888	7874	0	4325	8888	0
0010	8545	8739	0	8254	8321	0

DIRECCION FINAL: 0004  
 LONGITUD BUFFER: 0004  
 NORM.  
 LISTO

## FOTOGRAFIA 3.3.

Generación de tabla de datos.

Este último se genera como una indicación por parte del sistema, para que el usuario reconozca la siguiente tarea que se está ejecutando, y que es la de escalar o normalizar la tabla que generó para que los datos suministrados sean compatibles con los elementos restantes del sistema, fundamentalmente con la interfaz D/A. Esto es, construye lo que se denomina como "TABLA DE MAQUINA".

Para llevar a cabo la tarea de escalar la tabla de usuario, GENTA hace uso de la subrutina NORM. Esta es la encargada de tomar una a una las coordenadas de los puntos y

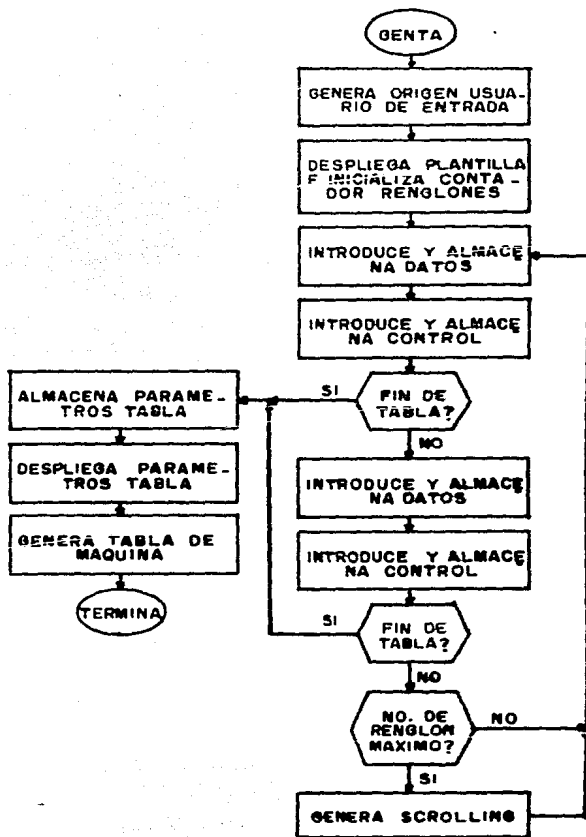
manipularlas de manera que el resultado numérico que genere sea proporcional al original pero dentro del rango en que trabaja la máquina (0-4096). Esta tarea la lleva a cabo por medio de las subrutinas DIV y +-48.

La tabla de máquina que genera la subrutina NORM queda almacenada en memoria inmediatamente después de la última localidad utilizada por la de usuario, y es necesario aclarar que las palabras de control pertenecientes a esta última, son transferidas a la primera sin sufrir modificación alguna. Además, esta subrutina tiene encomendada la función de copiar el valor corriente del origen relativo de usuario hacia las localidades que almacenan el de máquina (ORGX, ORGY).

La última función que realiza GENTA es la de informar al usuario que todas las acciones que tiene que realizar están concluidas. Esto lo lleva a cabo desplegando por medio de ESMEN el letrero "LISTO", como se puede observar también en la fotografía 3.3, después de lo que espera el comando "retorno de carro" para regresar al menú principal. Tomando lo anterior en cuenta, el diagrama a bloques de esta subrutina es el que se muestra en la siguiente ilustración.

Para la implementación de las funciones antes citadas, se desarrollaron las subrutinas necesarias, de las cuales las características principales de cada una de ellas son las que a continuación se exponen, pudiendose consultar el listado completo de las mismas en el apéndice A de este trabajo.





\* NOMBRE: GENTA.

FUNCION: Permite al usuario el poder introducir en formato decimal, las coordenadas de los puntos que conformarán la tabla de datos a partir de la cual el sistema graficador funcionará.

DIRECCION: 18D7H.

SUBROUTINAS QUE UTILIZA: ESPA, ESMEN1, LLENA, INDIR, GUADIR,

PLANT, BIASC, DATO, CONTRO, CHECA,  
 CRLF, INCR, INCREN, POSC, HEXA  
 DESP, ESMEN, NORM, LETEC.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Contrucción tabla usuario y de máquina.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: ESPA (citada anteriormente).

DIRECCION: 013FH.

\* NOMBRE: ESMEN1 (citada anteriormente).

DIRECCION: 18BEH.

\* NOMBRE: ESMEN (citada anteriormente).

DIRECCION: 01DCH.

\* NOMBRE: LETEC (citada anteriormente).

DIRECCION: 0136H.

\* NOMBRE: LLENA.

FUNCION: Inicializa el área reservada para la recepción de  
 un dato con cero y signo positivo.

DIRECCION: 19C4H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Area reservada para dato cargada con ceros,  
 a excepción del primer byte que se carga con

el código ASCII para espacio (20H).

REGISTROS QUE MODIFICA: IY.

\* NOMBRE: INDIR.

FUNCION: Recibe del teclado los códigos ASCII de un número hexadecimal de cuatro dígitos y los almacena en el área reservada para este fin.

DIRECCION: 19DDH.

SUBROUTINAS QUE UTILIZA: ESMEN1, POSC, LETEC, FHEX.

DATOS ENTRADA: debe invocarse antes a LLENA.

DATOS SALIDA: Dirección en área reservada para recibir dato desde teclado.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: POSC (citada anteriormente).

DIRECCION: C479H.

\* NOMBRE: FHEX.

FUNCION: Evalua si el código ASCII del último carácter introducido desde el teclado es un dígito hexadecimal válido. En caso afirmativo lo respeta, en caso negativo regresa un cero.

DIRECCION: 0236H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Código a evaluarse en A.

DATOS SALIDA: Código válido o cero en A.

REGISTROS QUE MODIFICA: Si es válido, ninguno. Si no lo es,

## A.

\* NOMBRE: GUADIR.

FUNCION: Almacena en localidades DIR1 y CONDIR, el número binario de 16 bits correspondiente a la dirección recibida desde el teclado.

DIRECCION: 1A09H.

SUBROUTINAS QUE UTILIZA: ASHEX, ROT4.

DATOS ENTRADA: Dirección a almacenarse en área de memoria reservada para dato recibido desde teclado.

DATOS SALIDA: Dirección almacenada en DIR1 y CONDIR.

REGISTROS QUE MODIFICA: A, HL, IY.

\* NOMBRE: ASHEX.

FUNCION: Convierte el código ASCII de un caracter a su equivalente en sistema hexadecimal.

DIRECCION: 022AH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Caracter ASCII en A.

DATOS SALIDA: Equivalente hexadecimal en A.

REGISTROS QUE MODIFICA: A.

\* NOMBRE: ROT4.

FUNCION: Recorre a la izquierda el contenido del acumulador cuatro veces, copiando en bit7 del mismo en el bit0 en cada corrimiento.

DIRECCION: 1B50H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a modificarse en A.

DATOS SALIDA: Dato modificado en A.

REGISTROS QUE MODIFICA: A.

\* NOMBRE: PLANT.

FUNCION: Despliega en pantalla la plantilla que sirve como  
guia al usuario para la introducción de datos.

DIRECCION: 1A37H.

SUBROUTINAS QUE UTILIZA: ESMEN, ESMEN1.

DATOS ENTRADA: Codigos de mensaje en memoria.

DATOS SALIDA: Mensaje en pantalla.

REGISTROS QUE MODIFICA: A, BC, HL.

\* NOMBRE: BIASC.

FUNCION: Convierte un número binario en su equivalente  
decimal, representando este último con los códigos  
ASCII de cada uno de los dígitos que lo conforman.

DIRECCION: 1A4AH.

SUBROUTINAS QUE UTILIZA: LLENA, COMP2.

DATOS ENTRADA: Numero binario en registro BC.

DATOS SALIDA: Códigos ASCII de dígitos decimales en área  
reservada para dato proveniente de teclado.

REGISTROS QUE MODIFICA: BC.

\* NOMBRE: LLENA (citada anteriormente).

DIRECCION: 1964H.

\* NOMBRE: COMP2 (citada anteriormente).

DIRECCION: 1B55H.

\* NOMBRE: DATO.

FUNCION: Introduce un dato desde teclado en formato decimal representado por los códigos ASCII de cada dígito, lo convierte a binario y lo almacena en memoria.

DIRECCION: 1AB4H.

SUBROUTINAS QUE UTILIZA: POSC, INDAT, GUADAS.

DATOS ENTRADA: Número de columna en que posiciona el cursor para recibir dato en A.

DATOS SALIDA: Número binario almacenado en memoria a partir de la dirección especificada por el valor guardado en CONDIR.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: INDAT.

FUNCION: Recibe desde el teclado un dato en formato decimal, representado por los códigos ASCII de cada uno de los dígitos, y lo almacena en el área de memoria reservada para este fin.

DIRECCION: 1B60H.

SUBROUTINAS QUE UTILIZA: LLENA, ESMEN1, POSC, LETEC.

DATOS ENTRADA: Número de columna en que posiciona al cursor para recibir dato en localidad COL.

DATOS SALIDA: Códigos ASCII de dígitos que conforman al

dato, en área reservada para ese fin.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: GUADAS.

FUNCION: Convierte el número recibido por INDAT a formato binario y lo almacena en memoria.

DIRECCION: 1BA6H.

SUBROUTINAS QUE UTILIZA: PREPA, CONV, COMP2.

DATOS ENTRADA: Códigos ASCII de dígitos de número a convertirse en área de memoria reservada para ese fin.

DATOS SALIDA: Número binario equivalente al decimal almacenado en dirección apuntada por el contenido de CONDIR.

REGISTROS QUE MODIFICA: BC, DE, HL.

\* NOMBRE: PREPA.

FUNCION: Prepara un código ASCII para su conversión a binario.

DIRECCION: 1BF1H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Código a convertirse apuntado por el contenido de BC.

DATOS SALIDA: Equivalente decimal de código ASCII almacenado en DE.

REGISTROS QUE MODIFICA: A, DE.

\* NOMBRE: CONV (citada anteriormente).

DIRECCION: 1BF8H.

\* NOMBRE: CONTRO.

FUNCION: Recibe una palabra de control desde el teclado y la almacena sin modificarla en memoria, excepto para el caso en que recibe el código de CTRLX, pues almacena un FFH.

DIRECCION: 1AC2H.

SUBROUTINAS QUE ALMACENA: POSC, INCONT, GUACON.

DATOS ENTRADA: Número de columna en que posiciona al cursor para recibir información en A.

DATOS SALIDA: Palabra de control almacenada en dirección apuntada por el contenido de CONDIR.

REGISTROS QUE MODIFICA: A, HL, IY.

\* NOMBRE: INCONT.

FUNCION: Recibe una palabra de control desde teclado y la almacena en la región menos significativa del Área reservada para dato proveniente desde teclado.

DIRECCION: 1C09H.

SUBROUTINAS QUE UTILIZA: ESPA, POSC, LETEC.

DATOS ENTRADA: Número de columna en que posiciona el cursor para recibir dato en A.

DATOS SALIDA: Palabra de control recibida en región menos significativa de Área reservada para dato proveniente de teclado.



REGISTROS QUE MODIFICA: A, IY.

\* NOMBRE: GUACON.

FUNCION: Almacena en memoria la palabra de control recibida por INCONT sin modificarla, a excepción de cuando se recibe el código de CTRLX, pues en ese caso almacena un FFH.

DIRECCION: 1C3DH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Palabra de control almacenada en parte menos significativa de Área reservada para dato proveniente de teclado.

DATOS SALIDA: Palabra de control almacenada en dirección apuntada por el contenido de CONDIR.

REGISTROS QUE MODIFICA: A, HL.

\* NOMBRE: CHECA.

FUNCION: Checa si la última palabra de control almacenada en memoria corresponde a fin de tabla.

DIRECCION: 1AD0H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dirección de palabra de control mas uno en localidad CONDIR.

DATOS SALIDA: Resultado de chequeo en registro de banderas.

REGISTROS QUE MODIFICA: HL.

\* NOMBRE: CRLF.

FUNCION: Manda a pantalla los caracteres de control para generar un retorno de carro y una alimentación de línea.

DIRECCION: C53AH.

SUBROUTINAS QUE UTILIZA: ESPA.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Comandos hacia pantalla.

REGISTROS QUE MODIFICA: A.

\* NOMBRE: INCR.

FUNCION: Incrementa el número de renglon en que está posicionado el cursor de la pantalla hasta que encuentra su tope máximo.

DIRECCION: C401H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Número de renglon en RENPOS.

DATOS SALIDA: Ya sea número incrementado o número máximo de renglon en RENPOS.

REGISTROS QUE MODIFICA: A.

\* NOMBRE: INCREN.

FUNCION: Incrementa en uno el número del renglon desplegado para la introducción de datos.

DIRECCION: 1AD8H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Número de renglon actual en RENG.

DATOS SALIDA: Número de renglon incrementado en RENG.

REGISTROS QUE MODIFICA: HL.

\* NOMBRE: HEXA.

FUNCION: Convierte un número binario de 8 bits en su equivalente ASCII en representación hexadecimal.

DIRECCION: 0257H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Número binario en A. Registro HL apuntando a la dirección en que se almacenarán los dos nibles resultantes de la transformación.

DATOS SALIDA: Nibbles resultantes de la transformación almacenados en dirección contenida en HL.

REGISTROS QUE MODIFICA: HL.

\* NOMBRE: DESP.

FUNCION: Despliega en pantalla el letrero "DIRECCION FINAL:" y el contenido del área reservada en memoria para dato proveniente desde teclado.

DIRECCION: 1AE0H.

SUBROUTINAS QUE UTILIZA: CRLF, ESMEN, ESMEN1.

DATOS ENTRADA: Valor de la dirección final en representación hexadecimal a base de códigos ASCII en área reservada para dato.

DATOS SALIDA: Mensaje en pantalla.

REGISTROS QUE MODIFICA: A, BC, HL.

\* NOMBRE: NORM.

**FUNCION:** Escala la tabla de usuario para obtener y estructurar la tabla de máquina de una determinada gráfica.

**DIRECCION:** 1AF3H.

**SUBROUTINAS QUE UTILIZA:** ESMEN, NORMA, +-48, NUEDAT, NUECON, SUMIY5.

**DATOS ENTRADA:** Dirección inicial de tabla de usuario en DIR1.

**DATOS SALIDA:** Tabla de máquina a partir de la última localidad utilizada por la de usuario. Coordenadas origen relativo de máquina en ORGX y ORGY. Dirección inicial de tabla de máquina en DIR1 y CONDIR.

**REGISTROS QUE MODIFICA:** A, BC, DE, HL, IY.

\* **NOMBRE:** SUMIY5 (citada anteriormente).

**DIRECCION:** 1CEDH.

\* **NOMBRE:** NORMA.

**FUNCION:** Divide un número de 16 bits entre otro de 8, y al resultado le adiciona 800H.

**DIRECCION:** 1C53H.

**SUBROUTINAS QUE UTILIZA:** DIV.

**DATOS ENTRADA:** Número de 16 bits en registro HL. Número de 8 bits en localidad DIVOR.

**DATOS SALIDA:** Número dividido mas 800H en registro HL.

**REGISTROS QUE MODIFICA:** A, BC, DE, HL.

\* NOMBRE: DIV (citada anteriormente).

DIRECCION: 1CSBH.

\* NOMBRE: +-48.

FUNCION: Adiciona + ó - 48 unidades (dependiendo del signo del número original), al resultado proporcionado por NORMA, con el fin de compensar el redondeo que se lleva a cabo en DIV.

DIRECCION: 1CC1H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato signado en DE. Resultado de NORMA en HL.

DATOS SALIDA: Dato compensado en HL.

REGISTROS QUE MODIFICA: BC, DE, HL.

\* NOMBRE: NUEDAT.

FUNCION: Almacena en memoria el dato resultado del escalamiento y actualiza el valor de CONDIR, para que este apunte al final de la tabla de máquina en construcción.

DIRECCION: 1CD7H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a almacenarse en HL. Dirección en que se almacenara en CONDIR.

DATOS SALIDA: Dato almacenado a partir de la dirección especificada por CONDIR. Localidad última de

tabla de máquina en construcción apuntada por  
CONDIR.

REGISTROS QUE MODIFICA: DE, HL.

\* NOMBRE: NUECON.

FUNCION: Copia la palabra de control correspondiente al  
último dato escalado en la última localidad  
disponible de la tabla de máquina en construcción.

DIRECCION: ICE4H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Último dato de usuario transformado apuntado  
por IY. Localidad correspondiente a palabra  
de control de último dato de máquina  
almacenado en CONDIR.

DATOS SALIDA: Palabra de control almacenada en localidad  
especificada por CONDIR. Valor almacenado en  
CONDIR incrementado en uno.

REGISTROS QUE MODIFICA: A, HL.

#### 3.4.4 Cargar tabla.

La alternativa que en esta sección se presenta, al igual  
que la de la siguiente, está estrechamente relacionada con la  
capacidad que posee el sistema para poder almacenar y  
recuperar posteriormente desde cinta magnética, la  
información que el usuario proporciona.

Específicamente, ACCETA (ACCESa TABla), que es el programa base que se invoca al elegir esta opción, es el encargado de acceder los datos almacenados en cinta y transportarlos hacia la memoria de la máquina, quedando disponibles para su manipulación en cualquiera de las formas que el sistema ofrece.

Para su funcionamiento, hace uso de las subrutinas pertenecientes al sistema operativo de la microcomputadora FAP-100 desarrolladas especialmente para su interrelación con dispositivos periféricos.

La forma en que estas trabajan es de acuerdo al protocolo de comunicación X-MODEM, el cual establece la transmisión de datos en bloques de 128 bytes cada uno, enviándose un indicador de fin de archivo (EOT) al final de éstos. Es necesario aclarar que si los datos no alcanzan a llenar el último bloque, entonces éste se completa con ceros.

En general, las tareas que lleva a cabo ACCETA se pueden dividir en tres partes:

1. Deshabilita la terminal de video y habilita al sistema para su comunicación con cinta.
2. Transmite la información desde la cinta hasta la memoria de la máquina.
3. Habilita nuevamente al sistema para su interrelación con el usuario.

La forma en que realiza estas, es la siguiente:

Borra la pantalla de la terminal y envía un letrero con el que indica al usuario que el sistema está habilitado para

la transmisión y espera el comando adecuado (CR) para iniciar ésta.

Ya que recibio éste, despliega otro mensaje para informar que está ejecutando la captura de datos y habilita al sistema para la comunicación X-MODEM. Para esto, genera ciertos parámetros requeridos por las subrutinas desarrolladas para llevar a cabo esta tarea, y que son: código que habilita a unidad de cinta y código que establece la precedencia de la información.

Con esto, y por medio de LETEC, recibe los parámetros pertenecientes a la tabla de datos residente en cinta (origen usuario, dirección final, longitud del buffer y dirección inicial) con lo establece otros parámetros adicionales, necesarios también para iniciar la comunicación.

En ese momento se llama a la subrutina XMENT, y ésta se encarga de recibir la información con protocolo X-MODEM hasta que se identifica el fin de archivo, con lo que la transmisión termina y se habilita nuevamente la terminal de video para desplegar el status final de la misma (LISTO si no se encuentra ningún error).

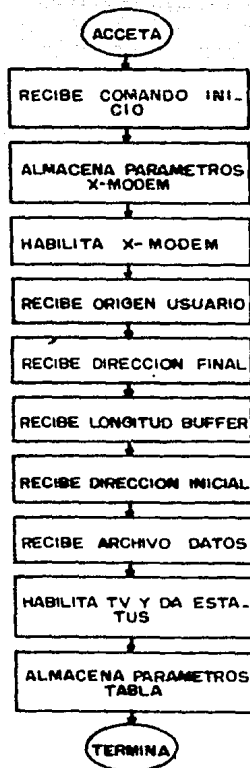
Con la tabla de datos ya en memoria, se procede a almacenar los parámetros pertenecientes a ésta en la forma en que se muestra en la figura 3.4, se actualizan los apuntadores de la misma y se termina con el proceso, esperando el comando adecuado (CR) para regresar al menú principal.

Cabe hacer notar, que todo este proceso se lleva a cabo



con la cinta magnética en movimiento desde que el usuario transmite el código de inicio de la transmisión, por lo que este no tiene que realizar ninguna otra tarea.

Así, el diagrama a bloques de ACCETA es el que se muestra a continuación.



Las características principales, tanto de las subrutinas desarrolladas para el sistema, como las de las implementadas

dentro del sistema operativo de FAP-100, se muestran enseguida, pudiendose consultar el apéndice A de este trabajo para un estudio profundo de las mismas, ya que en éste se expone el listado total de ellas.

\* NOMBRE: ACCETA.

FUNCION: Recibe desde cinta las coordenadas de los puntos que conforman a una determinada tabla de datos, además de los parámetros que pertenecen a esta. La forma de comunicación es en base al protocolo X-MODEM.

DIRECCION: 1CF3H.

SUBROUTINAS QUE UTILIZA: ESPA, ESMEN1, LETEC, ESMEN, INICIO, XMENT, FIN1.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Tabla residente en cinta almacenada en memoria a partir de la dirección especificada por los parámetros que a ella pertenecen.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: INICIO.

FUNCION: Habilita comunicación X-MODEM seleccionando como dispositivo de entrada a la unidad de cinta magnética.

DIRECCION: 0885H.

SUBROUTINAS QUE UTILIZA: RETDO, NVO.

DATOS ENTRADA: Código de dispositivo a seleccionar en

localidad 1100H.

DATOS SALIDA: Parámetro para repetir bloques en localidad 10FBH.

REGISTROS QUE MODIFICA: A, C, D.

\* NOMBRE: RETDO.

FUNCION: Produce un retardo de acuerdo al valor de un parámetro de entrada.

DIRECCION: 09A4H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Parametro de entrada en A.

DATOS SALIDA: Ninguno.

REGISTROS QUE MODIFICA: C, D.

\* NOMBRE: NVO.

FUNCION: Genera y almacena parámetro para repetición de bloque durante la transmisión. Accesa al código del dispositivo que se selecciona para establecer comunicación. Da retardo para activar comunicación X-MODEM.

DIRECCION: 0A63H.

SUBROUTINAS QUE UTILIZA: RETDO.

DATOS ENTRADA: Parámetro para retardo en A. Código dispositivo a seleccionar en localidad 1100H.

DATOS SALIDA: Código para repetición de bloque en localidad 10FB. Código de dispositivo en A.

REGISTROS QUE MODIFICA: A, C, D.

\* NOMBRE: XMENT.

FUNCION: Introduce datos desde cinta en modo serie con protocolo X-MODEM. Toma bloque de 128 bytes cada uno y finaliza la operación al recibir el código de fin de archivo (EOT).

DIRECCION: O9BOH.

SUBROUTINAS QUE UTILIZA: ENT, REPITE, ESPA.

DATOS ENTRADA: Dirección inicial del buffer de entrada en HL. Tope de memoria para recibir datos en TOPE. Parámetro para distinguir procedencia de información en IOFA. Código de dispositivo seleccionado en IIOOH.

DATOS SALIDA: Información almacenada a partir de la dirección especificada por HL. Status final de la transmisión en A.

REGISTROS QUE MODIFICA: A, BC, DE.

\* NOMBRE: ENT.

FUNCION: Checa si existe dato listo en UART. Si existe lo acepta. Checa si existe error de estatus en UART.

DIRECCION: O923H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Dato valido o codigo de error de estatus en

A.

REGISTROS QUE MODIFICA: A.

\* NOMBRE: REPITE.

FUNCION: Evalua parámetros para repetir bloque.

DIRECCION: 094BH.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Parámetro para repetir bloque en IOFB.

DATOS SALIDA: Código de error en repetición almacenado en A.

REGISTROS QUE MODIFICA: A, C.

\* NOMBRE: FINI.

FUNCION: Activa terminal de video y envia mensaje que contiene status final de transmisión.

DIRECCION: 1E10H.

SUBROUTINAS QUE UTILIZA: XMENSA.

DATOS ENTRADA: Código de error en A.

DATOS SALIDA: Mensaje en pantalla.

REGISTROS QUE MODIFICA: A, BC, DE.

\* NOMBRE: XMENSA.

FUNCION: Evalua el código de error y manda mensaje a pantalla de acuerdo al resultado de la evaluación.

DIRECCION: 1E28H.

SUBROUTINAS QUE UTILIZA: ESMEN.

DATOS ENTRADA: Código de error en A.

DATOS SALIDA: Mensaje en pantalla.

REGISTROS QUE MODIFICA: A, BC, DE.

#### 3.4.5 Almacenar tabla.

La función que el sistema graficador ofrece al elegirse esta alternativa, es como complemento a la citada en la sección anterior, ya que las acciones que se llevan a cabo permiten el almacenamiento de información en cinta magnetica.

El programa base diseñado para ejecutar éstas, recibe el nombre de TRACIN (TRAnsporta a CINTa), y su estructura es muy similar a la de ACCETA, con la diferencia que este último accesa la información almacenada por el primero.

Para su funcionamiento, hace también uso de los programas pertenecientes al sistema operativo de la máquina en lo que se refiere a comunicación con la unidad de cinta, en específico: INICIO y XMSAL, los cuales permiten transportar información desde la memoria de la máquina hacia ese dispositivo en formato X-MODEM.

Al igual que su complemento, la tarea la divide en tres partes: Activación de X-MODEM, transportación de información y habilitación de terminal de video para comunicación con el usuario, y la forma en que las lleva a cabo es la siguiente:

Inicia borrando la pantalla de la terminal y desplegando el mensaje que informa al usuario que el sistema esta listo para iniciar la transmisión (en este momento, el usuario debe activar la unidad de cinta en el modo de grabación). Con

esto, espera el comando para empezar el proceso de comunicación.

Cuando este es recibido, genera los parámetros para: seleccionar la unidad de cinta como destino de la información, reconocer el tipo de periférico con el que se relacionará y el formato en que los datos serán suministrados. Despliega el mensaje "TRANS" como indicativo de que la comunicación se esta llevando a cabo y desactiva la terminal de video para activar a la de cinta.

Actualiza los parámetros que pertenecen a la tabla de datos a transmitirse (longitud del buffer, dirección inicial, etc.), y por medio de ESPA los almacena en la cinta magnética, para inmediatamente generar los que utiliza el programa XMSAL.

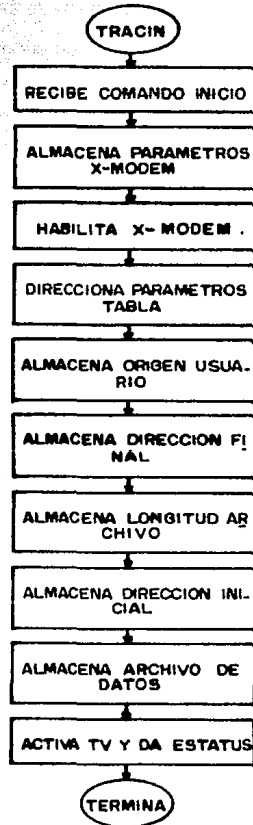
Invocando a este último, transmite la información que conforma la tabla de datos en formato X-MODEM hasta que encuentra el código de fin de archivo (EOT), con lo que termina esta etapa.

Finalmente activa nuevamente la terminal de video y despliega el status de la transmisión y el mensaje con el que hace del conocimiento de usuario que está ha sido terminada, para esperar el comando adecuado (CR) con el que regresa al menú principal.

Así, el diagrama a bloques de TRACIN es el que se muestra en la siguiente ilustración, en donde se puede observar claramente todo el proceso.

Para la implementación de éste, se desarrollaron las

subrutinas necesarias para complementar las que residen en memoria, y de las cuales se presenta a continuación las características principales, teniéndose que consultar el apéndice A de este trabajo para estudiar el listado completo de las mismas.





\* NOMBRE: TRACIN.

FUNCION: Almacena en cinta una determinada tabla de datos en modo serie con formato X-MODEM. Además, almacena también los parámetros que a ésta pertenecen.

DIRECCION: 1D8EH.

SUBROUTINAS QUE UTILIZA: ESPA, ESMEN1, LETEC, ESMEN, INICIO, XMSAL, FIN1.

DATOS ENTRADA: Direccion inicial, direccion final y longitud del buffer en las 6 localidades que anteceden al inicio de la tabla de usuario. Origen relativo de la tabla en localidades ORGUSX y ORGUSY.

DATOS SALIDA: Tabla almacenada en cinta.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY, IX.

\* NOMBRE: INICIO (citada anteriormente).

DIRECCION: 0885H.

\* NOMBRE: XMSAL.

FUNCION: Entrega a periférico de salida datos en modo serie con protocolo X-MODEM. En caso necesario completa el último bloque con ceros.

DIRECCION: 08AAH.

SUBROUTINAS QUE UTILIZA: ESPA, RESP.

DATOS ENTRADA: Direccion inicial del buffer en HL. Longitud del buffer en IX. Código de dispositivo a

seleccionarse en 1100. Parámetro para reconocer el tipo de dispositivo destino en 10FA. Código para reconocer formato en que se transmite el dato en 1102.

DATOS SALIDA: Código de status final de la transmisión en  
A. Tabla almacenada en cinta.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY, IX.

\* NOMBRE: RESP.

FUNCION: Evalua en base al parámetro almacenado en 10FA si el dispositivo destino genera respuesta después de que recibió un dato. En caso afirmativo busca a ésta, en caso contrario no.

DIRECCION: 093A.

SUBROUTINAS QUE UTILIZA: ENTACK.

DATOS ENTRADA: Código para reconocer o no respuesta en 10FA.

DATOS SALIDA: Resultado de la búsqueda en registro de banderas.

REGISTROS QUE MODIFICA: A, DE.

\* NOMBRE: ENTACK.

FUNCION: Busca respuesta del dispositivo destino en caso de que este pueda generarla.

DIRECCION: 0971H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Código de respuesta en A.

REGISTROS QUE MODIFICA: 2, DE.

\* NOMBRE: FIN1 (citada anteriormente).

DIRECCION: 1E10H.

### 3.4.6 Graficar tabla.

La alternativa que a continuación se estudia, es quizá la más significativa de las que ofrece el sistema graficador, pues el programa base que se invoca al elegirla, es el encargado de llevar cabo la conversión de la información numérica suministrada por el usuario, para exponerla en forma gráfica.

El que este programa ejecute esta tarea, implica que debe ser capaz de generar los códigos de control en base a los cuales se comunicará con la interfaz D/A, para el proceso de la información que desde la memoria de la máquina se suministrará, además de guardar una relación estrecha con los algoritmos de graficado, para el cálculo correcto de esta última.

El nombre con que se denota a este programa es GRAF12 (GRAFica con 12 bits), y un estudio detallado de su funcionamiento, así como de la relación que guarda con las subrutinas restantes del sistema, se presenta en la sección 2.1.3 de este trabajo, en donde se puede observar que cumple

con las características antes citadas.

### 3.4.7 Modificación dato a dato.

Con la presente sección, se inicia el estudio de las alternativas a las que se tiene acceso por medio del menú de modificación que ofrece el sistema graficador. En ellas se abordan los programas base diseñados para que el usuario tenga la oportunidad de manipular la información introducida previamente.

La primera de éstas, es la que se refiere a la modificación parcial de una determinada tabla de datos, y recibe el nombre de modificación dato a dato debido a que el proceso de transformación se lleva a cabo desde, para un solo dato o palabra de control, hasta para la totalidad de la tabla.

El programa base que se invoca al elegirla, recibe el nombre de MODADA (MODifica DATo a DATo), y su funcionamiento se puede dividir en las siguientes etapas:

1. Accesar una determinada tabla de datos a partir de cualquiera de los renglones que a ésta conforman.
2. Desplegar la información contenida en éste.
3. Aceptar las correcciones que sean necesarias.
4. Almacenar en memoria la nueva información y con esto generar la tabla de máquina modificada, para su posterior

uso.

Para ejecutar cada una de estas, MODADA hace uso de diferentes subrutinas, de las que destacan las básicas de comunicación con el exterior, y las cuales fueron estudiadas al inicio de este capítulo, pues en ellas en las que se basa la construcción de las implementadas específicamente como soporte de este programa base.

La forma en que lleva a cabo las diferentes tareas que tiene encomendadas es la siguiente:

Inicialmente, por medio de ESMEN1 despliega en pantalla el mensaje

"DIRECCION INICIAL: "

para requerir del usuario la dirección inicial de la tabla de datos que se modificara, para, por medio de INDIR introducir el valor de está e inmediatamente almacenarlo.

Hecho esto, manda hacia la terminal de video el mensaje indicativo para que el usuario introduzca el número de renglon a partir del cual se modificara el archivo. Este tiene la estructura siguiente:

" RENGLON NO.? "

y a partir de la información que se le suministra, direcciona los datos que corresponden al número de renglon elegido. Dado que en cada uno de éstos se despliegan dos pares de coordenadas x,y (donde cada una de ellas ocupa dos bytes), con sus respectivas palabras de control, la tarea de

direccionar al dato correcto la ejecuta en base a la siguiente expresión:

dato a desplegar = (no. de renglon \* 10) - 10

y el resultado lo adiciona a la dirección inicial de la tabla.

Con el dato direccionado, borra la pantalla de la terminal de video y despliega en el primer renglon de ésta, la plantilla que sirve al usuario para reconocer la correspondencia del dato exhibido y la función que desempeña dentro de la tabla de datos (ésta, es la misma que se utiliza en GENTA).

En ese momento, se invoca a la subrutina DESREN, que es la encargada de desplegar en la posición adecuada de la plantilla, tanto el número de renglon elegido por el usuario, como los valores de los datos que corresponden a cada una de las columnas. Con esto, se posiciona nuevamente en el primero de estos y por medio de MODA, admite el dato modificado que corresponde a esta columna hasta que recibe el código de "espacio", con lo que almacena en las mismas localidades de la tabla que ocupaba el dato anterior la nueva información, para inmediatamente direccionar el dato de la siguiente columna y realizar la misma acción.

Cuando el dato a modificarse es el correspondiente a una palabra de control, el programa puede realizar tres diferentes acciones:

- a. Admitir un código válido para graficación, con lo que

el proceso de corrección continua.

b. Reconocer el código correspondiente a el comando CTRL-Y, con lo que sale del modo de modificación y respeta la longitud del archivo y el valor de los datos que no fueron accedados.

c. Reducir el tamaño del archivo inicial recibiendo el código del comando CTRL-X.

Adicionalmente, si la palabra accesada es la indicadora de fin de tabla, esta puede ser sustituida con una valida para graficación, con lo que la longitud del archivo se alarga. De otra forma, si esta no se modifica, el proceso termina automaticamente.

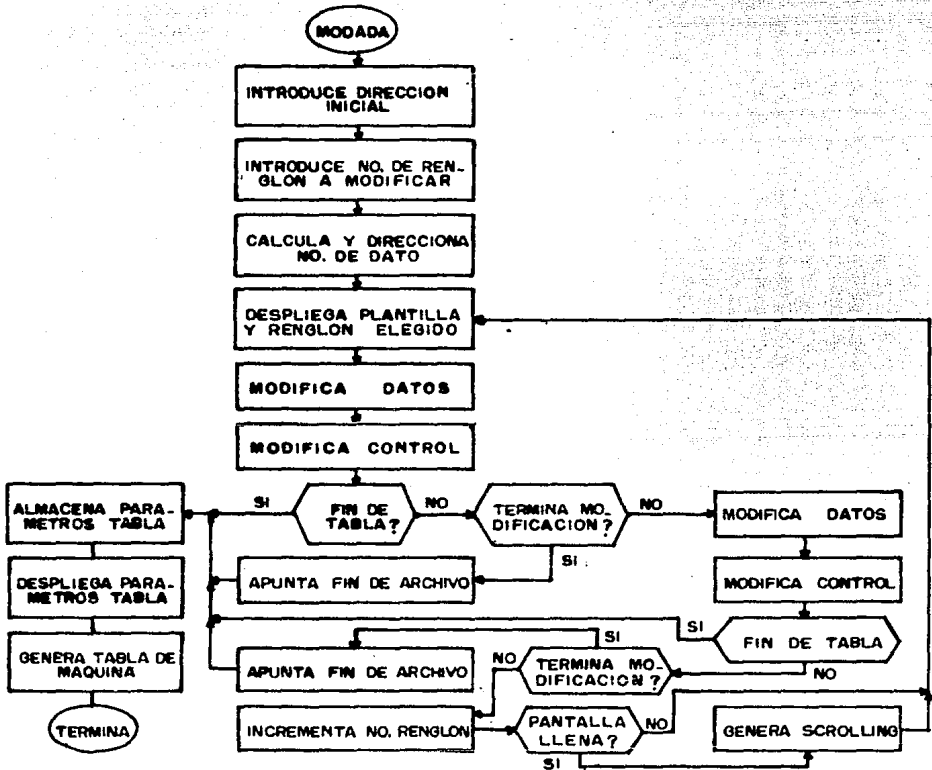
Si las correcciones se continuan para un número de renglones mayor que el desplegado primeramente, el proceso se repite para todos y cada uno de los subsecuentes, desplegandose el numero de renglon en turno y la información que a el corresponde.

Finalmente, cuando el proceso es terminado, la acción que se ejecuta es exactamente igual a la última que realiza GENTA, y es la que se refiere a el escalamiento de la tabla de usuario o la generación de la tabla de máquina. Esto conlleva la actualización de ciertos parametros, como son: origen relativo (se da el de entrada nuevamente), dirección inicial (si ésta sufrió modificaciones) y apuntador de fin de tabla.

También se generan los letreros indicativos para que el usuario tenga concimiento de algunos de los parámetros

pertencientes a la nueva tabla de datos, y estos se almacenan en forma adecuada (en los diez bytes anteriores al inicio), con lo que se espera el comando adecuado (CR) para regresar al menú de modificación.

Con esto, el diagrama a bloques de esta subrutina es entonces, el que la siguiente ilustración presenta.



Las características principales de cada una de las



subrutinas implementadas para este fin, se exponen a continuación, pudiendose consultar el listado completo de cada una de ellas en el apéndice A de este trabajo.

\* NOMBRE: MODADA.

FUNCION: Permite la realización de correcciones a un determinado archivo pudiendose acceder desde un solo dato hasta la totalidad de éste. Genera además, la tabla de máquina de este nuevo archivo.

DIRECCION: 1EB5H.

SUBROUTINAS QUE UTILIZA: ESPA, LLENA, ESMEN1, INDIR, GUADIR, CRLF, DATO, CALCRE, ACTCON, PLANT, INCR, DESREN, MODATO, MODCON, CHECA, FIN, TERM, INCREN, POSC, LETEC.

DATOS ENTRADA: Tabla a modificarse almacenada en memoria.

DATOS SALIDA: Tabla modificada a partir de la misma dirección con que iniciaba la tabla original. Tabla de máquina nueva a partir de la última localidad utilizada por la de usuario.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: LLENA (citada anteriormente).

DIRECCION: 19C4H.

\* NOMBRE: INDIR (citada anteriormente).

DIRECCION: 19DDH.

\* NOMBRE: GUADIR (citada anteriormente).

DIRECCION: 1A09H.

\* NOMBRE: CRLF (citada anteriormente).

DIRECCION: C53AH.

\* NOMBRE: DATO (citada anteriormente).

DIRECCION: 1AB4T.

\* NOMBRE: CALCRE.

FUNCION: Calcula el número de bytes que tiene que saltar para apuntar al primer dato que conforma al renglon elegido. El resultado obtenido tiene que ser adicionado a la dirección inicial.

DIRECCION: 1F66H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Número de renglon elegido en localidad RENG.

DATOS SALIDA: Número de bytes que se necesitan saltar en HL.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: ACTCON.

FUNCION: Actualiza el apuntador de la tabla de datos adicionando a la dirección inicial el número de bytes a saltarse para apuntar al dato primero del renglon elegido.

\* NOMBRE: GUADIR (citada anteriormente).

DIRECCION: 1A09H.

\* NOMBRE: CRLF (citada anteriormente).

DIRECCION: C53AH.

\* NOMBRE: DATO (citada anteriormente).

DIRECCION: 1AB4T.

\* NOMBRE: CALCRE.

FUNCION: Calcula el número de bytes que tiene que saltar para apuntar al primer dato que conforma al renglon elegido. El resultado obtenido tiene que ser adicionado a la dirección inicial.

DIRECCION: 1F66H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Número de renglon elegido en localidad RENG.

DATOS SALIDA: Número de bytes que se necesitan saltar en HL.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: ACTCON.

FUNCION: Actualiza el apuntador de la tabla de datos adicionando a la dirección inicial el número de bytes a saltarse para apuntar al dato primero del renglon elegido.

DIRECCION: 1F82H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Número de bytes a saltarse en registro HL.

DATOS SALIDA: Apuntador actualizado en localidad CONDIR.

REGISTROS QUE MODIFICA: DE, HL.

\* NOMBRE: PLANT (citada anteriormente).

DIRECCION: 1A37H.

\* NOMBRE: INCR (citada anteriormente).

DIRECCION: C401H.

\* NOMBRE: DESREN.

FUNCION: Despliega en pantalla tanto el número de renglon en turno, como la información que a el corresponde. Si la tabla termina cuando se despliega la primer palabra de control de éste, la accion es truncada.

DIRECCION: 1F8BH.

SUBROUTINAS QUE UTILIZA: CONDES, POSC, BCHL, ESPA.

DATOS ENTRADA: Número de renglon en RENG. Dirección del primer dato a desplegarse en CONDIR.

DATOS SALIDA: Datos despelgados en pantalla.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: CONDES.

FUNCION: Transforma un dato en formato binario a su

equivalente decimal expresado en base a los  
códigos ASCII de cada uno de los dígitos que  
lo conforman.

DIRECCION: 202CH.

SUBROUTINAS QUE UTILIZA: BIASC.

DATOS ENTRADA: Dato a transformarse en registro BC.

DATOS SALIDA: Códigos ASCII de dato transformado en área de  
memoria reservada para dato proveniente de  
teclado.

REGISTROS QUE MODIFICA: A, BC, HL.

\* NOMBRE: BIASC (citada anteriormente).

DIRECCION: 1A4AH.

\* NOMBRE: POSC (citada anteriormente).

DIRECCION: C479H.

\* NOMBRE: BCHL.

FUNCION: Almacena en el registro BC, el byte apuntado por  
el registro HL y el que le sigue a éste.

DIRECCION: 2036H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a cargarse en BC apuntado por HL.

DATOS SALIDA: Dato almacenado en BC. HL incrementado en  
dos.

REGISTROS QUE MODIFICA: BC, HL.

\* NOMBRE: MODATO.

FUNCION: Acepta correcciones sobre un dato perteneciente a una determinada tabla y almacena la nueva información en el mismo lugar que ocupaba la original.

DIRECCION: 1FF2H.

SUBROUTINAS QUE UTILIZA: MODA, GUADAS.

DATOS ENTRADA: Localidad CONDIR apuntando al dato a corregirse.

DATOS SALIDA: Dato corregido almacenado en la localidad apuntada por CONDIR.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: MODA.

FUNCION: Modifica el valor de un dato perteneciente a un determinado archivo.

DIRECCION: 203BH.

SUBROUTINAS QUE UTILIZA: BCHL, BIASC, LETEC, ESMEN1, POSC.

DATOS ENTRADA: Localidad CONDIR apuntando al dato a transformarse.

DATOS SALIDA: Dato corregido en área de memoria reservada para dato proveniente de teclado, almacenado en formato decimal a base de los códigos ASCII de cada uno de los dígitos.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: GUADAS (citada anteriormente).

DIRECCION: 1BA6H.

\* NOMBRE: MODCON.

FUNCION: Modifica una palabra de control perteneciente a una determinada tabla de datos y la almacena en la localidad de memoria que ocupaba la original.

DIRECCION: 2000H.

SUBROUTINAS QUE UTILIZA: POSC, MODICO, GUACON.

DATOS ENTRADA: Palabra de control a modificarse apuntada por HL.

DATOS SALIDA: Palabra de control nueva almacenada en dirección apuntada por CONDIR. Registro HL incrementado en uno.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: MODICO.

FUNCION: Modifica una palabra de control y el resultado lo almacena en el espacio menos significativo del área de memoria reservada para dato proveniente de teclado.

DIRECCION: 2084H.

SUBROUTINAS QUE UTILIZA: LETEC, ESPA, POSC.

DATOS ENTRADA: Palabra de control a modificarse apuntada por el registro HL.

DATOS SALIDA: Código ASCII de palabra de control nueva, en espacio menos significativo de área de memoria reservada para dato proveniente de

teclado.

REGISTROS QUE MODIFICA: A, IY.

\* NOMBRE: GUACON (citada anteriormente).

DIRECCION: 1C3DH.

\* NOMBRE: CHECA (citada anteriormente).

DIRECCION: 1ADQH.

\* NOMBRE: FIN.

FUNCION: Calcula y almacena en los 10 bytes que anteceden a un determinado archivo, los parámetros que a este pertenecen.

DIRECCION: 195DH.

SUBROUTINAS QUE UTILIZA: HEXA, DESP, ESMEN1, ESMEN, NORM, LETEC.

DATOS ENTRADA: Dirección inicial de archivo en DIR1.  
Dirección final del mismo en HL.

DATOS SALIDA: Parámetros de tabla almacenados y tabla de máquina a partir de la última localidad utilizada por la de usuario.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: HEXA (citada anteriormente).

DIRECCION: 0257H.

\* NOMBRE: DESP (citada anteriormente).



DIRECCION: 1AEOH.

\* NOMBRE: NORM (citada anteriormente).

DIRECCION: 1AF3H.

\* NOMBRE: TERM.

FUNCION: Actualiza el apuntador de fin de archivo para generar tabla de máquina y calcular, así como almacenar, los parámetros que pertenecen a ésta.

DIRECCION: 2OIAH.

SUBROUTINAS QUE UTILIZA: CHECA, FIN.

DATOS ENTRADA: Tabla de datos apuntada por HL.

DATOS SALIDA: Parámetros de tabla almacenados en los 10 bytes que anteceden a su inicio. Tabla de máquina a partir de la última localidad utilizada por la de usuario.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: INCREN (citada anteriormente).

DIRECCION: 1AD8H.

\* NOMBRE: POSC (citada anteriormente).

DIRECCION: C479H.

## 3.4.8 Escalamiento.

Otra de las posibilidades de modificación que ofrece el sistema graficador, es la que se engloba en el renglon de escalamiento, incluido en el menú de modificación. A diferencia de la anterior, por medio de ésta se afecta a la totalidad del archivo, y permite al usuario el poder reducir o amplificar el tamaño de una determinada gráfica.

Para ejecutar estas acciones, se cuenta con dos programas base diseñados específicamente para estos fines: RDUC (ReDUCcion) para la primera y AMP (AMPlificacion) para la segunda, los cuales se invocan de acuerdo al código que les antecede en el menú de selección.

Los fundamentos matemáticos en los que se basan estos programas, se analizan detalladamente en las secciones 2.3.2 y 2.3.3 del capítulo anterior, por lo que aquí solo se citan las ecuaciones finalmente desarrolladas, a partir de las cuales se estructuran estas alternativas.

- Para reducción:

$$X' = X \cdot 1/rx$$

$$Y' = Y \cdot 1/ry$$

- Para amplificación:

$$X' = X \cdot Ax$$

$$Y' = Y \cdot Ay$$

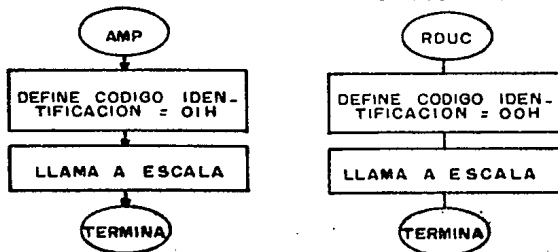
donde  $r_x$ ,  $r_y$ ,  $A_x$  y  $A_y$  son los factores de escalamiento, y pueden variar de manera que su valor sea mayor o igual que 1, siempre utilizando números enteros.

Debido a la similitud que existe entre las etapas que siguen estos dos programas para llevar a cabo las tareas encomendadas (en una se ejecuta división y en la otra solo se cambia a multiplicación), para su implementación en el sistema, muchas de las subrutinas desarrolladas para uno, se utilizaron también para el otro.

Esto trajo como consecuencia la construcción de la subrutina ESCALA, que es la encargada de modificar la operación requerida por el tipo de transformación indicado, el cual es reconocido por medio de un código de identificación que establecen los programas base cuando invocan a ésta.

Así, cuando el código de identificación es OOH, el dato es dividido entre el valor del factor de escalamiento, mientras que cuando es OIH, se multiplica por el mismo.

Ya que ESCALA es capaz de realizar la transformación tanto de la tabla de usuario como de la de máquina, además de requerir los factores de escalamiento, entonces la función de RDUC y AMP se reduce solamente a definir el código apropiado e invocarla. Con esto, los diagramas a bloques de estos dos programas base son los que a continuación se muestran.



En general, las acciones que se llevan a cabo cuando se invoca a ESCALA son las siguientes:

1. Se requiere del usuario los valores de los factores de escalamiento para cada uno de los ejes.
2. Se transforma la tabla de usuario.
3. Se modifica en la misma proporción la tabla de máquina.

Estas se realizan por medio de las tres subrutinas principales de ESCALA: REQPAR, ESCUSU y ESCMAQ respectivamente, de las cuales la última es ampliamente explicada en las secciones correspondientes a los programas de transformación del capítulo anterior, por lo que durante la explicación del proceso en esta sección, solo se expondrán sus características principales. En lo que se refiere a ESCUSU, el programa genera la posibilidad de utilizarla o no, dependiendo del interés del usuario, con lo que se brinda a este la oportunidad de llevar a cabo pruebas sin tener que perder la información original de la gráfica.

La forma en que estas acciones son ejecutadas, es la siguiente:

Con el código de identificación definido, se invoca a ESCALA, quien a su vez hace uso inmediatamente de REQPAR. Esta, borra la pantalla y despliega el mensaje

#### FACTOR X?

con lo que recibe el valor de este. Cabe aclarar que el sistema da de entrada el valor de 1 para este factor, y que el proceso de introducción termina con un CR.

Hecho esto, posiciona adecuadamente el cursor de la pantalla y despliega el mensaje

#### FACTOR Y?

con lo que realiza la captura de este dato, proceso cuyas características son las mismas que para el eje x.

Con los dos factores almacenados, proporciona al usuario un tercer mensaje, con el que ofrece a éste la oportunidad de modificar las dos tablas (de usuario y de máquina), o solamente la de máquina. Este tiene la siguiente estructura:

#### MODIFICA TABLA USUARIO? (S/N)

Con esta acción, termina la subrutina REQPAR.

Con este último mensaje desplegado en pantalla, por medio de LETEC se recibe la respuesta del usuario. Si esta es afirmativa se invoca directamente a ESCUSU, en caso contrario

se actualizan los apuntadores de tabla y se invoca a ESCMAQ, con lo que se realiza el proceso citado en las secciones 2.3.2 y 2.3.3.

Cuando la subrutina invocada es la que transforma la tabla de usuario, la forma en que se cubren las etapas para alcanzar este objetivo es muy semejante a la seguida por ESCMAQ, aunque a diferencia de ésta, el proceso principia requiriendo el valor de la dirección inicial de la tabla a transformarse.

Después de esto, se accesan secuencialmente los datos que conforman el archivo, modificandose por medio de la subrutina DEFOP, que es la encargada de implementar la operación adecuada en base al valor del código de identificación. La transformación no afecta las palabras de control y termina cuando se encuentra con la que define el fin de archivo (FFH).

Al terminar su tarea ESCUSU, se inicia la segunda parte del proceso, que es la misma que cuando la respuesta del usuario es negativa a la pregunta de modificación de la tabla de usuario. Con esto se garantiza la transformación de la grafica correspondiente al archivo en cuestión.

El proceso termina desplegando el letrero "LISTO." y esperando el comando adecuado (CR) para regresar al menú principal.

A continuación se exponen las características principales de las subrutinas implementadas para este fin, teniendose que consultar el apéndice A de este trabajo para

un estudio detallado de ellas, pues es ahí donde se presenta el listado total de las mismas.

\* NOMBRE: RDOC.

FUNCION: Reduce el tamaño de una gráfica escalando de manera apropiada la tabla de datos que le da

DIRECCION: 20C7H.

SUBROUTINAS QUE UTILIZA: ESCALA.

DATOS ENTRADA: Dirección inicial de tabla de máquina en localidad DIR1.

DATOS SALIDA: Tabla trasformada a partir de la dirección almacenada en DIR1.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: AMP.

FUNCION: Aplifica el tamaño de una gráfica escalando apropiadamente la tabla de datos que da origen a ésta.

DIRECCION: 20DOH.

SUBROUTINAS QUE UTILIZA: ESCALA.

DATOS ENTRADA: Dirección inicial de tabla de máquina en localidad DIR1.

DATOS SALIDA: Tabla transformada a partir de la dirección almacenada en DIR1.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: ESCALA.

**FUNCION:** Realiza el escalamiento de una tabla de datos aplicando la operación multiplicación o división, dependiendo del valor del código de identificación.

**DIRECCION:** 210FH.

**SUBROUTINAS QUE UTILIZA:** REQPAR, ESCUSU, ESCMAQ, DEFORI, LETEC.

**DATOS ENTRADA:** Código de identificación en localidad OP.  
Dirección inicial de tabla de máquina en DIR1.

**DATOS SALIDA:** Tabla transformada a partir de dirección almacenada en DIR1.

**REGISTROS QUE MODIFICA:** A, BC, DE, HL, IY.

\* **NOMBRE:** REQPAR.

**FUNCION:** Requiere del usuario el valor de cada uno de los factores de escalamiento. Para esto despliega en pantalla los letreros adecuados para preguntar por ellos. Adicionalmente, despliega la pregunta acerca de modificar o no la tabla de usuario.

**DIRECCION:** 2134H.

**SUBROUTINAS QUE UTILIZA:** ESPA, ESMEN1, INFAC, GUADAS, CRLF, INCR.

**DATOS ENTRADA:** Ninguno.

**DATOS SALIDA:** Factores de escalamiento en localidades FACTY y FACTY. Mensajes en pantalla.

**REGISTROS QUE MODIFICA:** A, BC, DE, HL.



\* NOMBRE: INFAC.

FUNCION: Recibe desde el teclado el valor de un factor de escalamiento, dando de entrada el valor de 1.

DIRECCION: 223AH.

SUBROUTINAS QUE UTILIZA: LLENA, ESMEN1, POSC, LETEC.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Dato en formato decimal a base de los códigos ASCII de cada uno de sus dígitos, en Área de memoria reservada para proveniente de teclado.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: LLENA (citada anteriormente).

DIRECCION: 19C4H.

\* NOMBRE: POSC (citada anteriormente).

DIRECCION: C479H.

\* NOMBRE: GUADAS (citada anteriormente).

DIRECCION: 18A6H.

\* NOMBRE: CRLF (citada anteriormente).

DIRECCION: C53AH.

\* NOMBRE: INCR (citada anteriormente).

DIRECCION: C401H.

\* NOMBRE: ESCUSU.

FUNCION: Transforma la tabla de usuario correspondiente a la gráfica que se este escalando. La operación que aplica depende del valor del código de identificación.

DIRECCION: 2186H.

SUBROUTINAS QUE UTILIZA: CRLF, ESPA, ESMEN1, LLENA, INDIR, GUADIR, ESMEN, DEFOP, NUEDAT, NUECON, SUMIY5.

DATOS ENTRADA: Código de identificación almacenado en localidad OP.

DATOS SALIDA: Tabla de usuario transformada en el mismo espacio de memoria que la original.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: INDIR (citada anteriormente).

DIRECCION: 19DDH.

\* NOMBRE: GUADIR (citada anteriormente).

DIRECCION: 1A09H.

\* NOMBRE: DEFOP.

FUNCION: En base al código de identificación, elige y aplica a un dato la operación adecuada para el escalamiento requerido.

DIRECCION: 2281H.

SUBROUTINAS QUE UTILIZA: CONV, DESPLA, DIV.

DATOS ENTRADA: Dato a ser escalado en registro HL. Código de identificación en OP.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: CONV (citada anteriormente).

DIRECCION: 1BF8H.

\* NOMBRE: DESPLA (citada anteriormente).

DIRECCION: 22F7H.

\* NOMBRE: DIV (citada anteriormente).

DIRECCION: 1C5BH.

\* NOMBRE: NUEDAT (citada anteriormente).

DIRECCION: 1CD7H.

\* NOMBRE: NUECON (citada anteriormente).

DIRECCION: 1CE4H.

\* NOMBRE: SUMIY5 (citada anteriormente).

DIRECCION: 1CEDH.

### 3.4.9 Desplazamiento.

La tercer alternativa a la que se tiene acceso dentro de

los programas de transformación, es la que se refiere a desplazamiento, la cual permite trasladar una gráfica de un lugar a otro en el plano sin modificar su tamaño ni su posición angular con respecto a los ejes de este último.

El programa base que se invoca al elegir esta alternativa recibe el nombre de DES (DESplazamiento), y al igual que RDUC y AMP, basa su funcionamiento en la subrutina ESCALA, por lo que las etapas en que se divide el proceso son:

1. Requerimiento e introducción de factores de escalamiento.
2. Transformación de la tabla de usuario.
3. Transformación de tabla de máquina.

De estas, la segunda es también optativa, y la última se explica detalladamente en la sección 2.3.4.

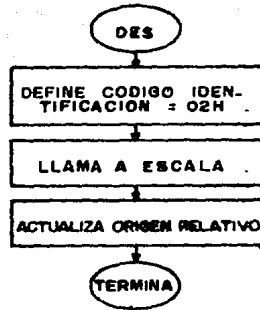
Aunque en general la forma en que las tareas arriba mencionadas se ejecutan como se explicó en la sección anterior, la función de DESPLA no se restringe únicamente a definir el código de identificación que corresponde a ella (O2H), sino que lleva a cabo otras acciones adicionales, las cuales son:

A. Escala los factores asignados a cada uno de los ejes, de manera que sean compatibles con el área de graficado del sistema.

B. Actualiza las coordenadas del origen relativo

adicionando el resultado obtenido en la etapa anterior a cada una de ellas.

Con esto, el diagrama a bloques de este programa base, es el que a continuación se muestra.



Cabe aclarar que las subrutinas en las que recae la responsabilidad de llevar a cabo el desplazamiento son DEFOP y DEFOPM, quien a su vez invocan a DESPLA y DESPM respectivamente. Estas últimas son las encargadas de adicionar el valor de los factores de escalamiento a cada uno de los datos que conforman al archivo.

Las características principales de estas, y de todas las subrutinas desarrolladas para este fin, son las que se exponen a continuación, pudiendose consultar el apéndice A de este trabajo para estudiar el listado completo de ellas.

\* NOMBRE: DES.

FUNCION: Establece el código de identificación correspondiente a la alternativa de desplazamiento. Transforma un archivo de datos de

manera que la gráfica que representa se ve trasladada en el plano.

DIRECCION: 20D9H.

SUBROUTINAS QUE UTILIZA: ESCALA.

DATOS ENTRADA: Dirección inicial de tabla de máquina a transformarse en DIR1.

DATOS SALIDA: Tabla transformada a partir de la dirección almacena en DIR1.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: ESCALA (citada anteriormente).

DIRECCION: 21OFH.

Aunque las características de las subrutinas que componen a ESCALA se citaron en su totalidad en la sección anterior, se listan a continuación las relacionadas estrechamente con la alternativa en estudio, para una mejor exposición de ella.

\* NOMBRE: DESPLA.

FUNCION: Adiciona a un dato proveniente del archivo a transformarse, el valor del factor de escalamiento sin modificar el valor de este último. Se utiliza para transformar la tabla de usuario.

DIRECCION: 22F7H.

SUBROUTINAS QUE UTILIZA: Ninguna.

DATOS ENTRADA: Dato a modificarse en HL. Factor de

escalamiento en localidad DIVOR.

DATOS SALIDA: Dato desplazado en registro HL.

REGISTROS QUE MODIFICA: DE, HL.

\* NOMBRE: DESPM.

FUNCION: Adiciona a un dato proveniente del archivo a transformarse, el factor de desplazamiento escalado, de manera que su valor es compatible con el área de graficado del sistema. Se utiliza para transformar la tabla de máquina.

DIRECCION: 22FDH.

SUBROUTINAS QUE UTILIZA: DIV, +-48.

DATOS ENTRADA: Dato a transformarse en registro HL. Factor de escalamiento en localidad DIVOR.

DATOS SALIDA: Dato transformado en registro HL.

REGISTROS QUE MODIFICA: BC, DE, HL.

### 3.4.10 Rotación.

La última de las alternativas de transformación que ofrece el sistema graficador, es la que se refiere a la de rotación de gráficas. Por medio de ésta, el usuario es capaz de rotar alrededor de un determinado eje, la representación en el plano de un archivo de datos.

El programa base encargado de realizar esta tarea recibe el nombre de ROT, y aunque su funcionamiento es muy parecido

al de los programas de transformación antes citados, éste no depende de ESCALA para ejecutar las acciones necesarias para el fin establecido

Los fundamentos matemáticos en los que se basa para llevar a cabo la transformación, se exponen con detalle en la sección 2.3.4, de donde se puede resumir lo siguiente:

Una gráfica puede ser rotada alrededor de un origen relativo cualquiera, si primero se traslada de manera que este coincida con el absoluto. Con esto, todos y cada uno de los puntos extremos que la definen se afectan por las ecuaciones

$$X' = X \cos \theta - Y \sin \theta .$$

$$Y' = X \sin \theta + Y \cos \theta .$$

con lo que se genera la figura rotada. Finalmente, esta última se traslada de manera que el origen relativo retome su posición inicial.

Cabe aclarar que el sentido de rotación siempre será contra el movimiento de las manecillas del reloj, y puede variar entre 0 y 360 grados.

Las acciones citadas, ROT las ejecuta dos veces, una para la tabla de usuario y otra para la de máquina, aunque la primera de ellas es opcional.

En general, las tareas que se realizan se dividen entonces en tres partes:

1. Requerimiento al usuario tanto del ángulo de rotación como de la respuesta a escalar o no la tabla de



usuario.

2. Transformación de la tabla de usuario.
3. Transformación de tabla de máquina.

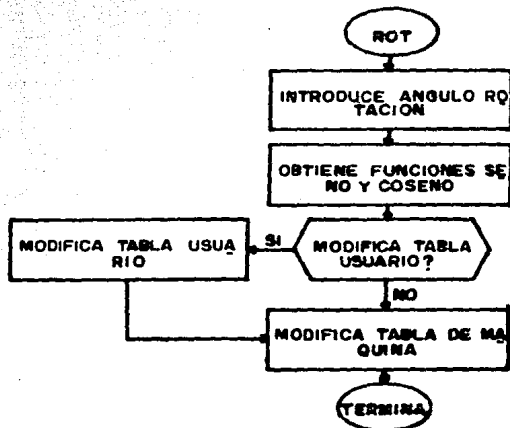
La primera de ellas es ejecutada por la subrutina REQANG, la cual, además de introducir el ángulo de rotación proporcionado por el usuario, calcula la función seno y coseno del mismo (esto en base al algoritmo citado en el apéndice E), para finalmente establecer la pregunta acerca de modificar o no la tabla de usuario.

La segunda es realizada por la subrutina ROTUSU, la que al igual que en las alternativas anteriores, requiere para principiar su tarea, la dirección inicial del archivo a modificarse. Esta toma en cuenta durante el proceso de transformación, si el elemento a rotarse es un círculo o una recta, y dependiendo de la naturaleza de éste lo ataca debidamente, como se estudio en el capítulo anterior.

La última de las funciones es ejecutada por ROTMAQ, y su funcionamiento se pude consultar en forma detallada, en la sección dedicada a esta alternativa de transformación del capítulo anterior.

Con esto, el diagrama a bloques de este programa es el que en la siguiente ilustración se muestra. Antes de citar las características principales de cada una de las subrutinas implementadas para la construcción de éste, es necesario aclarar que durante el uso de la alternativa, no se recomienda transformar la tabla de usuario. Esto se debe a

que como el algoritmo de obtención de las funciones seno y coseno se implementó con aritmética entera, introduce un error, por lo que éste se verá reflejado en la manipulación posterior de la gráfica.



El listado completo de cada una de los subprogramas desarrollados, se puede consultar en el apéndice A de este trabajo, las características principales de cada uno de ellos son las siguientes.

\* NOMBRE: ROT.

FUNCION: Realiza la rotación de una determinada gráfica con respecto a un eje cualquiera y un ángulo de rotación proporcionado por el usuario.

DIRECCION: 2314H.

SUBROUTINAS QUE UTILIZA: REQANG, ROTUSU, ROTMAQ.

DATOS ENTRADA: Dirección inicial de tabla de máquina de archivo a modificarse en DIR1.

DATOS SALIDA: Tabla transformada partir de la dirección almacenada en DIR1.

REGISTROS QUE MODIFICA: A, BC, DE, HL, IY.

\* NOMBRE: REQANG.

FUNCION: Introduce desde el teclado el ángulo de rotación que proporciona el usuario. Calcula la función seno y coseno de éste y establece la pregunta acerca de modificar o no la tabla de usuario.

DIRECCION: 2336H.

SUBROUTINAS QUE UTILIZA: ESPA, ESMEN1, INFAC, GUADAS, CRLF, INCR, SENCOS.

DATOS ENTRADA: Ninguno.

DATOS SALIDA: Angulo de rotación en localidad GRAD. Valor de coseno y seno en SEN y COS. Mensajes en pantalla.

REGISTROS QUE MODIFICA: A, BC, DE, HL.

\* NOMBRE: INFAC (citada anteriormente).

DIRECCION: 223AH.

\* NOMBRE: GUADAS (citada anteriormente).

DIRECCION: 1BA6H.

\* NOMBRE: CRLF (citada anteriormente).

DIRECCION: C53AH.

\* NOMBRE: SENCOS.

FUNCION: Obtiene la función seno y coseno del ángulo de rotación. Trabaja en base a un algoritmo recursivo.

DIRECCION: 241FH.

SUBROUTINAS QUE UTILIZA: RECUR.

DATOS ENTRADA: Valor ángulo de rotación en localidad GRAD.

DATOS SALIDA: Valor de las funciones seno y coseno del ángulo de rotación en localidades SEN y COS respectivamente.

REGISTROS QUE MODIFICA: BC, DE, HL.

\* NOMBRE: RECUR.

FUNCION: Amplifica el valor del ángulo de rotación para poder implementar con aritmética entera el algoritmo para obtención de sen y cos.

DIRECCION:

SUBROUTINAS QUE UTILIZA: COMP2, CONV, DIV.

DATOS ENTRADA: Valor a amplificar en registro DE.

DATOS SALIDA: Valor amplificado en registro HL.

REGISTROS QUE MODIFICA: BC, DE, HL.

\* NOMBRE: COMP2 (citada anteriormente).

DIRECCION: 1B55H.

\* NOMBRE: CONV (citada anteriormente).

DIRECCION: 1BF8H.

\* NOMBRE: DIV (citada anteriormente).

DIRECCION: 1B5BH.

## CONCLUSIONES

Durante la concepción de la idea para el desarrollo del tema presentado en este trabajo, se plantearon una serie de objetivos que cubrían en forma general el temario presentado a lo largo de este texto. Ahora, al final de él, es necesario reconocer que los resultados obtenidos, así como el número de conocimientos adquiridos durante la implementación de estas ideas, es mucho mayor que el imaginado inicialmente.

Con esto, el número de conclusiones que se pueden vertir como resultado de la realización de este trabajo, sobrepasa a las que se pudieran exponer en esta sección, por lo que solo se citan las más relevantes.

Dentro de éstas, es indudable que la primera que debe ser tomada en consideración, es la existencia física del sistema graficador proyectado, ya que demuestra que en lo general se cumplió con la meta planteada.

El tener acceso real a este dispositivo, implica también otra serie de conclusiones particulares, que si bien están inmersas dentro de la anterior, es necesario hacer mención de ellas. Estas, se relacionan con dos campos distintos, el que abarca aspectos técnicos sobre la construcción del sistema y

el que se refiere a la vida personal.

Acerca del primero, se demuestra la factibilidad de interrelación entre los diversos ámbitos de la ciencia, tanto dentro de la misma electrónica, pues se combina la digital con la analógica, como la combinación de ésta con disciplinas

Adicionalmente, se genera una visión más amplia de las posibilidades de aplicación existentes actualmente para las microcomputadoras, observándose el gran potencial que éstas ofrecen para facilitar las tareas del ser humano, ya sea para fines científicos, como el caso del sistema estudiado, o para cuestiones cotidianas, como las aplicaciones con que hoy en día se cuenta.

En lo que se refiere al terreno personal, es indudable que el número de experiencias recogidas marca de una forma u otra, el comportamiento que un futuro se tenga. Conllevan además, la toma de conciencia de la capacidad que puede poseer el ser humano para desarrollar proyectos que faciliten la vida de seres de su misma especie.

Finalmente, se hace énfasis en que la tarea iniciada con este trabajo no está terminada, y se espera que alguien más comparta las mismas inquietudes y recoga las experiencias aquí vertidas para continuarla.

# APENDICE A

## Listados de programas.

El inicio de este apéndice, es para agradecer al Lic. Eduardo Virueña Silva el haberme permitido utilizar el ensamblador -- que él diseñó para el microprocesador Z-80, y también por haber accedido tan amablemente a modificar el formato de impresión del mismo para una mejor presentación de este trabajo.



APEND A

Sistema Graficador FAP-100

```

1. 0000 ;
2. 0000 ;*****
3. 0000 ;***
4. 0000 ;*** PROGRAMAS DE GRAFICACION ***
5. 0000 ;***
6. 0000 ;*****
7. 0000 ;
8. 0000 ;
9. 13A0 ; == $13A0
10. 13A0 DIRI = $110B
11. 13A0 BUFY = $110D
12. 13A0 BUFY = $110F
13. 13A0 PLUMA = $1110
14. 13A0 ;
15. 13A0 2A 0B 11 POS LD HL,(DIRI) ;Direcciona inicio tabla.
16. 13A3 D0 21 0D 11 LD IX,BUFY ;Direcciona buffer salida
17. 13A7 ; ;coordenada X.
18. 13A7 C0 7C 14 CALL PDATA ;Formatea coordenada X.
19. 13AA 23 INC HL ;Accesa coordenada Y.
20. 13AB D0 21 0F 11 LD IX,BUFY ;Direcciona buffer salida
21. 13AF ; ;coordenada Y.
22. 13AF C0 7C 14 CALL PDATA ;Formatea coordenada Y.
23. 13B2 23 INC HL ;Accesa palabra de control.
24. 13B3 C0 FA 13 CALL TRAPO ;Manda informacion hacia el
25. 13B6 ; ;graficador.
26. 13B6 ES PUSH HL ;Genera parametro para com-
27. 13B7 21 04 00 LD HL,$0004 ;mandar retardo.
28. 13BA C0 CC 14 CALL RPOS ;Da retardo para compensar
29. 13B0 ; ;inercia de graficador.
30. 13B0 E1 POP HL ;
31. 13BE C9 RET ;Termina.
32. 13BF D0 21 0D 11 CASA LD IX,BUFY ;Direcciona buffer salida
33. 13C3 ; ;coordenada X.
34. 13C3 C0 36 00 00 LD (IX+0),$00 ;Carga valor coordenada X
35. 13C7 D0 36 FF 00 LD (IX-1),$00 ;de punto CASA.
36. 13CB C0 21 0F 11 LD IX,BUFY ;Direcciona buffer salida
37. 13CF ; ;coordenada Y.
38. 13CF D0 36 00 00 LD (IX+0),$00 ;Carga valor coordenada Y
39. 13D3 C0 36 FF 00 LD (IX-1),$00 ;de punto CASA.
40. 13D7 C0 FA 13 CALL TRAPO ;Manda informacion hacia el
41. 13DA ; ;graficador.
42. 13DA C9 RET ;Termina.
43. 13DB D0 21 0F 11 TRANS LD IX,BUFY ;Direcciona buffer salida
44. 13DF ; ;coordenada Y.
45. 13DF C0 6F 14 CALL ACTY ;Activa canal Y interfaz.
46. 13E2 C0 2D 14 CALL SACA ;Manda informacion coord-
47. 13E5 ; ;nada Y con datos control.
48. 13E5 3E 05 LD A,$05 ;Da retardo para compensar
49. 13E7 C0 E3 14 CALL RTD01 ;movimiento de graficador.
50. 13EA D0 21 0D 11 LD IX,BUFY ;Direcciona buffer de salida

```



# TESIS COM. FALLAS DE ORIGEN

APEND A		Sistema Graficador FAP-100	
102.	144C 3E 05		LD A,\$05 ;Activa senal SEC.
103.	144E D3 03		OUT (<#03),A ;
104.	1450 3A 03 00		LD A,<#03> ;Habilita senal PRIN.
105.	1453 D3 03		OUT (<#03),A ;
106.	1455 DD 56 FF		LD D,<IX-1> ;Checa que dato no salga
107.	1458 CD A6 14		CALL RANG0 ;de area de graficado.
108.	145B D3 04		OUT (<#04),A ;Manda dato.
109.	145D 3E 07		LD A,\$07 ;Activa senal PRIN.
110.	145F D3 03		OUT (<#03),A ;
111.	1461 C9		RET ;Termina.
112.	1462 3A 10 11	ACTX	LD A,<PLUMA> ;Accesa informacion pluma.
113.	1465 CB 47		BIT 0,A ;Checa posicion pluma.
114.	1467 28 03		JR Z,ET1 ;Salta si pluma abajo.
115.	1469 3E 00		LD A,\$00 ;Genera palabra para selec-
116.	146B		cion canal X/pluma arriba.
117.	146B C9		RET ;Termina.
118.	146C 3E 01	ET1	LD A,\$01 ;Genera palabra para selec-
119.	146E		cion canal X/pluma abajo.
120.	146E C9		RET ;Termina.
121.	146F 3A 10 11	ACTY	LD A,<PLUMA> ;Accesa informacion pluma.
122.	1472 CB 47		BIT 0,A ;Checa posicion pluma.
123.	1474 28 03		JR Z,ET2 ;Salta si pluma abajo.
124.	1476 3E 02		LD A,\$02 ;Genera palabra para selec-
125.	1478		cion canal Y/pluma arriba.
126.	1478 C9		RET ;Termina.
127.	1479 3E 03	ET2	LD A,\$03 ;Genera palabra para selec-
128.	147B		cion canal Y/pluma abajo.
129.	147B C9		RET ;Termina.
130.	147C 7E	PDAT0	LD A,<HL> ;Toma primera parte dato.
131.	147D E6 3F		AND #3F ;Borra bits que no ocupa.
132.	147F DD 77 00		LD <IX+0>,A ;Almacena parte baja dato.
133.	1482 7E		LD A,<HL> ;Toma primera parte dato.
134.	1483 07		RLCA ;Prepara dos bits utiles
135.	1484 07		RLCA ;para parte alta de dato.
136.	1485 E6 03		AND #03 ;Borra bits que no ocupa.
137.	1487 23		INC HL ;Apunta parte alta de dato.
138.	1488 55		LD D,<HL> ;Toma parte alta de dato.
139.	1489 CB 22		SLA D ;Toma informacion del valor
140.	148B 30 04		JR NC,PDA1 ;del dato y la almacena
141.	148D DD CB 00 FE		SET 7,<IX+0> ;para reconocer si esta
142.	1491 CB 22	PDA1	SLA D ;dentro del rango de grafi-
143.	1493 30 04		JR NC,PDA2 ;cado. Genera espacio para
144.	1495 DD CB 00 F6		SET 6,<IX+0> ;los dos bits parte baja.
145.	1499 82	PDA2	ADD A,D ;Conjunta los dos bits de
146.	149A		parte baja con parte alta.
147.	149A 2F		CPL ;Invierte.
148.	149B DD 77 FF		LD <IX-1>,A ;Almacena en buffer salida.
149.	149E DD 7E 00		LD A,<IX+0> ;Toma parte baja dato.
150.	14A1 2F		CPL ;Invierte.
151.	14A2 DD 77 00		LD <IX+0>,A ;Almacena en buffer salida.
152.	14A5 C9		RET ;Termina.

APEHD A

Sistema Graficador FAP-100

153.	14A6	DD	CB	FF	76	RANGO	BIT 6,(IX-1)		;Checa rango parte alta de
154.	14AA	28	14				JR Z,RAN1		;dato, y salta si sale de
155.	14AC	DD	CB	FF	7E		BIT 7,(IX-1)		;este.
156.	14B0	28	0E				JR-Z,RAN1		;
157.	14B2	DD	CB	00	76		BIT 6,(IX+0)		;Checa rango parte baja de
158.	14B6	28	08				JR Z,RAN1		;dato, y salta si sale de
159.	14B8	DD	CB	00	7E		BIT 7,(IX+0)		;este.
160.	14BC	28	0B				JR-Z,NEGA		;
161.	14BE	7A					LD A,D		;Guarda si no sale de area.
162.	14BF	C9					RET		;Termina.
163.	14C0	DD	CB	00	7E	RAN1	BIT 7,(IX+0)		;Checa rango parte baja de
164.	14C4	28	03				JR Z,NEGA		;dato y salta si es negati-
165.	14C6								vo fuera de rango.
166.	14C6	3E	00				LD A,\$00		;Almacena valor maximo para
167.	14C8								positivo fuera de rango.
168.	14C8	C9					RET		;Termina.
169.	14C9	3E	FF			NEGA	LD A,\$FF		;Almacena valor maximo para
170.	14CB								negativo fuera de rango.
171.	14CB	C9					RET		;Termina.
172.	14CC	01	00	00		RPOS	LD BC,\$0000		;Inicia reg. comparador.
173.	14CF	28				ET3	DEC HL		;Decrementa contador.
174.	14D0	A7					AND A		;Compara para reconocer fin
175.	14D1	ED	42				SBC HL,BC		;de retardo.
176.	14D3	20	01				JR-NZ,ET4		;No termina.
177.	14D5	C9					RET		;Termina.
178.	14D6	E5				ET4	PUSH HL		;Guarda contador principal.
179.	14D7	21	00	00			LD HL,\$0000		;Inicia cont. secundario.
180.	14DA	28				ET5	DEC HL		;Decrementa contador sec.
181.	14DB	A7					AND A		;Checa fin conteo sec.
182.	14DC	ED	42				SBC HL,BC		;
183.	14DE	20	FA				JR-NZ,ET5		;No termina conteo sec.
184.	14E0	E1					POP HL		;Recupera cont. principal.
185.	14E1	18	EC				JR ET3		;Termina conteo secundario.
186.	14E3	47				RTD01	LD B,A		;Copia parametro entrada.
187.	14E4	05				ET6	DEC B		;Decrementa contador.
188.	14E5	28	07				JR Z,ET6		;Salta si termina conteo.
189.	14E7	0E	00				LD C,\$00		;Inicia conteo secundario.
190.	14E9	0D				ET7	DEC C		;Decrementa contador sec.
191.	14EA	20	FD				JR-NZ,ET7		;Salta si no termina conteo
192.	14EC								;secundario.
193.	14EC	18	F6				JR ET6		;Retoma conteo principal.
194.	14EE	C9				ET8	RET		;Termina.
195.	14EF	47				RTD0	LD B,A		;Copia parametro entrada.
196.	14F0	05				ET9	DEC B		;Decrementa contador.
197.	14F1	20	FD				JR-NZ,ET9		;Salta si no termina RTD01.
198.	14F3	C9					RET		;Termina.
198.	14F4								

T a b l a d e S i m b o l o s

Nombre	Valor	Linea de Definicion
dir1	110B	10
bufx	110D	11
bufy	110F	12
pluma	1110	13
pos	13A0	15
casa	13BF	32

trans	13DB	43
trapo	13FA	58
sacal	1417	73
saca	142D	86
manda	143B	95
actx	1462	112
et1	146C	118
acty	146F	121
et2	1479	127
pdato	147C	130
pda1	1491	142
pda2	1499	145
rango	14A6	153
ran1	14C0	163
nega	14C9	169
rpos	14CC	172
et3	14CF	173
et4	14D6	178
et5	14DA	180
rtdol	14E3	186
et6	14E4	187
et7	14E9	190
et8	14EE	194
rtd0	14EF	195
et9	14F0	196

APEND A

Sistema Graficador FAP-100

```

1. 0000 ;
2. 0000 ;*****
3. 0000 ;***
4. 0000 ;*** PROGRAMAS GENERADORES DE DATOS PARA GRAFICADO ***
5. 0000 ;***
6. 0000 ;*****
7. 0000 ;
8. 14F4 ; *= $14F4
9. 14F4 POS = $13A0
10. 14F4 TRANS = $13DB
11. 14F4 PDATO = $147C
12. 14F4 CASA = $13BF
13. 14F4 BRX = $111C
14. 14F4 BRV = $111E
15. 14F4 DIS = $1120
16. 14F4 OCT = $111B
17. 14F4 BUFX = $110D
18. 14F4 BUFY = $110F
19. 14F4 BUF = $1111
20. 14F4 RAD = $1122
21. 14F4 PLUMA = $1110
22. 14F4 ABSX = $1117
23. 14F4 ABSY = $1119
24. 14F4 DIFX = $1113
25. 14F4 DIFY = $1115
26. 14F4 ;
27. 14F4 ;
28. 14F4 CD A0 13 GRAF12 CALL POS ;Posiciona pluma.
29. 14F7 7E GUARDA LD A,<HL> ;Toma palabra de control.
30. 14F8 32 10 11 LD <PLUMA>,A ;Almacena palabra control.
31. 14FB E5 PUSH HL ;Guarda direccion control.
32. 14FC 2B DEC HL ;Apunta a la coordenada X
33. 14FD 2B DEC HL ;del primer punto que forma
34. 14FE 2B DEC HL ;la pareja a unirse.
35. 14FF 2B DEC HL ;
36. 1500 E5 PUSH HL ;Copia la direccion de este
37. 1501 FD E1 POP IY ;dato.
38. 1503 CB 4F BIT I,A ;Checa si se trata de recta
39. 1505 ; ;o circulo.
40. 1505 28 12 JR Z,LINEA ;Salta si es linea.
41. 1507 CD 3D 16 CALL BREC12 ;Llama subrutina algoritmo
42. 150A ; ;de circulos.
43. 150A CD 4F 15 CALL RE15B7 ;Direcciona siguiente dato
44. 150D ; ;y pregunta fin de tabla.
45. 150D 20 1B JR NZ,FINAL ;Salta si fin de archivo.
46. 150F 7E LD A,<HL> ;Toma palabra de control.
47. 1510 32 10 11 LD <PLUMA>,A ;Almacena palabra control.
48. 1513 E5 PUSH HL ;Guarda direccion control.
49. 1514 CD 58 15 CALL DIF2 ;Obtiene parametros tomando
50. 1517 ; ;en cuenta que es despues de

```

```

51. 1517 ; ;El trazo de un circulo.
52. 1517 18 03 ; JR CIRLIN ;Salta a unir con recta
53. 1519 ; ;final del circulo con si-
54. 1519 ; ;guiente elemento.
55. 1519 CD 0E 15 LINEA CALL DIFI ;Obtiene parametros tomando
56. 151C ; ;en cuenta que se trazo
57. 151C ; ;linea.
58. 151C CD C4 15 CIRLIN CALL DETO ;Define octante para linea.
59. 151F CD 2E 15 CALL ASIG ;Inicializa variables en
60. 1522 ; ;base al octante definido.
61. 1522 CD C9 17 ; CALL BREL12 ;Realiza el trazo de recta.
62. 1525 CD 4F 15 RETOMA CALL RE15B7 ;Direcciona siguiente dato
63. 1528 ; ;y pregunta fin de tabla.
64. 1528 28 CD ; JR 2, GUARDA ;Salta si no fin de tabla.
65. 152A CD BF 13 FINAL CALL CASA ;Termina y posiciona pluma
66. 152D ; ;en punto casa.
67. 152D C9 ; RET ;
68. 152E 3A 1B 11 ASIG LD A,<OCT> ;Toma palabra que define
69. 1531 ; ;octante.
70. 1531 CB 47 ; BIT 0,A ;Checa en que eje se da el
71. 1533 ; ;movimiento principal.
72. 1533 20 0D ; JR NZ,AS1 ;Salta si mov. princ. en Y.
73. 1535 2A 19 11 LD HL,<ABSY> ;Realiza asignacion de
74. 1538 22 1C 11 LD <BRX>,HL ;contadores de etapas en
75. 153B 2A 17 11 LD HL,<ABSX> ;forma cruzada para eje X.
76. 153E 22 1E 11 LD <BRY>,HL ;
77. 1541 C9 ; RET ;Termina.
78. 1542 2A 17 11 AS1 LD HL,<ABSX> ;Realiza asignacion de
79. 1545 22 1C 11 LD <BRX>,HL ;contadores de etapas en
80. 1548 2A 19 11 LD HL,<ABSY> ;forma directa para eje Y.
81. 154B 22 1E 11 LD <BRY>,HL ;
82. 154E C9 ; RET ;Termina.
83. 154F D1 RE15B7 POP DE ;Guarda direccion subrutina
84. 1550 ; ;que la invoco.
85. 1550 E1 POP HL ;Toma direccion control.
86. 1551 D5 PUSH DE ;Guarda direccion subrutina
87. 1552 ; ;que la invoco.
88. 1552 0E 05 LD C,#05 ;Adiciona cinco unidades a
89. 1554 09 ADD HL,BC ;esta direccion y accesa
90. 1555 ; ;siguiente par de puntos.
91. 1555 CB 7E BIT 7,<HL> ;Checa si es fin de tabla.
92. 1557 C9 ; RET ;Termina.
93. 1558 01 00 00 DIF2 LD BC,#0000 ;Prepara reg. comparador.
94. 155B FD 6E 0A LD L,<IY+10> ;Toma coordenada X punto
95. 155E FD 66 0B LD H,<IY+11> ;extremo final.
96. 1561 FD 5E 00 LD E,<IY+0> ;Toma coordenada X punto
97. 1564 FD 56 01 LD D,<IY+1> ;extremo inicial.
98. 1567 A7 AND A ;Encuentra diferencia en
99. 1568 ED 52 SBC HL,DE ;eje X.
100. 156A 22 13 11 LD <DIFX>,HL ;Almacena diferencia.
101. 156D CD 17 16 CALL ABS ;Obtiene valor absoluto.

```

APEND A

Sistema Graficador FAP-100

102.	1570	22 17 11		LD <ABSX>,HL	;Almacena valor absoluto.
103.	1573	FD 6E 0C		LD L,<IY+2>	;Toma coordenada Y punto
104.	1576	FD 66 0D		LD H,<IY+3>	;extremo final.
105.	1579	FD 5E 02		LD E,<IY+2>	;Toma coordenada Y punto
106.	157C	FD 56 03		LD D,<IY+3>	;extremo inicial.
107.	157F	A7		AND A	;Encuentra diferencia en
108.	1580	ED 52		SBC HL,DE	;eje Y.
109.	1582	22 15 11		LD <DIFY>,HL	;Almacena diferencia.
110.	1585	CD 17 16		CALL ABS	;Obtiene valor absoluto.
111.	1588	22 19 11		LD <ABSY>,HL	;Almacena valor absoluto.
112.	158B	3E 00		LD A,\$00	;Inicializa palabra de con-
113.	158D				;trol que define octante.
114.	158D	C9		RET	;Termina.
115.	158E	01 00 00	DIF1	LD BC,\$0000	;Prepara reg. comparador.
116.	1591	FD 6E 05		LD L,<IY+5>	;Toma coordenada X punto
117.	1594	FD 66 06		LD H,<IY+6>	;extremo final.
118.	1597	FD 5E 00		LD E,<IY+0>	;Toma coordenada X punto
119.	159A	FD 56 01		LD D,<IY+1>	;extremo inicial.
120.	159D	A7		AND A	;Encuentra diferencia en
121.	159E	ED 52		SBC HL,DE	;eje X.
122.	15A0	22 13 11		LD <DIFX>,HL	;Almacena diferencia.
123.	15A3	CD 17 16		CALL ABS	;Obtiene valor absoluto.
124.	15A6	22 17 11		LD <ABSX>,HL	;Almacena valor absoluto.
125.	15A9	FD 6E 07		LD L,<IY+7>	;Toma coordenada Y punto
126.	15AC	FD 66 08		LD H,<IY+8>	;extremo final.
127.	15AF	FD 5E 02		LD E,<IY+2>	;Toma coordenada Y punto
128.	15B2	FD 56 03		LD D,<IY+3>	;extremo inicial.
129.	15B5	A7		AND A	;Encuentra diferencia en
130.	15B6	ED 52		SBC HL,DE	;eje Y.
131.	15B8	22 15 11		LD <DIFY>,HL	;Almacena diferencia.
132.	15BB	CD 17 16		CALL ABS	;Encuentra valor absoluto.
133.	15BE	22 19 11		LD <ABSY>,HL	;Almacena valor absoluto.
134.	15C1	3E 00		LD A,\$00	;Inicializa palabra de con-
135.	15C3				;trol que define octante.
136.	15C3	C9		RET	;Termina.
137.	15C4	CD 36 16	DETO	CALL EVAY	;Evalua diferencia eje Y.
138.	15C7	20 13		JR NZ,DET2	;Salta si no es cero.
139.	15C9	CD 2F 16		CALL EVAX	;Evalua diferencia eje X.
140.	15CC	CA 25 15		JP Z;RETOMA	;Regresa programa princi-
141.	15CF				;pal ya que es mov. nulo.
142.	15CF	F2 07 15		JP P,DET1	;Salta si mayor que cero.
143.	15D2	3E 05		LD A,\$05	;Genera palabra que define
144.	15D4	C3 13 16		JP CARGA	;cuarto octante.
145.	15D7	3E 03	DET1	LD A,\$03	;Genera palabra que define
146.	15D9	C3 13 16		JP CARGA	;octavo octante.
147.	15DC	CD 2F 16	DET2	CALL EVAX	;Evalua diferencia eje X.
148.	15DF	20 0E		JR NZ,DET4	;Salta si no es cero.
149.	15E1	CD 36 16		CALL EVAY	;Evalua diferencia eje Y.
150.	15E4	F2 EA 15		JP P,DET3	;Salta si mayor que cero.
151.	15E7	C3 13 16		JP CARGA	;Almacena palabra que co-
152.	15EA				;rresponde a sexto octante.



APEND A

Sistema Graficador FAP-100

153.	15EA	3E	06	DET3	LD A,#06	;Genera palabra que define
154.	15EC	C3	13		JP CARGA	;segundo octante.
155.	15EF	FA	F4	15	DET4	JP M,DET5
156.	15F2	CB	CF		SET 1,A	;Define bit 1 para x mayor
157.	15F4	CD	36	16	DET5	CALL EVAY
158.	15F7	FA	FC	15		JP M,DET6
159.	15FA	CB	D7		SET 2,A	;Define bit 2 para Y mayor
160.	15FC	CD	24	16	DET6	CALL AXAY
161.	15FF					;Checa si las diferencias
162.	15FF	28	08		JP Z,XNOR	;en ambos ejes son iguales.
163.	1601					;Realiza op. nor exclu-
164.	1601	FA	06	16		;siva para bit cero.
165.	1604	CB	C7		JP M,DET7	;Define B0 dependiendo
166.	1606	C3	13	16	DET7	SET 0,A
167.	1609	C3	13	16	DET7	JP CARGA
168.	160A	4F			XNOR	PUSH BC
169.	160B	CB	3F			LD C,A
170.	160D	A9				SRL A
171.	160E	2F				XOR C
172.	160F	CB	3F			CPL
173.	1611	B1				SRL A
174.	1612	C1				OR C
175.	1613	32	1B	11	CARGA	POP BC
176.	1616	C9				LD <OCT>,A
177.	1617	CB	7C		ABS	RET
178.	1619	28	08			BIT 7,H
179.	161B	E5				JP Z,ET10
180.	161C	D1				PUSH HL
181.	161D	21	00	00		POP DE
182.	1620	A7				LD HL,#0000
183.	1621	ED	52			AND A
184.	1623	C9			ET10	SBC HL,DE
185.	1624	2A	17	11	AXAY	RET
186.	1627	ED	5B	19	11	LD HL,<ABSX>
187.	162B	A7				LD DE,<ABSY>
188.	162C	ED	52			AND A
189.	162E	C9				SBC HL,DE
190.	162F	2A	13	11	EVAX	RET
191.	1632	A7				LD HL,<DIFX>
192.	1633	ED	42			AND A
193.	1635	C9				SBC HL,BC
194.	1636	2A	15	11	EVAY	RET
195.	1639	A7				LD HL,<DIFY>
196.	163A	ED	42			AND A
197.	163C	C9				SBC HL,BC
198.	163D	01	00	00	BREC12	RET
199.	1640	3E	06			LD BC,#0000
200.	1642	32	1B	11		LD A,#06
201.	1645	FD	6E	02	BC1	LD <OCT>,A
202.	1648	FD	66	03		LD L,<IY+2>
203.	164B	FD	5E	07		LD H,<IY+3>
						LD E,<IY+7>

APEND A

Sistema Graficador FAP-100

204.	164E	FD	56	08		LD D,(Y+S)	;tro.
205.	1651	A7				AND A	;Obtiene valor del radio.
206.	1652	ED	52			SBC HL,DE	;
207.	1654	22	22	11		LD <RAD>,HL	;Almacena valor radio.
208.	1657	22	1E	11		LD  ,HL	;Inicializa contador eta-
209.	165A						;pas eje Y.
210.	165A	ED	43	1C	11	LD <BRX>,BC	;Inicializa contador eta-
211.	165E						;pas eje X.
212.	165E	29				ADD HL,HL	;Obtiene discriminante
213.	165F	E5				PUSH HL	;inicial:
214.	1660	D1				POP DE	; D = 3 - 2R
215.	1661	21	03	00		LD HL,#0003	;
216.	1664	A7				AND A	;
217.	1665	ED	52			SBC HL,DE	;
218.	1667	22	20	11		LD <DIS>,HL	;Almacena discriminante.
219.	166A	CD	52	17	BC2	CALL X1MY1	;Checa si es fin de trazo
220.	1660						;en octante par.
221.	1660	F2	A4	16		JP P,BC5	;Salta si es fin de trazo.
222.	1670	CD	DB	13		CALL TRANS	;Manda datos a graficador.
223.	1673	CD	64	17		CALL EVADIS	;Evalua discriminante.
224.	1676	FA	8E	16		JP M,BC3	;Elige punto S1.
225.	1679	CD	52	17		CALL X1MY1	;Elige punto T1 y calcula
226.	167C	CD	6B	17		CALL SUMA4	;nuevo discriminante:
227.	167F	11	0A	00		LD DE,#000A	; D = D + 4(X-Y) + 10
228.	1682	19				ADD HL,DE	;
229.	1683	22	20	11		LD <DIS>,HL	;Almacena discriminante.
230.	1686	CD	73	17		CALL Y11	;Decrementa contador eje Y.
231.	1689	CD	83	17		CALL MOV2	;Actualiza coordenada para
232.	168C						;movimiento eje Y.
233.	168C	18	0D			JR BC4	;
234.	168E	2A	1C	11	BC3	LD HL,<BRX>	;Calcula nuevo discrimi-
235.	1691	CD	6B	17		CALL SUMA4	;nante:
236.	1694	11	06	00		LD DE,#0006	; D = D + 4X + 6
237.	1697	19				ADD HL,DE	;
238.	1698	22	20	11		LD <DIS>,HL	;Almacena discriminante.
239.	169B	CD	7B	17	BC4	CALL X11	;Incrementa contador eje X.
240.	169E	CD	9A	17		CALL MOV1	;Actualiza coordenada para
241.	16A1						;movimiento eje X.
242.	16A1	C3	6A	16		JP BC2	;Salta para checar fin de
243.	16A4						;trazo octante par.
244.	16A4	CD	DB	13	BC5	CALL TRANS	;Manda datos a graficador.
245.	16A7	CD	0A	17		CALL DEFCON	;Define control octante
246.	16AA						;impar.
247.	16AA	29				ADD HL,HL	;Calcula discriminante
248.	16AB	E5				PUSH HL	;inicial:
249.	16AC	D1				POP DE	; D = 3 - 2X
250.	16AD	21	03	00		LD HL,#0003	;
251.	16B0	A7				AND A	;
252.	16B1	ED	52			SBC HL,DE	;
253.	16B3	22	20	11		LD <DIS>,HL	;Almacena discriminante.
254.	16B6	CD	5D	17	BC6	CALL Y1CERO	;Checa fin de trazo en

APEHD A

Sistema Graficador FAP-100

255.	16B9				JP Z,BC9	;octante impar.
256.	16B9	CA	FB	16	CALL TRANS	;Salta si es fin de trazo.
257.	16BC	CD	DB	13	CALL EVADIS	;Manda datos a graficador.
258.	16BF	CD	64	17		;Evalua discriminante.
259.	16C2	FA	E5	16	JP M,BC8	;Elige Ti.
260.	16C5	CA	E5	16	JP Z,BC8	
261.	16C8	2A	1E	11	LD HL,(BRY)	;Elige Ri y calcula nuevo
262.	16CB	29			ADD HL,HL	;discriminante:
263.	16CC	29			ADD HL,HL	; D = D - 4Y + 6
264.	16CD	E5			PUSH HL	
265.	16CE	D1			POP DE	
266.	16CF	2A	20	11	LD HL,(DIS)	
267.	16D2	A7			AND A	
268.	16D3	ED	52		SBC HL,DE	
269.	16D5	11	06	00	LD DE,\$0006	
270.	16D8	19			ADD HL,DE	
271.	16D9	22	20	11	LD (DIS),HL	;Almacena discriminante.
272.	16DC	CD	73	17	CALL Y11	;Decrementa contador eje Y.
273.	16DF	CD	9A	17	CALL MOV1	;Actualiza coordenada para
274.	16E2					;movimiento eje Y.
275.	16E2	C3	B6	16	JP BC6	;Checa fin de trazo.
276.	16E5	CD	52	17	CALL X1MY1	;Calcula nuevo discrimi-
277.	16E8	CD	6B	17	CALL SUMA4	;ninante:
278.	16E9	11	0A	00	LD DE,\$000A	; D = D + 4(X-Y) + 10
279.	16EE	19			ADD HL,DE	
280.	16EF	22	20	11	LD (DIS),HL	;Almacena discriminante.
281.	16F2	CD	7B	17	CALL X11	;Incrementa contador eje X.
282.	16F5	CD	83	17	CALL MOV2	;Actualiza coordenada para
283.	16F8					;movimiento eje X.
284.	16F8	C3	DC	16	JP BC7	
285.	16FB	CD	0B	13	CALL TRANS	;Manda datos a graficador.
286.	16FE	CD	0A	17	CALL DEFCON	;Define palabra de control
287.	1701					;para nuevo octante.
288.	1701	3A	1B	11	LD A,(OCT)	;Checa fin de circulo.
289.	1704	FE	06		CP #06	
290.	1706	C2	45	16	JP NZ,BC1	;Inicia nuevo cuadrante.
291.	1709	C9			RET	;Termina.
292.	170A	3A	1B	11	LD A,(OCT)	;Toma control anterior.
293.	170D	FE	06		CP #06	;Checa inicio 2o. octante.
294.	170F	20	06		JR NZ,DEF1	;No inicia 2o. octante.
295.	1711	3E	0A		LD A,\$0A	;Genera control 2o. oct.
296.	1713	32	1B	11	LD (OCT),A	;Guarda control.
297.	1716	C9			RET	;Termina.
298.	1717	FE	0A		CP #0A	;Checa inicio 8o. octante.
299.	1719	20	05		JR NZ,DEF2	;No inicia 8o. octante.
300.	171B	3E	08		LD A,\$08	;Genera control 8o. oct.
301.	171D	C3	13	17	JR CONTA	;Guarda control.
302.	1720	FE	08		CP #08	;Checa inicio 7o. octante.
303.	1722	20	05		JR NZ,DEF3	;No inicia 7o. octante.
304.	1724	3E	04		LD A,\$04	;Genera control 7o. oct.
305.	1726	C3	13	17	JR CONTA	;Guarda control.

APEND A

Sistema Graficador FAP-100

306.	1729	FE 04	DEF3	CP \$04	;Checa inicio 60. octante.
307.	172B	20 05		JR NZ,DEF4	;No inicia 60. octante.
308.	172D	3E 05		LD A,\$05	;Genera control 60. oct.
309.	172F	C3 13 17		JP CONTA	;Guarda control.
310.	1732	FE 05	DEF4	CP \$05	;Checa inicio 50. octante.
311.	1734	20 05		JR NZ,DEF5	;No inicia 50. octante.
312.	1736	3E 09		LD A,\$09	;Genera control 50. oct.
313.	1738	C3 13 17		JP CONTA	;Guarda control.
314.	173B	FE 09	DEF5	CP \$09	;Checa inicio 40. octante.
315.	173D	20 05		JR NZ,DEF6	;No inicia 40. octante.
316.	173F	3E 0B		LD A,\$0B	;Genera control 40. oct.
317.	1741	C3 13 17		JP CONTA	;Guarda control.
318.	1744	FE 0B	DEF6	CP \$0B	;Checa inicio 3er. octante.
319.	1746	20 05		JR NZ,DEF7	;No inicia 3er. octante.
320.	1748	3E 07		LD A,\$07	;Genera control 3er. oct.
321.	174A	C3 13 17		JP CONTA	;Guarda control.
322.	174D	3E 06	DEF7	LD A,\$06	;Genera control fin de
323.	174F				;trazo.
324.	174F	C3 13 17		JP CONTA	;Guarda control.
325.	1752	2A 1C 11	X1NY1	LD HL,(BRX)	;Toma diferencia eje X.
326.	1755	ED 5B 1E 11		LD DE,(BRY)	;Toma diferencia eje Y.
327.	1759	A7		AND A	;Realiza comparacion.
328.	175A	ED 52		SBC HL,DE	
329.	175C	C9		RET	;Termina.
330.	175D	2A 1E 11	YICERO	LD HL,(BRY)	;Toma contador eje Y.
331.	1760	A7		AND A	;Realiza comparacion con
332.	1761	ED 42		SBC HL,BC	;cero.
333.	1763	C9		RET	;Termina.
334.	1764	2A 20 11	EVADIS	LD HL,(DIS)	;Toma valor discriminante.
335.	1767	A7		AND A	;Realiza comparacion con
336.	1768	ED 42		SBC HL,BC	;cero.
337.	176A	C9		RET	;Termina.
338.	176B	29	SUMA4	ADD HL,HL	;Multiplica por 2.
339.	176C	29		ADD HL,HL	;Multiplica por 4.
340.	176D	ED 5B 20 11		LD DE,(DIS)	;Toma valor discriminante.
341.	1771	19		ADD HL,DE	;Suma discriminante mas 4
342.	1772				;veces el contenido de HL.
343.	1772	C9		RET	;Termina.
344.	1773	2A 1E 11	Y11	LD HL,(BRY)	;Toma contador eje Y.
345.	1776	2B		DEC HL	;Decrementa contador.
346.	1777	22 1E 11		LD (BRY),HL	;Guarda nuevo contador.
347.	177A	C9		RET	;Termina.
348.	177B	2A 1C 11	X11	LD HL,(BRX)	;Toma contador eje X.
349.	177E	23		TNC HL	;Incrementa contador.
350.	177F	22 1C 11		LD (BRX),HL	;Guarda nuevo contador.
351.	1782	C9		RET	;Termina.
352.	1783	3A 1B 11	MOV2	LD A,(OCT)	;Toma palabra octante.
353.	1786	CB 5F		BIT 3,A	;Checa en que eje es el
354.	1788				;movimiento principal.
355.	1788	28 08		JR Z,ET11	;Salta si es eje Y.
356.	178A	DD 21 0D 11		LD IX,BUFx	;Toma coordenada X.

APEND A

Sistema Graficador FAP-100

357.	179E	CD	BD	17		CALL BIT1	
358.	1791						;Modifica coordenada en
359.	1791						;base al estado de B1 de
360.	1791	C9				RET	;palabra octante.
361.	1792	DD	21	0F	11	ET11	;Termina.
362.	1796	CD	B1	17		LD IX,BUFY	;Toma coordenada Y.
363.	1799					CALL BIT0	;Modifica coordenada en
364.	1799						;base al estado de B0 de
365.	1799	C9				RET	;palabra octante.
366.	179A	3A	1B	11		MOV1	;Termina.
367.	179D	CB	57			LD A,(OCT)	;Toma palabra octante.
368.	179F					BIT 2,A	;Checa en que eje es el
369.	179F	28	08			JR Z,ET12	;movimiento auxiliar.
370.	17A1	DD	21	0D	11		;Salta si es eje Y.
371.	17A5	CD	BD	17		LD IX,BUFK	;Toma coordenada X.
372.	17A8					CALL BIT1	;Modifica coordenada en
373.	17A8						;base al estado de B1 de
374.	17A8	C9					;palabra octante.
375.	17A9	DD	21	0F	11	ET12	;Termina.
376.	17AD	CD	B1	17		LD IX,BUFY	;Toma coordenada Y.
377.	17B0					CALL BIT0	;Modifica coordenada en
378.	17B0						;base al estado de B0 de
379.	17B0	C9					;palabra octante.
380.	17B1	C8	47			BIT0	;Termina.
381.	17B3	28	04			BIT 0,A	;Checa sentido mov. eje Y.
382.	17B5	CD	7B	18		JR Z,ET13	;Salta si es negativo.
383.	17B8	C9				CALL INCD	;Prepara mov. positivo.
384.	17B9	CD	88	18		RET	;Termina.
385.	17BC	C9				CALL DECD	;Prepara mov. negativo.
386.	17BD	C8	4F			RET	;Termina.
387.	17BF	28	04			BIT 1,A	;Checa sentido mov. eje X.
388.	17C1	CD	7B	18		JR Z,ET14	;Salta si es negativo.
389.	17C4	C9				CALL INCD	;Prepara mov. positivo.
390.	17C5	CD	88	18		RET	;Termina.
391.	17C8	C9				CALL DECD	;Prepara mov. negativo.
392.	17C9	01	00	00		RET	;Termina.
393.	17CC	2A	1E	11		BREL12	;Prepara reg. comparador.
394.	17CF	29				LD HL,(BRY)	;Calcula discriminante
395.	17D0	ED	5B	1C	11	ADD HL,HL	;inicial:
396.	17D4	AD				LD DE,(BRX)	; D = 2Y - X
397.	17D5	ED	52			AND A	;
398.	17D7	22	20	11		SBC HL,DE	;
399.	17DA	CA	19	18		LD (DIS),HL	;Almacena discriminante.
400.	17DD	FA	19	18		JP Z,BL2	;Elige punto Si.
401.	17E0	CD	39	18		JP M,BL2	;
402.	17E3					CALL DIS1	;Actualiza coordenadas
403.	17E3	CD	DB	13			;para acceder punto Ti.
404.	17E6	2A	1E	11		CALL TRANS	;Manda datos a graficador.
405.	17E9	29				LD HL,(BRY)	;Calcula nuevo discrimi-
406.	17EA	ED	5B	1C	11	ADD HL,HL	;nante;
407.	17EE	AD				LD DE,(BRX)	; D = D + 2Y - 2X
						AND A	;

APEPEND A

Sistema Graficador FAP-100

408.	17EF	ED	52		SBC HL,DE	
409.	17F1	A7			AND A	
410.	17F2	ED	52		SBC HL,DE	
411.	17F4	ED	5B 20 11		LD DE,<DIS>	
412.	17F8	19			ADD HL,DE	
413.	17F9	22	20 11		LD <DIS>,HL	;Almacena discriminante.
414.	17FC	2A	1C 11		LD HL,<BRX>	;Actualiza contador para
415.	17FF	2B			DEC HL	;eje X.
416.	1800	22	1C 11		LD <BRX>,HL	
417.	1803	A7			AND A	;Checa fin de trazo.
418.	1804	ED	42		SBC HL,BC	
419.	180E	CA	38 18		JP Z,FIN	;Termina si DX es cero.
420.	1809	2A	1E 11		LD HL,<BRY>	;No es fin. Actualiza
421.	180C	2B			DEC HL	;contador para eje Y.
422.	180D	22	1E 11		LD <BRY>,HL	
423.	1810	2A	20 11	EVAL	LD HL,<DIS>	;Evalua valor discrimi-
424.	1813	A7			AND A	;minante.
425.	1814	ED	42		SBC HL,BC	
426.	1816	C3	DA 17		JP BL1	;Prosigue trazo.
427.	1819	CD	5C 18	BL2	CALL DIS3	;Actualiza coordenadas
428.	181C					;para acceder punto Si.
429.	181C	CD	DB 13		CALL TRANS	;Manda datos a graficador.
430.	181F	2A	1E 11		LD HL,<BRY>	;Calcula nuevo discrimi-
431.	1822	29			ADD HL,HL	;nante.
432.	1823	ED	5B 20 11		LD DE,<DIS>	; D = D + 2Y
433.	1827	19			ADD HL,DE	
434.	1828	22	20 11		LD <DIS>,HL	;Almacena discriminante.
435.	182B	2A	1C 11		LD HL,<BRX>	;Actualiza contador eje
436.	182E	2B			DEC HL	;X.
437.	182F	22	1C 11		LD <BRX>,HL	
438.	1832	A7			AND A	;Checa fin de trazo.
439.	1833	ED	42		SBC HL,BC	
440.	1835	C2	10 18		JP NZ,EVAL	;No es fin y prosigue.
441.	1838	C9		FIN	RET	;Termina.
442.	1839	3A	18 11	DIS1	LD A,<OCT>	;Toma palabra octante.
443.	183C	DD	21 0D 11		LD IX,BUFX	;Direcciona buffer salida
444.	1840					;coordenada X.
445.	1840	CB	4F		BIT 1,A	;Checa sentido mov. X.
446.	1842	20	05		JR NZ,D1	;Movimiento positivo.
447.	1844	CD	88 18		CALL DECD	;Mov. negativo, decrementa
448.	1847					;coordenada.
449.	1847	18	03		JR D2	;Salta y modifica coord.Y.
450.	1849	CD	7B 18	D1	CALL INCD	;Incrementa coordenada.
451.	184C	DD	21 0F 11	D2	LD IX,BUFY	;Direcciona buffer salida
452.	1850					;coordenada Y.
453.	1850	CB	57		BIT 2,A	;Checa sentido mov. Y.
454.	1852	20	04		JR NZ,D3	;Movimiento positivo.
455.	1854	CD	88 18		CALL DECD	;Mov. negativo, decrementa
456.	1857					;coordenada.
457.	1857	C9			RET	;Termina.
458.	1858	CD	78 18	D3	CALL INCD	;Incrementa coordenada.

APEND A

Sistema Graficador FAP-100

459.	185B	C9			RET	;Termina.
460.	185C	3A 18 11	DIS3		LD A,(OCT)	;Toma palabra octante.
461.	185F	CB 47			BIT 0,A	;Checa eje para mov.
462.	1861					;principal.
463.	1861	20 10			JR NZ,D13	;Mov. en eje X.
464.	1863	DD 21 0F 11			LD IX,BUFY	;Direcciona buffer salida
465.	1867					;coordenada Y.
466.	1867	CB 57			BIT 2,A	;Checa sentido movimiento.
467.	1869	20 04	D11		JR NZ,D12	;Movimiento positivo.
468.	186B	CD 88 18			CALL DECD	;Mov. negativo, decrementa
469.	186E					;coordenada.
470.	186E	C9			RET	;Termina.
471.	186F	CD 78 18	D12		CALL INCD	;Incrementa coordenada.
472.	1872	C9			RET	;Termina.
473.	1873	DD 21 0D 11	D13		LD IX,BUFX	;Direcciona buffer salida
474.	1877					;coordenada X.
475.	1877	CB 4F			BIT 1,A	;Checa sentido movimiento.
476.	1879	18 EE			JR D11	;Actualiza coordenadas.
477.	187B	F5	INCD		PUSH AF	;Preserva contenido A y
478.	187C					;estatus.
479.	187C	CD 9D 18			CALL RDATA	;Transforma dato de for-
480.	187F					;mato de salida a formato
481.	187F					;de trabajo.
482.	187F	13			INC DE	;Incrementa dato.
483.	1880	CD 95 18			CALL PPDA	;Prepara dato para ser
484.	1883					;formateado.
485.	1883	CD 7C 14			CALL PDATA	;Formatea dato.
486.	1886	F1			POP AF	;Recupera informacion.
487.	1887	C9			RET	;Termina.
488.	1888	F5	DECD		PUSH AF	;Preserva contenido A y
489.	1889					;estatus.
490.	1889	CD 9D 18			CALL RDATA	;Transforma dato de for-
491.	188C					;mato de salida a formato
492.	188C					;de trabajo.
493.	188C	18			DEC DE	;Decrementa dato.
494.	188D	CD 95 18			CALL PPDA	;Prepara dato para ser
495.	1890					;formateado.
496.	1890	CD 7C 14			CALL PDATA	;Formatea dato.
497.	1893	F1			POP AF	;Recupera informacion.
498.	1894	C9			RET	;Termina.
499.	1895	ED 53 11 11	PPDA		LD <BUF>,DE	;Almacena dato en buffer
500.	1899					;temporal.
501.	1899	21 11 11			LD HL,BUF	;Apunta buffer temporal.
502.	189C	C9			RET	;Termina.
503.	189D	DD 7E FF	RDATA		LD A,<IX-1>	;Toma parte alta dato.
504.	18A0	0F			RRCA	;Posiciona dos bits que
505.	18A1	0F			RRCA	;pertenezen a parte baja
506.	18A2					;en formato de trabajo.
507.	18A2	E6 C0			AND \$C0	;Salva solo estos bits.
508.	18A4	5F			LD E,A	;Respalda valor de estos.
509.	18A5	DD 7E 00			LD A,<IX+0>	;Toma parte baja dato.

APEND A

Sistema Graficador FAP-100

510.	18A8	E6 3F	AND #3F	;Prepara espacio.
511.	18AA	83	ADD A,E	;Estructura primer byte.
512.	18AB	2F	CPL	;Obtiene primeros ocho
513.	18AC			;bits formato trabajo.
514.	18AC	5F	LD E,A	;Almacena este byte.
515.	18AD	DD 7E FF	LD A,(IX-1)	;Toma parte alta dato.
516.	18B0	CB 3F	SRL A	;Prepara espacio para
517.	18B2	CB 3F	SRL A	;dos bits almacenados
518.	18B4			;en parte baja.
519.	18B4	57	LD D,A	;Respalda dato.
520.	18B5	DD 7E 00	LD A,(IX+0)	;Toma parte baja dato.
521.	18B8	E6 C0	AND #C0	;Salva dos bits pertene-
522.	18BA			;cientes parte alta.
523.	18BA	82	ADD A,D	;Estructura segundos ocho
524.	18BB			;bits.
525.	18BB	2F	CPL	;Obtiene segundo byte
526.	18BC			;formato de trabajo.
527.	18BC	57	LD D,A	;Almacena segundo byte.
528.	18BD	C9	RET	;Termina.
528.	18BE			

T a b l a d e S i m b o l o s

Nombre	Valor	Linea de Definicion
pos	13A0	9
trans	13DB	10
pdato	147C	11
casa	13BF	12
brx	111C	13
bry	111E	14
dis	1120	15
oct	111B	16
bufx	110D	17
bufy	110F	18
buf	1111	19
rad	1122	20
pluma	1110	21
absx	1117	22
absy	1119	23
difx	1113	24
dify	1115	25
graf12	14F4	28
guarda	14F7	29
linea	1519	55
cirlin	151C	58
retoma	1525	62
final	152A	65
asig	152E	68
asl	1542	78
rel5b7	154F	83
dif2	1558	93
dif1	159E	115
deto	15C4	137
det1	15D7	145
det2	15DC	147
det3	15EA	153
det4	15EF	155



det5	15F4	157
det6	15FC	160
det7	1606	166
xnor	1609	167
carga	1613	175
abs	1617	177
et10	1623	184
axay	1624	185
evax	162F	190
evay	1636	194
brec12	163D	198
bc1	1645	201
bc2	166A	219
bc3	168E	234
bc4	169B	239
bc5	16A4	244
bc6	16B6	254
bc7	16DC	272
bc8	16E5	276
bc9	16FB	285
defcon	170A	292
conta	1713	296
def1	1717	298
def2	1720	302
def3	1729	306
def4	1732	310
def5	173B	314
def6	1744	318
def7	174D	322
xlmy1	1752	325
ylcero	175D	330
evadis	1764	334
suma4	176B	338
y11	1773	344
x11	177B	348
mov2	1783	352
et11	1792	361
mov1	179A	366
et12	17A9	375
bit0	17B1	380
et13	17B9	384
bit1	17BD	386
et14	17C5	390
brel12	17C9	392
b11	17DA	399
eval	1810	423
b12	1819	427
fin	1838	441
dis1	1839	442
d1	1849	450
d2	184C	451
d3	1858	458
dis3	185C	460
di1	1869	467
di2	186F	471
di3	1873	473
incd	187B	477
decd	1888	488
ppda	1895	499
rdata	189D	503

APEND A

Sistema Graficador FAP-100

```
1. 0000
2. 0000
3. 0000
4. 0000
5. 0000
6. 0000
7. 0000
8. 0000
9. 0000
10. 18D7
11. 18D7
12. 18D7
13. 18D7
14. 18D7
15. 18D7
16. 18D7
17. 18D7
18. 18D7
19. 18D7
20. 18D7
21. 18D7
22. 18D7
23. 18D7
24. 18D7
25. 18D7
26. 18D7
27. 18D7
28. 18D7
29. 18D7
30. 18D7
31. 18D7
32. 18D7
33. 18D7
34. 18D7
35. 18D7 3E 1A
36. 18D9 CD 3F 01
37. 18DC 21 00 08
38. 18DF 22 27 11
39. 18E2 22 29 11
40. 18E5 21 45 11
41. 18E8 CD BE 18
42. 18EB CD C4 19
43. 18EE
44. 18EE CD DD 19
45. 18F1 CD 09 1A
46. 18F4 3E 1A
47. 18F6 CD 3F 01
48. 18F9 CD 37 1A
49. 18FC 3E 21
50. 18FE 32 44 11
```

```
*****
;***          PROGRAMAS DE COMUNICACION CON USUARIO      ***
;***          *****
;
;***          G E N E R A C I O N   D E   T A B L A      ***
;
;          ** = $18D7
LET1      = $1145
LET2      = $115A
LET3      = $1160
LET4      = $1167
LET5      = $118A
LET6      = $119D
LET9      = $11CA
LET11     = $11E3
LETEC     = $0136
ESPA      = $013F
ESMEN     = $01DC
ESMEN1    = $18BE
ORGUSX    = $1127
ORGUSY    = $1129
ORCX      = $1135
ORCY      = $1137
RENPOS    = $1144
RENC      = $112B
DIVOR     = $112F
COL       = $1126
CONDIR    = $1124
DIR1      = $110B
;
;
GENTA     LD A,$1A          ;Manda comando para borrar
          CALL ESPA        ;pantalla.
          LD HL,$0800      ;Genera origen relativo que
          LD <ORGUSX>,HL   ;da de entrada.
          LD <ORGUSY>,HL   ;
          LD HL,LET1       ;Manda a pantalla mensaje
          CALL ESMEN1      ;"DIRECCION INICIAL: ".
          CALL LLENA       ;Prepara espacio para reci-
          ;bir dato de teclado.
          CALL INDIR       ;Recibe direccion inicial.
          CALL QUADIR      ;Almacena dir. inicial.
          LD A,$1A        ;Manda comando para borrar
          CALL ESPA        ;pantalla.
          CALL PLANT       ;Despliega plantilla guia.
          LD A,$21        ;Almacena numero de primer
          LD <RENPOS>,A    ; renglon en pantalla para
```

APEND A

Sistema Graficador FAP-100

51.	1901				LD HL,#0001	;generar tabla.
52.	1901	21	01	00	CALL ESMEH1	;Inicializa contador de renglones generados.
53.	1904	22	2B	11	LD BC,(RENG)	;Toma numero de renglon.
54.	1907	ED	4B	2B	CALL BIASC	;Realiza conversion a formato ASCII.
55.	190B	CD	4A	1A	LD HL,LET2	;Direcciona numero de renglon y lo despliega.
56.	190E				CALL ESMEH1	;Genera numero 1er. columna y recibe dato.
57.	190E	21	5A	11	LD A,#0B	;Genera numero 2a. columna y recibe dato.
58.	1911	CD	BE	1B	CALL DATO	;Genera numero 3er. columna y recibe control.
59.	1914	3E	0B		LD A,#24	;Checa si fin de tabla.
60.	1916	CD	B4	1A	CALL DATO	;Salta si es fin de tabla.
61.	1919	3E	16		LD A,#2D	;Genera numero 4a. columna y recibe dato.
62.	191B	CD	B4	1A	CALL DATO	;Genera numero 5a. columna y recibe dato.
63.	191E	3E	24		LD A,#38	;Genera numero 6a. columna y recibe control.
64.	1920	CD	C2	1A	CALL CONTROL	;Checa si fin de tabla.
65.	1923	CD	D0	1A	CALL CHECA	;Salta si es fin de tabla.
66.	1926	20	35		JR NZ,FIN	;Genera numero 4a. columna y recibe dato.
67.	1928	3E	2D		LD A,#2D	;Genera numero 5a. columna y recibe dato.
68.	192A	CD	B4	1A	CALL DATO	;Genera numero 6a. columna y recibe control.
69.	192D	3E	38		LD A,#46	;Checa si fin de tabla.
70.	192F	CD	B4	1A	CALL CONTROL	;Salta si es fin de tabla.
71.	1932	3E	46		LD A,#46	;Accesa siguiente renglon.
72.	1934	CD	C2	1A	CALL CHECA	;Incrementa numero de renglon en pantalla.
73.	1937	CD	D0	1A	CALL CONTROL	;Incrementa numero de renglon generado.
74.	193A	20	21		JR NZ,FIN	;Checa si la pantalla esta llena.
75.	193C	CD	3A	C5	CALL CRLF	;Manda comando para posiciona cursor en "home".
76.	193F	CD	01	C4	CALL INCR	;Despliega plantilla guia.
77.	1942				CALL INCR	;Posiciona cursor en primera columna de pantalla.
78.	1942	CD	D8	1A	CALL INCR	;Salta a recibir siguiente dato.
79.	1945				LD A,<RENPOS>	;Toma direccion inicial.
80.	1945	3A	44	11	CP #37	;Apunta inicio de tabla.
81.	1948	FE	37		JP NZ,OTRA	;Almacena direccion final de tabla.
82.	194A	C2	07	19	LD A,#1E	;Obtiene longitud del buffer utilizado.
83.	194D	3E	1E		CALL ESPA	;Almacena longitud del buffer.
84.	194F	CD	3F	01	CALL PLANT	;Almacena direccion inicial de tabla.
85.	1952	CD	37	1A	LD D,#00	;Apunta area de memoria para dato.
86.	1955	16	00		LD D,#00	
87.	1957	CD	79	C4	CALL PDSC	
88.	195A	C3	07	19	JP OTRA	
89.	195D					
90.	195D	ED	5B	0B	LD DE,<DIR1>	;Toma direccion inicial.
91.	1961	FD	2A	0B	LD IY,<DIR1>	;Apunta inicio de tabla.
92.	1965	FD	75	FA	LD <Y-6>,L	;Almacena direccion final de tabla.
93.	1968	FD	74	FB	LD <Y-5>,H	;Obtiene longitud del buffer utilizado.
94.	196B	A7			AND A	;Almacena longitud del buffer.
95.	196C	ED	52		SBC HL,DE	;Almacena direccion inicial de tabla.
96.	196E	FD	75	FC	LD <Y-4>,L	;Apunta area de memoria para dato.
97.	1971	FD	74	FD	LD <Y-3>,H	
98.	1974	FD	73	FE	LD <Y-2>,E	
99.	1977	FD	72	FF	LD <Y-1>,D	
100.	197A	21	5A	11	LD HL,LET2	
101.	197D					

Sistema Graficador FAP-100

APEND A

102.	197D	36	20		LD <HL>,\$20	;Establece signo positivo.
103.	197F	23			INC HL	;Accesa parte baja de area
104.	1980	23			INC HL	;de memoria.
105.	1981	23			INC HL	;
106.	1982	FD	7E	FA	LD A,<IY-6>	;Toma parte baja direccion
107.	1985					;final.
108.	1985	CD	57	02	CALL HEXA	;Genera codigo ASCII de
109.	1988					;parte baja dir. final.
110.	1988	2B			DEC HL	;Accesa parte alta de area
111.	1989	2B			DEC HL	;de memoria.
112.	198A	2B			DEC HL	;
113.	198B	2B			DEC HL	;
114.	198C	FD	7E	FB	LD A,<IY-5>	;Toma parte alta direccion
115.	198F					;final.
116.	198F	CD	57	02	CALL HEXA	;Genera codigo ASCII de
117.	1992					;parte alta dir. final.
118.	1992	CD	E0	1A	CALL DESP	;Despliega valor direccion
119.	1995					;final.
120.	1995	21	5A	11	LD HL,LET2	;Direcciona area de memoria
121.	1998					;reservada para dato.
122.	1998	36	20		LD <HL>,\$20	;Establece signo positivo.
123.	199A	23			INC HL	;Accesa parte baja de area
124.	199B	23			INC HL	;de memoria.
125.	199C	23			INC HL	;
126.	199D	FD	7E	FC	LD A,<IY-4>	;Toma parte baja longitud
127.	19A0					;de buffer.
128.	19A0	CD	57	02	CALL HEXA	;Genera codigo ASCII de
129.	19A3					;parte baja long. buffer.
130.	19A3	2B			DEC HL	;Accesa parte alta de area
131.	19A4	2B			DEC HL	;de memoria.
132.	19A5	2B			DEC HL	;
133.	19A6	2B			DEC HL	;
134.	19A7	FD	7E	FD	LD A,<IY-3>	;Toma parte alta longitud
135.	19AA					;de buffer.
136.	19AA	CD	57	02	CALL HEXA	;Genera codigo ASCII de
137.	19AD					;parte alta long. buffer.
138.	19AD	21	9D	11	LD HL,LET6	;Despliega mensaje "LONGI-
139.	19B0	CD	BE	18	CALL ESMEN1	;TUD BUFFER: "
140.	19B3	21	5A	11	LD HL,LET2	;Despliega valor longitud
141.	19B6	CD	DC	01	CALL ESMEN	;buffer.
142.	19B9	CD	F3	1A	CALL NORM	;Genera tabla de maquina.
143.	19BC	CD	36	01	CALL LETEC	;Espera comando para re-
144.	19BF	FE	0D		CP #0D	;grezar a menu principal.
145.	19C1	20	F9		JR NZ,FINUNO	;
146.	19C3	C9			RET	;Termina.
147.	19C4	FD	21	5A	LD IY,LET2	;Direcciona area de me-
148.	19C8					;moría para dato.
149.	19C8	FD	36	00	LD <IY+0>,\$20	;Establece signo positivo.
150.	19CC	FD	36	01	LD <IY+1>,\$30	;Carga espacio con ceros.
151.	19D0	FD	36	02	LD <IY+2>,\$30	;
152.	19D4	FD	36	03	LD <IY+3>,\$30	;

FINUNO

LLENA

APEND A

Sistema Graficador FAP-100

153.	19D8	FD	36	04	30		LD <IY+4>,\$30	;
154.	19DC	C9					RET	;Termina.
155.	19DD	3E	20			INDIR	LD A,\$20	;Almacena numero renglon de
156.	19DF	32	44	11			LD <RENPOS>,A	;pantalla en que se posi-
157.	19E2							;ciona cursor.
158.	19E2	16	14			DIRDOS	LD D,\$14	;Establece numero columna.
159.	19E4	21	5A	11			LD HL,LET2	;Despliega valor de entrada
160.	19E7	CD	BE	18			CALL ESMEN1	;para dir. inicial.
161.	19EA	CD	79	C4			CALL POSC	;Posiciona cursor.
162.	19ED	CD	36	01		DIRUNO	CALL LETEC	;Recibe digito.
163.	19F0	FE	0D				CP #0D	;Checa si es fin de dir.
164.	19F2	C8					RET 2	;Salta si es fin de dir.
165.	19F3	CD	36	02			CALL FHFX	;Checa si es numero hex.
166.	19F6	FE	00				CP #00	;
167.	19F8	28	F3				JR Z,DIRUNO	;Salta si no es num. hex.
168.	19FA	21	5C	11			LD HL,\$115C	;Prepara espacio para
169.	19FD	21	5B	11			LD HL,\$115B	;nuevo digito.
170.	1A00	01	03	00			LD BC,\$0003	;
171.	1A03	ED	B0				LDIR	;
172.	1A05	2B					DEC HL	;
173.	1A06	77					LD <HL>,A	;Almacena nuevo digito.
174.	1A07	18	D9				JR DIRDOS	;Recibe siguiente digito.
175.	1A09	FD	21	5A	11	QUADIR	LD IY,LET2	;Apunta area de memoria
176.	1A0D							;de dato.
177.	1A0D	FD	7E	04			LD A,<IY+4>	;Toma digito menos sig-
178.	1A10							;nificativo.
179.	1A10	CD	2A	02			CALL ASHEX	;Genera formato hex.
180.	1A13	6F					LD L,A	;Almacena temporalmente.
181.	1A14	FD	7E	03			LD A,<IY+3>	;Toma siguiente digito.
182.	1A17	CD	2A	02			CALL ASHEX	;Genera formato hex.
183.	1A1A	CD	50	1B			CALL ROT4	;Prepara espacio para
184.	1A1D							;formar byte.
185.	1A1D	85					ADD A,L	;Genera primer byte de dir.
186.	1A1E	FD	7E	02			LD A,<IY+2>	;Toma siguiente digito.
187.	1A21	CD	2A	02			CALL ASHEX	;Genera formato hex.
188.	1A24	67					LD H,A	;Almacena temporalmente.
189.	1A25	FD	7E	01			LD A,<IY+1>	;Toma ultimo digito.
190.	1A28	CD	2A	02			CALL ASHEX	;Genera formato hex.
191.	1A2B	CD	50	1B			CALL ROT4	;Prepara espacio para
192.	1A2E							;formar segundo byte.
193.	1A2E	84					ADD A,H	;Genera 2o. byte de dir.
194.	1A2F	67					LD H,A	;
195.	1A30	22	0B	11			LD <DIR1>,HL	;Almacena dir. inicial.
196.	1A33	22	24	11			LD <CONDIR>,HL	;Inicializa apuntador.
197.	1A36	C9					RET	;Termina.
199.	1A37	21	60	11		PLANT	LD HL,LET3	;Despliega mensaje
199.	1A3A	CD	BE	18			CALL ESMEN1	;"RENGLON".
200.	1A3D	21	67	11			LD HL,LET4	;Despliega dos veces el
201.	1A40	CD	BE	18			CALL ESMEN1	mensaje"COORDX COORY
202.	1A43	21	67	11			LD HL,LET4	; CONTROL".
203.	1A46	CD	DC	01			CALL ESMEN	;

APEND A

Sistema Graficador FAP-100

204.	1A49	C9		RET	;Termina.
205.	1A4A	F5	BIASC	PUSH AF	;Preserva contenido de
206.	1A4B	E5		PUSH HL	;registros.
207.	1A4C	D5		PUSH DE	;
208.	1A4D	CD	C4 19	CALL LLENA	;Carga con ceros area de
209.	1A50				;de memoria para dato.
210.	1A50	CB 79		BIT 7,B	;Checa si es numero neg.
211.	1A52	28 0B		JR Z,BICERO	;Salta si es positivo.
212.	1A54	C5		PUSH BC	;Copia valor de numero.
213.	1A55	E1		POP HL	;
214.	1A56	CD 55 1B		CALL COMP2	;Obtiene valor positivo.
215.	1A59	E5		PUSH HL	;Copia valor de numero.
216.	1A5A	C1		POP BC	;
217.	1A5B	FD 36 00 2D		LD (1Y+0),#2D	;Establece signo negativo.
218.	1A5F	11 00 00	BICERO	LD DE,#0000	;Prepara reg. comparador.
219.	1A62	C5	BITRES	PUSH BC	;Copia valor numero.
220.	1A63	E1		POP HL	;
221.	1A64	A7		AND A	;Checa fin transformacion.
222.	1A65	28 49		JR Z,FINBIA	;Salta si es fin.
223.	1A67	FD 34 04		INC (1Y+4)	;Incrementa unidades.
224.	1A6A	FD 7E 04		LD A,(1Y+4)	;Checa si es numero maximo
225.	1A6D	FE 27		CP 39	;de unidades.
226.	1A6F	28 03		JR Z,BIUNO	;No es numero maximo.
227.	1A71	F2 77 1A		JP P,BIDOS	;Salta a decenas.
228.	1A74	0B	BIUNO	DEC BC	;Decrementa valor numero.
229.	1A75	13 EB		JR BITRES	;Prepara siguiente unidad.
230.	1A77	FD 36 04 30	BIDOS	LD (1Y+4),#30	;Prepara decena.
231.	1A7B	FD 34 03		INC (1Y+3)	;Incrementa valor decena.
232.	1A7E	FD 7E 03		LD A,(1Y+3)	;Checa si es numero maximo
233.	1A81	FE 39		CP #39	;de decenas.
234.	1A83	28 EF		JR Z,BIUNO	;Genera siguiente unidad.
235.	1A85	FA 74 1A		JP M,BIUNO	;
236.	1A88	FD 36 03 30		LD (1Y+3),#30	;Prepara centena.
237.	1A8C	FD 34 02		INC (1Y+2)	;Incrementa valor centena.
238.	1A8F	FD 7E 02		LD A,(1Y+2)	;Checa si es numero maximo
239.	1A92	FE 39		CP #39	;de centenas.
240.	1A94	28 DE		JR Z,BIUNO	;Genera siguiente unidad.
241.	1A96	FA 74 1A		JP M,BIUNO	;
242.	1A99	FD 36 02 30		LD (1Y+2),#30	;Prepara millar.
243.	1A9D	FD 34 01		INC (1Y+1)	;Incrementa valor millar.
244.	1AA0	FD 7E 01		LD A,(1Y+1)	;Checa si es numero maximo
245.	1AA3	FE 39		CP #39	;de millares.
246.	1AA5	28 CD		JR Z,BIUNO	;Genera siguiente unidad.
247.	1AA7	FA 74 1A		JP M,BIUNO	;
248.	1AAA	FD 36 01 30		LD (1Y+1),#30	;Establece cero total.
249.	1AAE	18 C4		JR BIUNO	;Reinicia.
250.	1AB0	01	FINBIA	POP DE	;Recupera contenido re-
251.	1AB1	E1		POP HL	;gistros.
252.	1AB2	F1		POP AF	;
253.	1AB3	C9		RET	;Termina.
254.	1AB4	32 26 11	DATO	LD (COL),A	;Almacena numero columna.

APEND A

Sistema Graficador FAP-100

255.	1AB7	57		LD D,A	;Copia numero columna.
256.	1AB8	CD	79	CALL POSC	;Posiciona cursor.
257.	1AB8	CD	60	CALL INDAT	;Recibe dato de teclado.
258.	1ABE	CD	A6	CALL GUADAS	;Almacena dato.
259.	1AC1	C9		RET	;Termina.
260.	1AC2	32	26	LD <COL>,A	;Almacena numero columna.
261.	1AC5	57		LD D,A	;Copia numero columna.
262.	1AC6	CD	79	CALL POSC	;Posiciona cursor.
263.	1AC9	CD	09	CALL INCONT	;Recibe palabra control.
264.	1ACC	CD	3D	CALL GUACON	;Almacena control.
265.	1ACF	C9		RET	;Termina.
266.	1AD0	2A	24	LD HL,<CONDIR>	;Toma ultimo control.
267.	1AD3	2B		DEC HL	;
268.	1AD4	CB	7E	BIT 7,<HL>	;Checa si es fin de tabla.
269.	1AD6	23		INC HL	;Actualiza contador.
270.	1AD7	C9		RET	;Termina.
271.	1AD8	2A	2B	LD HL,<RENG>	;Toma numero renglon.
272.	1AD8	23		INC HL	;Incrementa numero reng.
273.	1ADC	22	2B	LD <RENG>,HL	;Almacena no. actualizado.
274.	1ADF	C9		RET	;Termina.
275.	1AE0	CD	3A	CALL CRLF	;Accesa siguiente renglon.
276.	1AE3	CD	3A	CALL CRLF	;Accesa siguiente renglon.
277.	1AE6	21	8A	LD HL,LET5	;Despliega mensaje "DIREC-
278.	1AE9	CD	BE	CALL ESMEN1	;CION FINAL: "
279.	1AEC	21	5A	LD HL,LET2	;Despliega valor direccion
280.	1AEF	CD	DC	CALL ESMEN	;final.
281.	1AF2	C9		RET	;Termina.
282.	1AF3	21	E3	LD HL,LET11	;Despliega letrero
283.	1AF6	CD	DC	CALL ESMEN	; "NORM."
284.	1AF9	FD	2A	LD IY,<DIR1>	;Apunta inicio de tabla.
285.	1AFD	FD	6E	LD L,<IY-6>	;Toma valor de direccion
286.	1B00	FD	66	LD H,<IY-5>	;final.
287.	1B03	22	24	LD <CONDIR>,HL	;Actualiza apuntador.
288.	1B06	22	0B	LD <DIR1>,HL	;Apunta dir. inicial tabla
289.	1B09			;	;de maquina.
290.	1B09	21	05	LD HL,#0005	;Prepara factor de escala-
291.	1B0C	22	2F	LD <DIVOR>,HL	;miento.
292.	1B0F	FD	6E	LD L,<IY+0>	;Toma dato.
293.	1B12	FD	66	LD H,<IY+1>	;
294.	1B15	E5		PUSH HL	;Preserva signo de dato.
295.	1B16	CD	53	CALL NORMA	;Escala dato.
296.	1B19	D1		POP DE	;Retoma signo de dato.
297.	1B1A	CD	C1	CALL 1148	;Compensa redondeo en DIV.
298.	1B1D	CD	D7	CALL NUEDAT	;Almacena dato escalado.
299.	1B20	FD	6E	LD L,<IY+2>	;Toma dato.
300.	1B23	FD	66	LD H,<IY+3>	;
301.	1B26	E5		PUSH HL	;Preserva signo de dato.
302.	1B27	CD	53	CALL NORMA	;Escala dato.
303.	1B2A	D1		POP DE	;Retoma signo de dato.
304.	1B2B	CD	C1	CALL 1148	;Compensa redondeo en DIV.
305.	1B2E	CD	D7	CALL NUEDAT	;Almacena dato escalado.

APEND A

Sistema Graficador FAP-100

306.	1B31	CD	E4	1C	CALL NUECON	;Copia palabra de control.
307.	1B34	CB	7F		BIT 7,A	;Checa si fin de tabla.
308.	1B36	20	05		JR NZ,LISTO	;Salta si fin de tabla.
309.	1B38	CD	ED	1C	CALL SUMIYS	;Direcciona siguiente pa-
310.	1B3B					;reja de datos.
311.	1B3B	18	D2		JR DAT	;Prosigue escalamiento.
312.	1B3D	2A	27	11	LD HL,(ORGUSX)	;Genera origen relativo
313.	1B40	22	35	11	LD (ORGX),HL	;de tabla de maquina.
314.	1B43	2A	29	11	LD HL,(ORGUSY)	;
315.	1B46	22	37	11	LD (ORGY),HL	;
316.	1B49	21	CA	11	LD HL,LET9	;Despliega mensaje
317.	1B4C	CD	DC	01	CALL ESMEN	; "LISTO &".
318.	1B4F	C9			RET	;Termina.
319.	1B50	07			ROT4	;Rota a la izquierda
320.	1B51	07			RLCA	;cuatro veces el acumu-
321.	1B52	07			RLCA	;lador.
322.	1B53	07			RLCA	;
323.	1B54	C9			RET	;Termina.
324.	1B55	7D			LD A,L	;Toma parte baja de numero.
325.	1B56	2F			CPL	;Obtiene complemento a uno.
326.	1B57	6F			LD L,A	;Almacena temporalmente.
327.	1B58	7C			LD A,H	;Toma parte alta de numero.
328.	1B59	2F			CPL	;Obtiene complemento a uno.
329.	1B5A	67			LD H,A	;Almacena temporalmente.
330.	1B5B	11	01	00	LD DE,\$0001	;Suma uno y obtiene com-
331.	1B5E	19			ADD HL,DE	;plemento a dos.
332.	1B5F	C9			RET	;Termina.
333.	1B60	CD	C4	19	INDAT	CALL LLENA
334.	1B63					;Carga con ceros espacio
335.	1B63	21	5A	11	INDTRE	;reservado para dato.
336.	1B66	CD	BE	18	CALL ESMEN1	;Despliega valor de entr-
337.	1B69	3A	26	11	LD A,(COL)	;da para dato.
338.	1B6C	57			LD D,A	;Toma numero de columna.
339.	1B6D	CD	79	C4	INDUNO	;Copia numero de columna.
340.	1B70	CD	36	01	CALL LLETC	;Posiciona cursor.
341.	1B73	FE	20		CP #20	;Recibe digito.
342.	1B75	C8			RET Z	;Checa fin de dato.
343.	1B76	FE	2D		CP #2D	;Termina si fin de dato.
344.	1B78	20	05		JR NZ,OTROLA	;Checa si numero negativo.
345.	1B7A	FD	77	00	LD (Y+0),A	;Salta si numero positivo.
346.	1B7D	18	E4		JR INDTRE	;Establece signo negativo.
347.	1B7F	FE	2B		CP #2B	;Recibe siguiente digito.
348.	1B81	20	06		JR NZ,C30	;Checa si numero positivo.
349.	1B83	FD	36	00	LD (Y+0),\$20	;Salta si no es signo.
350.	1B87	18	DA		JR INDTRE	;Establece signo positivo.
351.	1B89	FE	30		CP #30	;Recibe siguiente digito.
352.	1B8B	FA	70	1B	JP M,INDUNO	;Realiza filtraje para
353.	1B8E	FE	39		CP #39	;recibir solamente numeros
354.	1B90	CA	96	1B	JP Z,INDDOS	;dentro del rango decimal.
355.	1B93	F2	70	1B	JP P,INDUNO	;
356.	1B96	21	5C	11	INDDOS	LD HL,\$115C
						;Prepara espacio para nue-



APEND A

Sistema Graficador FAP-100

357.	1B99	11	58	11	LD DE,\$115B	;	ivo digito.
358.	1B9C	01	03	00	LD BC,\$0003	;	
359.	1B9F	ED	B0		LDIR	;	
360.	1BA1	2B			DEC HL	;	
361.	1BA2	77			LD <HL>,A	;	Almacena nuevo digito.
362.	1BA3	C3	63	1B	JP INDTRE	;	Reciba siguiente digito.
363.	1BA6	01	5E	11	LD BC,\$115E	;	Apunta digito menos signi-
364.	1BA9					;	ficativo de area dato.
365.	1BA9	21	00	00	LD HL,\$0000	;	Inicializa reg. resultado.
366.	1BAC	CD	F1	1B	CALL PREPA	;	Convierte digito de ASCII
367.	1BAF					;	a binario.
368.	1BAF	C5			PUSH BC	;	Preserva direccion.
369.	1BB0	01	01	00	LD BC,\$0001	;	Prepara multiplicador.
370.	1BB3	CD	F8	1B	CALL CONV	;	Genera unidades de dato.
371.	1BB6	C1			POP BC	;	Retoma direccion.
372.	1BB7	0B			DEC BC	;	Accesa decenas de dato.
373.	1BB8	CD	F1	1B	CALL PREPA	;	Convierte de ASCII a bin.
374.	1BB8	C5			PUSH BC	;	Preserva direccion.
375.	1BB0	01	0A	00	LD BC,\$000A	;	Prepara multiplicador.
376.	1BBF	CD	F8	1B	CALL CONV	;	Genera unidades mas decen-
377.	1BC2					;	mas de dato en binario.
378.	1BC2	C1			POP BC	;	Retoma direccion.
379.	1BC3	0B			DEC BC	;	Accesa centenas de dato.
380.	1BC4	CD	F1	1B	CALL PREPA	;	Convierte de ASCII a bin.
381.	1BC7	C5			PUSH BC	;	Preserva direccion.
382.	1BC8	01	64	00	LD BC,\$0064	;	Prepara multiplicador.
383.	1BC8	CD	F8	1B	CALL CONV	;	Genera unidades, decenas
384.	1BCE					;	y centenas de dato en bin.
385.	1BCE	C1			POP BC	;	Retoma direccion.
386.	1BCF	0B			DEC BC	;	Accesa millares de dato.
387.	1BD0	CD	F1	1B	CALL PREPA	;	Convierte de ASCII a bin.
388.	1BD3	C5			PUSH BC	;	Preserva direccion.
389.	1BD4	01	E8	03	LD BC,\$03E8	;	Prepara multiplicador.
390.	1BD7	CD	F8	1B	CALL CONV	;	Genera unidades, decenas,
391.	1BDA					;	centenas y millares de da-
392.	1BDA					;	to en binario.
393.	1BDA	C1			POP BC	;	Retoma direccion.
394.	1BDB	0B			DEC BC	;	Accesa signo de dato.
395.	1BDC	0A			LD A,<BC>	;	Toma signo de dato.
396.	1BDD	FE	20		CP #20	;	Checa si es positivo.
397.	1BDF	28	03		JR Z.POSI	;	Salta si es positivo.
398.	1BE1	CD	55	1B	CALL COMP2	;	Obtiene complemento a dos
399.	1BE4					;	si es negativo.
400.	1BE4	E5			PUSH HL	;	Copia valor dato en bi-
401.	1BE5	D1			POP DE	;	nario.
402.	1BE6	2A	24	11	LD HL,<CONDIR>	;	Toma ultima direccion uti-
403.	1BE9					;	lizada.
404.	1BE9	73			LD <HL>,E	;	Almacena valor de dato.
405.	1BEA	23			INC HL	;	
406.	1BEB	72			LD <HL>,D	;	
407.	1BEC	23			INC HL	;	

GUADAS

POSI

APEND A

Sistema Graficador FAP-100

408.	1BED	22	24	11		LD <CONDIR>,HL	;Actualiza apuntador.
409.	1BF0	C9				RET	;Termina.
410.	1BF1	0A			PREPA	LD A,<BC>	;Toma digito.
411.	1BF2	E6	0F			AND \$0F	;Convierte de ASCII a bin.
412.	1BF4	5F				LD E,A	;Almacena valor binario.
413.	1BF5	16	00			LD D,\$00	;
414.	1BF7	C9				RET	;Termina.
415.	1BF8	19			CONV	ADD HL,DE	;Realiza suma.
416.	1BF9	0B				DEC BC	;Decrementa multiplicador.
417.	1BFA	E5				PUSH HL	;Preserva contenido de pro-
418.	1BFB	05				PUSH DE	ducto, multiplicando y
419.	1BFC	C5				PUSH BC	;multiplicador.
420.	1BFD	E1				POP HL	;Toma multiplicador.
421.	1BFE	11	00	00		LD DE,\$0000	;Checa si este es cero.
422.	1C01	A7				AND A	;
423.	1C02	E0	52			SBC HL,DE	;
424.	1C04	D1				POP DE	;Retoma multiplicando.
425.	1C05	E1				POP HL	;Retoma producto.
426.	1C06	20	F0			JR NZ,CONV	;Prosigue si multiplicador
427.	1C08						;no es cero.
428.	1C08	C9				RET	;Si es cero, termina.
429.	1C09	FD	21	5A	11	INCONT	LD IY,LET2
430.	1C0D						;Apunta area de memoria
431.	1C0D	FD	36	03	30		;para dato.
432.	1C11					LD <IY+3>,\$30	;Establece valor control
433.	1C11	FD	7E	03			de entrada.
434.	1C14	CD	3F	01		LD A,<IY+3>	;Despliega valor de en-
435.	1C17	3A	26	11	INCTRE	CALL ESPA	trada para control.
436.	1C1A	57				LD A,<COL>	;Toma numero columna.
437.	1C1B	CD	79	C4		LD D,A	;Copia numero columna.
438.	1C1E	CD	36	01	INCUNO	CALL POSC	;Posiciona cursor.
439.	1C21	FE	20			CALL LETEC	;Recibe control.
440.	1C23	C8				CP \$20	;Checa si es fin control.
441.	1C24	FE	18			RET Z	;Termina si fin control.
442.	1C26	28	0F			CP \$18	;Checa si fin tabla.
443.	1C28	FE	30			JR Z,MANDX	;Salta si fin tabla.
444.	1C2A	FA	1E	1C		CP \$30	;Establece filtro para re-
445.	1C2D	FE	33			JP M,INCUNO	cibir solo control valido.
446.	1C2F	F2	1E	1C		CP \$33	;
447.	1C32	FD	77	03	INCDOS	JP P,INCUNO	;
448.	1C35	18	DD			LD <IY+3>,A	;Almacena nuevo control.
449.	1C37	3E	58			JR INCTRE*	;Prosigue captura.
450.	1C39	CD	3F	01	MANDX	LD A,\$58	;Despliega etiqueta fin
451.	1C3C	C9				CALL ESPA	de tabla.
452.	1C3D	FE	20		GUACON	RET	;Termina.
453.	1C3F					CP \$20	;Checa si es control va-
454.	1C3F	2A	24	11			lido.
455.	1C42	28	04			LD HL,<CONDIR>	;Accesa ultima localidad.
456.	1C44	3E	FF			JR Z,GUAUHO	;Salta si control valido.
457.	1C46	18	05			LD A,\$FF	;Carga codigo fin tabla.
458.	1C48	FD	7E	03	GUAUHO	JR GUADOS	;Salta a almacenar.
						LD A,<IY+3>	;Toma control valido.



APEND A

Sistema Graficador FAP-100

510.	ICAS	20 02		JR NZ,CUENTA	;Salta divisor>dividendo.
511.	ICAT	CB C3		SET 0,E	;Pone a 1 bit de cociente.
512.	ICAN	3A 26 11	CUENTA	LD A,(COL)	;Checa si el numero de
513.	ICAC	3D		DEC A	;corrimientos requeridos
514.	ICAD	32 26 11		LD (COL),A	;se ha cumplido.
515.	ICB0	FE 00		CP \$00	;
516.	ICB2	20 C5		JR NZ,SHIFT	;Salta si no termina.
517.	ICB4	7A		LD A,D	;Toma 4 bits validos de
518.	ICB5	E6 0F		AND \$0F	;registro D.
519.	ICB7	67		LD H,A	;Los almacena en H.
520.	ICB8	6B		LD L,E	;Copia 8 bits validos de
521.	ICB9				;registro E.
522.	ICB9	D1		POP DE	;Retoma signo inicial.
523.	ICBA	CB 7A		BIT 7,D	;Checa signo inicial.
524.	ICBC	C8		RET Z	;Termina si es positivo.
525.	ICBD	CD 55 1B		CALL COMP2	;Obtiene complemento a 2
526.	ICCC				;si es negativo.
527.	ICCC	C9		RET	;Termina.
528.	ICCC	E5	1148	PUSH HL	;Guarda resultado.
529.	ICCC	21 00 00		LD HL,\$0000	;Prepara reg. comparador.
530.	ICCC	01 30 00		LD BC,\$0030	;Prepara el factor para la
531.	ICCC				;compensacion.
532.	ICCC	A7		AND A	;Evalua si es cero.
533.	ICCC	ED 52		SBC HL,DE	;
534.	ICCC	E1		POP HL	;Retoma resultado.
535.	ICCC	C8		RET Z	;Termina si es cero.
536.	ICCC	CB 7A		BIT 7,D	;Evalua signo de dato.
537.	ICCF	20 02		JR NZ,MENOS	;Salta si negativo.
538.	ICD1	09		ADD HL,BC	;Suma factor si positivo.
539.	ICD2	C9		RET	;Termina.
540.	ICD3	A7	MENOS	AND A	;Resta factor si negativo.
541.	ICD4	ED 42		SBC HL,BC	;
542.	ICD6	C9		RET	;Termina.
543.	ICD7	E5	HUEDAT	PUSH HL	;Preserva valor dato.
544.	ICD8	D1		POP DE	;Toma valor dato.
545.	ICD9	2A 24 11		LD HL,(CONDIR)	;Toma direccion ultima
546.	ICDC				;localidad utilizada.
547.	ICDC	73		LD (HL),E	;Almacena parte baja dato.
548.	ICDD	23		INC HL	;Actualiza apuntador.
549.	ICDE	72		LD (HL),D	;Almacena parte alta dato.
550.	ICDF	23		INC HL	;Actualiza apuntador.
551.	ICE0	22 24 11		LD (CONDIR),HL	;Guarda apuntador.
552.	ICE3	C9		RET	;Termina.
553.	ICE4	FD 7E 04	HUECON	LD A,(IY+4)	;Toma palabra control.
554.	ICE7	77		LD (HL),A	;Almacena en ultima loca-
555.	ICE8				lidad disponible.
556.	ICE8	23		INC HL	;Actualiza apuntador.
557.	ICE9	22 24 11		LD (CONDIR),HL	;Guarda apuntador.
558.	ICEC	C9		RET	;Termina.
559.	ICED	11 05 00	SUMIY5	LD DE,\$0005	;Prepara factor a sumar.
560.	ICF0	FD 19		ADD IY,DE	;Suma factor a IY.

APEND A

Sistema Graficador FAP-100

```

561. 1CF2 C9          RET          ;Termina.
562. 1CF3
563. 0236
564. 0236
565. 0236 F5          FHEX          PUSH AF          ;Preserva dato y estatus.
566. 0237 FE 30      CP #30          ;Checa limite inferior
567. 0239
568. 0239 F2 3E 02   JP P,FHEX0      ;Salta si esta dentro de
569. 023C
570. 023C 18 15      JR FINFH        ;Termina si esta fuera.
571. 023E FE 3A      FHEX0          CP #3A          ;Checa limite superior
572. 0240
573. 0240 FA 51 02   JP M,FHEX1      ;Salta si esta dentro de
574. 0243
575. 0243 FE 41      CP #41          ;Checa limite inferior
576. 0245
577. 0245 F2 4A 02   JP P,FHEX2      ;Salta si esta dentro de
578. 0248
579. 0248 18 09      JR FINFH        ;Termina si esta fuera.
580. 024A FE 47      FHEX2          CP #47          ;Checa limite superior
581. 024C
582. 024C FA 51 02   JP M,FHEX1      ;Salta si esta dentro de
583. 024F
584. 024F 18 02      JR FINFH        ;Termina si esta fuera.
585. 0251 F1          FHEX1          POP AF          ;Recupera dato y estatus.
586. 0252 C9          RET            ;Termina.
587. 0253 F1          FINFH          POP AF          ;Recupera dato y estatus.
588. 0254 3E 00      LD A,#00        ;Carga codigo error.
589. 0256 C9          RET            ;Termina.
590. 0257 E5          HEXA          PUSH HL        ;Preserva registros.
591. 0258 C5          PUSH BC
592. 0259 F5          PUSH AF
593. 025A 23          INC HL
594. 025B
595. 025B 0E 00      LD C,#00        ;Inicializa contador.
596. 025D E6 0F      AND #0F         ;Define nibble bajo.
597. 025F 06 0A      LD B,#0A        ;Prepara reg. comparador.
598. 0261 B8          HEX0          CP B           ;Checa si es numero.
599. 0262 FA 69 02   JP M,SUMA0      ;Salta si es numero.
600. 0265 C6 37      ADD A,#37       ;Suma si es letra.
601. 0267 18 02      JR COMUN
602. 0269 C6 30      SUMA0          ADD A,#30       ;Suma si es numero.
603. 026B 77          COMUN         LD <HL>,A      ;Guarda nibble bajo.
604. 026C 2B          DEC HL          ;Direcciona espacio nibble
605. 026D
606. 026D 0C          INC C          ;Incrementa contador.
607. 026E 3E 02      LD A,#02       ;Prepara reg. comparador.
608. 0270 B9          CP C           ;Checa si fin conversion.
609. 0271 28 0C      JR Z,HEX1      ;Salta si fin conversion.
610. 0273 F1          POP AF         ;Recupera dato.
611. 0274 F5          PUSH AF        ;Preserva dato.

```

APEND A

Sistema Graficador FAP-100

```

612. 0275 CB 3F          SRL A          ;Toma nibble alto.
613. 0277 CB 3F          SRL A          ;
614. 0279 CB 3F          SRL A          ;
615. 027B CB 3F          SRL A          ;
616. 027D 18 E2          JR HEX0        ;Prosigue conversion.
617. 027F F1             POP AF         ;Recupera registros.
618. 0280 C1             POP BC         ;
619. 0281 E1             POP HL         ;
620. 0282 23             INC HL         ;Actualiza apuntador.
621. 0283 23             INC HL         ;
622. 0284 C9             RET            ;Termina.
623. 0285                ;
624. 022A                ;*# $022A
625. 022A                ;
626. 022A CB 77          ASHEX         BIT 6,A        ;Checa si es letra.
627. 022C 20 03          JR NZ,ASHE0   ;Salta si es letra.
628. 022E E6 0F          AND $0F       ;Obtiene numero decimal.
629. 0230 C9             RET            ;Termina.
630. 0231 E6 0F          ASHE0         AND $0F        ;Obtiene letra hexadeci-
631. 0233 C6 09          ADD A,$09     ;mal.
632. 0235 C9             RET            ;Termina.
633. 0236                ;
634. C401                ;*# $C401
635. C401                ;
636. C401 3A 44 11       INCR          LD A,(RENPOS) ;Toma numero renglon.
637. C404 FE 37          CP $37        ;Checa si numero maximo.
638. C405 C8             RET Z         ;Termina si no maximo.
639. C407 3C             INC A         ;Incrementa numero.
640. C408 32 44 11       LD (RENPOS),A ;Almacena nuevo numero.
641. C40B C9             RET            ;Termina.
642. C40C                ;
643. C479                ;*# $C479
644. C479                ;
645. C479 3E 1B          POSC          LD A,$1B      ;Manda comando para que TV
646. C47B CD 3F 01       CALL ESPA     ;posicione cursor.
647. C47E 3E 3D          LD A,$3D     ;
648. C480 CD 3F 01       CALL ESPA     ;
649. C483 3A 44 11       LD A,(RENPOS) ;Manda numero de renglon.
650. C486 CD 3F 01       CALL ESPA     ;
651. C489 7A             LD A,D       ;Manda numero de columna.
652. C48A CD 3F 01       CALL ESPA     ;
653. C48D C9             RET            ;Termina.
654. C48E                ;
655. C53A                ;*# $C53A
656. C53A                ;
657. C53A 3E 0D          CRLF         LD A,$0D      ;Manda codigo de retorno
658. C53C CD 3F 01       CALL ESPA     ;de carro a TV.
659. C53F 3E 0A          LD A,$0A     ;Manda codigo de alimen-
660. C541 CD 3F 01       CALL ESPA     ;tacion de linea a TV.
661. C544 C9             RET            ;Termina.
661. C545

```

T a b l a d e S i m b o l o s

Nombre	Valor	Linea de Definicion
let1	1145	11
let2	115A	12

let3	1160	13
let4	1167	14
let5	118A	15
let6	1190	16
let9	11CA	17
let11	11E3	18
letec	0136	19
espa	013F	20
esmen	01DC	21
esmen1	18BE	22
orgusx	1127	23
orgusy	1129	24
orgx	1135	25
orgy	1137	26
renpos	1144	27
reng	112B	28
divor	112F	29
col	1126	30
condir	1124	31
dir1	1108	32
genta	18D7	35
otra	1907	54
fin	195D	90
fingen	197A	100
finuno	198C	143
llena	19C4	147
indir	19D0	155
dirdos	19E2	158
diruno	19E0	162
guadir	1A09	175
plant	1A37	198
biasc	1A4A	205
bicero	1A5F	218
bitres	1A62	219
biuno	1A74	228
bidos	1A77	230
finbia	1A80	250
dato	1A84	254
contro	1AC2	260
checa	1AD0	266
incran	1AD8	271
desp	1AE0	275
norm	1AF3	282
dat	1B0F	292
listo	1B3D	312
rot4	1B50	319
comp2	1B55	324
indat	1B60	333
indtre	1B63	335
induno	1B70	340
otrola	1B7F	347
c30	1B89	351
inddos	1B96	356
guadas	1BA6	363
posi	1BE4	400
prepa	1BF1	410
conv	1BF8	415
incont	1C09	429
inctre	1C14	434
incuno	1C1E	438

incdos	1C32	447
mandx	1C37	449
guacon	1C3D	452
guauno	1C48	458
guados	1C4D	460
norma	1C53	464
div	1C5B	468
divuno	1C63	473
div16	1C70	481
shift	1C79	484
divdos	1C83	489
divtre	1C8B	493
divcua	1C93	497
resta	1C9C	503
limh	1C9F	505
cuenta	1CA9	512
1148	1CC1	528
menos	1CD3	540
nuedat	1CD7	543
nuecon	1CE4	553
sumiy5	1CED	559
fhex	0236	565
fhex0	023E	571
fhex2	024A	580
fhex1	0251	585
finfh	0253	587
hexa	0257	590
hex0	0261	598
suma0	0269	602
comun	026B	603
hex1	027F	617
ashex	022A	626
ashe0	0231	630
incr	C401	636
posc	C479	645
crlf	C53A	657



APEND A

Sistema Graficador FAP-100

```

1. 0000 ;
2. 0000 ;*** COMUNICACION CON CINTA ***
3. 0000 ;
4. 0000 ;
5. 1CF3 * = $1CF3
6. 1CF3 LET7 = $11B0
7. 1CF3 ( LET8 = $11B7
8. 1CF3 LET9 = $11CA
9. 1CF3 LETEC = $0136
10. 1CF3 ESPA = $013F
11. 1CF3 ESMEN = $01DC
12. 1CF3 ESMEN1 = $18BE
13. 1CF3 UARTST = $03
14. 1CF3 UARTIO = $02
15. 1CF3 ORGUSX = $1127
16. 1CF3 ORGUSY = $1129
17. 1CF3 BUFL = $10FA
18. 1CF3 BUFB = $10FC
19. 1CF3 TOPE = $1100
20. 1CF3 DISP = $1102
21. 1CF3 MODO = $112D
22. 1CF3 LONG = $1124
23. 1CF3 CONDIR = $110B
24. 1CF3 DIR1 = $0892
25. 1CF3 FINAL = $0971
26. 1CF3 ENTACK =
27. 1CF3 ;
28. 1CF3 ;
29. 1CF3 3E 1A ACCETA LD A,$1A ;Manda comando para borrar
30. 1CF5 CD 3F 01 CALL ESPA ;pantalla.
31. 1CF8 21 B7 11 LD HL,LET8 ;Despliega mensaje "CR PARA
32. 1CFB CD BE 18 CALL ESMEN1 ;TRANSMITIR".
33. 1CFE CD 36 01 ACUNO CALL LETEC ;Espera comando inicio.
34. 1D01 FE 0D CP $0D ;
35. 1D03 20 F9 JR NZ,ACUNO ;Salta si no es comando.
36. 1D05 CD 3F 01 CALL ESPA ;Manda CR a pantalla.
37. 1D09 3E 0A LD A,$0A ;Manda line feed a panta-
38. 1D0A CD 3F 01 CALL ESPA ;lla.
39. 1D0D 3E F8 LD A,$F8 ;Almacena codigo seleccion
40. 1D0F 32 00 11 LD <DISP>,A ;dispositivo.
41. 1D12 3E 00 LD A,$00 ;Establece comunicacion sin
42. 1D14 32 FA 10 LD <BUFL>,A ;respuesta.
43. 1D17 21 B0 11 LD HL,LET7 ;Despliega mensaje
44. 1D1A CD DC 01 CALL ESMEN ;"TRANS.".
45. 1D1D CD 85 08 CALL INICIO ;Habilita comunicacion X-
46. 1D20 ;MODEM.
47. 1D20 CD 36 01 CALL LETEC ;Recibe y almacena coordena-
48. 1D23 32 27 11 LD <$1127>,A ;nada X de origen usuario.
49. 1D26 CD 36 01 CALL LETEC ;
50. 1D29 32 28 11 LD <$1128>,A ;

```

Sistema Graficador FAP-100

APEND	A				
51.	102C	CD	36	01	CALL LETEC ;Recibe y almacena coorde-
52.	102F	32	29	11	LD (\$1129),A ;nada Y de origen usuario.
53.	1032	CD	36	01	CALL LETEC ;
54.	1035	32	2A	11	LD (\$112A),A ;
55.	1038	CD	36	01	CALL LETEC ;Recibe y almacena direc-
56.	103B	32	FC	10	LD (\$10FC),A ;cion final de tabla.
57.	103E	CD	36	01	CALL LETEC ;
58.	1041	32	FD	10	LD (\$10FD),A ;
59.	1044	CD	36	01	CALL LETEC ;Recibe y almacena longi-
60.	1047	32	2D	11	LD (\$112D),A ;tud de buffer.
61.	104A	CD	36	01	CALL LETEC ;
62.	104D	32	2E	11	LD (\$112E),A ;
63.	1050	CD	36	01	CALL LETEC ;Recibe y almacena direc-
64.	1053	6F			LD L,A ;cion inicial tabla.
65.	1054	CD	36	01	CALL LETEC ;
66.	1057	67			LD H,A ;
67.	1058	22	0B	11	LD (<DIR1>),HL ;
68.	105B	CD	80	09	CALL XHENT ;Recibe archivo de datos.
69.	105E	CD	10	1E	CALL FINI ;Habilita TV y da estatus.
70.	1061	FD	2A	0B 11	LD IY,<DIR1> ;Apunta inicio de tabla.
71.	1065	2A	0B	11	LD HL,<DIR1> ;Toma direccion inicial.
72.	1068	FD	74	FF	LD <IY-1>,H ;Almacena como parametro
73.	106B	FD	75	FE	LD <IY-2>,L ;la direccion inicial.
74.	106E	2A	2D	11	LD HL,<LONG> ;Toma longitud de buffer.
75.	1071	FD	74	FD	LD <IY-3>,H ;Almacena como parametro
76.	1074	FD	75	FC	LD <IY-4>,L ;longitud de buffer.
77.	1077	2A	FC	10	LD HL,<TOPE> ;Toma direccion final.
78.	107A	FD	74	FB	LD <IY-5>,H ;Almacena como parametro
79.	107D	FD	75	FA	LD <IY-6>,L ;direccion final tabla.
80.	1080	2A	0B	11	LD HL,<DIR1> ;Actualiza apuntador de
81.	1083	22	24	11	LD <CONDIR>,HL ;tabla.
82.	1086	CD	36	01	CALL LETEC ;Espera comando para fin.
83.	1089	FE	0D		CP \$0D ;Checa si es comando fin.
84.	108B	20	F9		JR NZ,ACDOS ;Salta si no es comando.
85.	108D	C9			RET ;Termina.
86.	108E	3E	1A		LD A,\$1A ;Manda comando para borrar
87.	1090	CD	3F	01	CALL ESPA ;pantalla.
88.	1093	21	B7	11	LD HL,LET8 ;Despliega mensaje "CR PA-
89.	1096	CD	BE	18	CALL ESMEN1 ;RA TRANSMITIR".
90.	1099	CD	36	01	CALL LETEC ;Espera comando inicio.
91.	109C	FE	0D		CP \$0D ;Checa si comando inicio.
92.	109E	20	F9		JR NZ,CINUHO ;Salta si no es comando.
93.	10A0	CD	3F	01	CALL ESPA ;Manda CR a pantalla.
94.	10A3	3E	0A		LD A,\$0A ;Manda line feed a panta-
95.	10A5	CD	3F	01	CALL ESPA ;lla.
96.	10A8	3E	F8		LD A,\$F8 ;Genera codigo seleccion
97.	10AA	32	00	11	LD <DISP>,A ;de dispositivo.
98.	10AD	3E	00		LD A,\$00 ;Genera codigo de procer-
99.	10AF	32	FA	10	LD <BUFL>,A ;dencia informacion.
100.	10B2	3E	80		LD A,\$80 ;Genera parametro que de-
101.	10B4	32	02	11	LD <MODE>,A ;fine modo de transmision.

APEND A

Sistema Graficador FAP-100

102.	10B7	21	80	11		LD HL,LET7	;Despliega mensaje
103.	10BA	CD	DC	01		CALL ESMEN	;*TRANS.*
104.	10BD	CD	85	08		CALL INICIO	;Activa comunicacion X-
105.	10C0						;MODEM.
106.	10C0	2A	24	11		LD HL,(CONDIR)	;Calcula longitud del ar-
107.	10C3	ED	5B	0B	11	LD DE,(DIRT)	;chivo a almacenarse.
108.	10C7	A7				AND A	;
109.	10C8	ED	52			SBC HL,DE	;
110.	10CA	EB				EX DE,HL	;Obtiene direccion inicial.
111.	10CB	A7				AND A	;
112.	10CC	ED	52			SBC HL,DE	;
113.	10CE	01	0A	00		LD BC,#000A	;Direcciona espacio reser-
114.	10D1	A7				AND A	;vado para parametros de
115.	10D2	ED	42			SBC HL,BC	;tabla.
116.	10D4	E5				PUSH HL	;Indexa parametros.
117.	10D5	DD	E1			POP IX	;
118.	10D7	ED	5B	27	11	LD DE,(ORGUSX)	;Toaa y almacena coordenada
119.	10DB	DD	73	00		LD (IX+0),E	;X origen usuario.
120.	10DE	DD	72	01		LD (IX+1),D	;
121.	10E1	ED	5B	29	11	LD DE,(ORGUSY)	;Toaa y almacena coordenada
122.	10E5	DD	73	02		LD (IX+2),E	;Y origen usuario.
123.	10E8	DD	72	03		LD (IX+3),D	;
124.	10EB	7E				LD A,(HL)	;Transmite la cinta paramet-
125.	10EC	CD	3F	01		CALL ESPA	;tros de tabla.
126.	10EF	0D				DEC C	;
127.	10F0	23				INC HL	;
128.	10F1	79				LD A,C	;
129.	10F2	FE	00			CP #00	;Checa fin de parametros.
130.	10F4	20	F5			JR NZ,CINDOS	;Salta si no es fin.
131.	10F6	E5				PUSH HL	;Actualiza parametros para
132.	10F7	FD	E1			POP IY	;X-MODEM.
133.	10F9	FD	4E	FC		LD C,(IY-4)	;
134.	10FC	FD	46	FD		LD B,(IY-3)	;
135.	10FF	C5				PUSH BC	;
136.	1E00	DD	E1			POP IX	;
137.	1E02	CD	AA	08		CALL XNSAL	;Transmite archivo a cinta.
138.	1E05	CD	10	1E		CALL FIN1	;Activa TV y da estatus.
139.	1E08	CD	36	01		CALL LETEC	;Espere comando fin.
140.	1E0B	FE	0D			CP #0D	;Checa si es comando fin.
141.	1E0D	20	F9			JR NZ,CINTRE	;Salta si no es comando.
142.	1E0F	C9				RET	;Termina.
143.	1E10	F5				PUSH AF	;Preserva registros y es-
144.	1E11	C5				PUSH BC	;tatus.
145.	1E12	D5				PUSH DE	;
146.	1E13	3E	38			LD A,#38	;Genera retardo.
147.	1E15	CD	A4	09		CALL RETD0	;
148.	1E18	3E	CF			LD A,#CF	;Genera codigo TV.
149.	1E1A	D3	00			OUT (#00),A	;Habilita terminal video.
150.	1E1C	3E	0D			LD A,#0D	;Genera retardo.
151.	1E1E	CD	A4	09		CALL RETD0	;
152.	1E21	D1				POP DE	;Recupera registros.

APEND A

Sistema Graficador FAP-100

```

153. 1E22 C1          POP BC          ;
154. 1E23 F1          POP AF          ;
155. 1E24 CD 28 1E    CALL XMENSA     ;Despliega mensaje de es-
156. 1E27             ;                    ;tstatus.
157. 1E27 C9          RET                    ;Termina.
158. 1E28 FE 00       XMENSA CP $00          ;Checa si existe error.
159. 1E2A 20 07       JR NZ,XUNO           ;Salta si existe error.
160. 1E2C 21 CA 11    XTRES LD HL,LET9     ;Despliega mensaje
161. 1E2F CD DC 01    CALL ESMEN     ;"LISTO &".
162. 1E32 C9          RET                    ;Termina.
163. 1E33 FE 05       XUNO CP $05           ;Checa tipo de error.
164. 1E35 28 F5       JR Z,XTRES           ;Salta si error no signi-
165. 1E37             ;                    ;ficativo.
166. 1E37 CD DC 01    CALL ESMEN     ;Despliega tipo de error.
167. 1E3A C9          RET                    ;Termina.
168. 1E3B             ;
169. 0885             ;                    ;== $0885
170. 0885             ;
171. 0885 3E 3E       INICIO LD A,$3E      ;Genera parametro retardo.
172. 0887 CD 63 0A    CALL NVO        ;Da retardo y toma codigo
173. 088A             ;                    ;seleccion dispositivo.
174. 088A D3 00       OUT (<$00>,A)        ;Activa comunicacion XM.
175. 088C 3E 0D       LD A,$0D        ;Genera retardo.
176. 088E CD A4 09    CALL RETDO     ;
177. 0891 C9          RET                    ;Termina.
178. 0892             ;
179. 08AA             ;                    ;== $08AA
180. 08AA             ;
181. 08AA 0E 00       XMSAL LD C,$00      ;Inicia numero de bloque.
182. 08AC 3E 01       XMS1 LD A,$01      ;Envia inicio de encabe-
183. 08AE CD 3F 01    CALL ESPA     ;zado.
184. 08B1 79          LD A,C            ;Manda numero de bloque.
185. 08B2 CD 3F 01    CALL ESPA     ;
186. 08B5 2F          CPL                    ;Envia numero de bloque
187. 08B6 CD 3F 01    CALL ESPA     ;Invertido.
188. 08B9 46          LD B,<HL>           ;Inicializa CKSUM.
189. 08BA 16 80       LD D,$80        ;Inicializa numero de
190. 08BC             ;                    ;datos por bloque.
191. 08BC 7E          XMS2 LD A,<HL>      ;Toma y envia dato.
192. 08BD CD 3F 01    CALL ESPA     ;
193. 08C0 DD 2B       DEC IX            ;Decrementa contador.
194. 08C2 E5          PUSH HL           ;Preserva direccion
195. 08C3             ;                    ;apuntador.
196. 08C3 21 00 90    LD HL,$0000    ;Checa si contador de
197. 08C6 05          PUSH DE            ;caracteres es igual a
198. 08C7 DD E5       PUSH IX            ;cero.
199. 08C9 D1          POP DE            ;
200. 08CA A7          AND A            ;
201. 08CB ED 52       SBC HL,DE        ;
202. 08CD D1          POP DE            ;
203. 08CE E1          POP HL            ;Recupera apuntador.

```

APEND A

Sistema Graficador FAP-100

204.	08CF	CA	E8	08		JP Z,XMSU	;Salta si termina datos.
205.	08D2	23				INC HL	;Actualiza apuntador.
206.	08D3	15				DEC D	;Decrementa no. datos.
207.	08D4	CA	DD	08		JP Z,XMS3	;Salta si completo bloque.
208.	08D7	78				LD A,B	;Actualiza CKSUM en B.
209.	08D8	86				ADD A,(HL)	;
210.	08D9	47				LD B,A	;
211.	08DA	C3	BC	08		JP XMS2	;Prosigue comunicacion.
212.	08DD	78			XMS3	LD A,B	;Toma CKSUM.
213.	08DE	CD	3F	01		CALL ESPA	;Envia CKSUM.
214.	08E1	CD	3A	09		CALL RESP	;Recibe respuesta.
215.	08E4	0C			BETA	INC C	;Incrementa no. de bloque.
216.	08E5	C3	AC	08		JP XMS1	;Cierra bucle de salida.
217.	08E8	FD	21	00	00 XMSU	LD IY,\$0000	;Completa con ceros el
218.	08EC	15			XMSU1	DEC D	;ultimo bloque.
219.	08ED	CA	FA	08		JP Z,XMSU2	;
220.	08F0	3E	0A			LD A,\$00	;
221.	08F2	CD	3F	01		CALL ESPA	;
222.	08F5	FD	23			INC IY	;
223.	08F7	C3	EC	08		JP XMSU1	;
224.	08FA	78			XMSU2	LD A,B	;Envia CKSUM.
225.	08FB	CD	3F	01		CALL ESPA	;
226.	08FE	CD	3A	09		CALL RESP	;Recibe respuesta.
227.	0901	FE	15			CP \$15	;Checa si el dispositivo de
228.	0903						;comunicacion da respuesta.
229.	0903	C2	18	09		JP HZ,XMSFIN	;Salta si no hay respuesta.
230.	0906	FD	E5			PUSH IY	;HL = HL + IY
231.	0908	D1				POP DE	;
232.	0909	19				ADD HL,DE	;
233.	090A	E5				PUSH HL	;Guarda HL
234.	090B	DD	E5			PUSH IX	;IX = IX - IY.
235.	090D	E1				POP HL	;
236.	090E	A7				AND A	;
237.	090F	ED	52			SBC HL,DE	;
238.	0911	E5				PUSH HL	;
239.	0912	DD	E1			POP IX	;
240.	0914	E1				POP HL	;Recupera HL para vol-
241.	0915	C3	E4	08		JP BETA	;ver a transmitir.
242.	0918	3E	04		XMSFIN	LD A,\$04	;Envia fin de archivo
243.	091A	CD	3F	01		CALL ESPA	;(<EOT>).
244.	091D	11	85	08	ET20	LD DE,\$0885	;Direcciona mensaje.
245.	0920	3E	00			LD A,\$00	;Codigo para mensaje.
246.	0922	C9				RET	;Termina.
247.	0923	DB	03		ENT	IN A,<UARTST>	;Lee estatus.
248.	0925	CB	4F			BIT 1,A	;Checa si dato listo.
249.	0927	CA	23	09		JP Z,ENT	;Salta si no hay dato.
250.	092A	E6	1C		ALFA	AND \$1C	;Enmascara.
251.	092C	C2	32	09		JP HZ,ERRST	;Salta si error estatus.
252.	092F	DB	02		ALFA1	IN A,<UARTID>	;Lee dato.
253.	0931	C9				RET	;Termina.
254.	0932	11	5F	08	ERRST	LD DE,\$085F	;Direcciona mensaje.

APEND A

Sistema Graficador FAP-100

255.	0935	3E 00		LD A,#00	;Codigo mensaje.
256.	0937	C3 92 08		JP FINAL	;Salta a activar TV.
257.	093A	3A FA 10	RESP	LD A,(BUFL)	;Checa si es comunicacion
258.	093D	FE 00		CP #00	;con cinta.
259.	093F	C8		RET Z	;Regresa si es con cinta.
260.	0940	CD 71 09	IMS	CALL ENTACK	;Lee respuesta.
261.	0943	FE 06		CP #06	;
262.	0945	C8		RET Z	;
263.	0946	FE 15		CP #15	;
264.	0948	C2 40 09		JP NZ,IMS	;
265.	0948	3A FB 10	REPITE	LD A,(BUFB)	;Checa si ha repetido au-
266.	094E	3D		DEC A	;chas veces el mismo blo-
267.	094F	32 FB 10		LD (BUFB),A	;que.
268.	0952	CA 69 09		JP Z,EFB	;Salta si lo ha hecho.
269.	0955	D5		PUSH DE	;Prepara parametros para
270.	0956	11 80 00		LD DE,#0080	;repetir bloque.
271.	0959	A7		AND A	;
272.	095A	ED 52		SBC HL,DE	;HL = HL - 80.
273.	095C	DD 19		ADD IX,DE	;IX = XI + 80.
274.	095E	0D		DEC C	;
275.	095F	D1		POP DE	;
276.	0960	C9		RET	;Termina.
277.	0961	11 75 08	EFA	LD DE,#0875	;Direcciona mensaje.
278.	0964	3E 01		LD A,#01	;Codigo mensaje.
279.	0966	C3 92 08		JP FINAL	;Termina.
280.	0969	11 75 08	EFB	LD DE,#0875	;Direcciona mensaje.
281.	096C	3E 02		LD A,#02	;Codigo mensaje.
282.	096E	C3 92 08		JP FINAL	;Termina.
283.	0971				
284.	0980			** \$0980	
285.	0980				
286.	0980	0E 00	XMENT	LD C,#00	;Inicializa numero bloque.
287.	0982	CD 70 0A	XM1	CALL ENTHP	;Toma 1er. byte de bloque.
288.	0985	FE 01		CP #01	;Checa si es encabezado.
289.	0987	C2 14 0A		JP NZ,ENTEOT	;Salta si no lo es.
290.	098A	CD 23 09		CALL ENT	;Toma numero de bloque.
291.	098D	B9		CP C	;Checa si existe error.
292.	09BE	C2 1C 0A		JP NZ, ERROR1	;Salta si hay error.
293.	09C1	79		LD A,C	;Invierte numero de bloque.
294.	09C2	2F		CPL	;
295.	09C3	4F		LD C,A	;
296.	09C4	CD 23 09		CALL ENT	;Toma no. bloque invertido.
297.	09C7	B9		CP C	;Checa si existe error.
298.	09C8	C2 1C 0A		JP NZ,ERROR1	;Salta si lo hay.
299.	09CB	79		LD A,C	;Invierte numero de bloque.
300.	09CC	2F		CPL	;
301.	09CD	4F		LD C,A	;
302.	09CE	16 80		LD D,#80	;Inicializa no. de datos.
303.	09D0	06 00		LD B,#00	;Inicializa CKSUM.
304.	09D2	CD 23 09	XMENT1	CALL ENT	;Toma dato.
305.	09D5	DS		PUSH DE	;Checa si todavia existe

APEND A

Sistema Graficador FAP-100

306.	09D6	E5			PUSH HL	;espacio en memoria para
307.	09D7	ED	5B FC 10		LD DE,(TOPE)	;datos.
308.	09DB	A7			AND A	;
309.	09DC	ED	52		SBC HL,DE	;
310.	09DE	E1			POP HL	;
311.	09DF	D1			POP DE	;
312.	09E0	F2	2E 0A		JP P,ERROR3	;Salta si no hay espacio.
313.	09E3	77			LD (HL),A	;Guarda dato.
314.	09E4	78			LD A,B	;Actualiza CKSUM.
315.	09E5	86			ADD A,(HL)	;
316.	09E6	47			LD B,A	;
317.	09E7	23			INC HL	;Actualiza apuntador.
318.	09E8	15			DEC D	;Decrementa no. de datos.
319.	09E9	C2 D2 09			JP NZ,XMENT1	;Salta si faltan datos de
320.	09EC					bloque.
321.	09EC	CD 23 09			CALL ENT	;Toma CKSUM.
322.	09EF	B8			CP B	;Checa CKSUM.
323.	09F0	C2 00 0A			JP NZ,XMENT2	;Salta si error.
324.	09F3	3A FA 10			LD A,(BUFL)	;Checa si comunicacion con
325.	09F6	FE 00			CP \$00	;cinta.
326.	09F8	CA 10 0A			JP Z,XMCIN	;Salta si comunicacion con
327.	09FB					cinta.
328.	09FB	3E 06			LD A,\$06	;Envia ACK.
329.	09FD	C3 0D 0A			JP XMIN	;
330.	0A00	3A FA 10	XMENT2		LD A,(BUFL)	;Checa si comunicacion con
331.	0A03	FE 00			CP \$00	;cinta.
332.	0A05	CA 28 0A			JP Z,XMCFIN	;Salta si es con cinta.
333.	0A08	CD 48 09			CALL REPITE	;Establece condicion para
334.	0A08					repetir bloque.
335.	0A08	3E 15			LD A,\$15	;Envia NACK.
336.	0A0D	0C 3F 01	XMIN		CALL ESPA	;
337.	0A10	0C	XMCIN		INC C	;Incrementa no. de bloque.
338.	0A11	C3 B2 09			JP XM1	;Cierra bucle.
339.	0A14	FE 04	ENTEOT		CP \$04	;Checa si fin de archivo.
340.	0A16	C2 22 0A			JP NZ,ERROR2	;Salta si error.
341.	0A19	C3 1D 09			JP ET20	;Termina.
342.	0A1C	11 75 08	ERROR1		LD DE,\$0875	;Direcciona mensaje.
343.	0A1F	3E 03			LD A,\$03	;Codigo mensaje.
344.	0A21	C9			RET	;Termina.
345.	0A22	11 75 08	ERROR2		LD DE,\$0875	;Direcciona mensaje.
346.	0A25	3E 04			LD A,\$04	;Codigo mensaje.
347.	0A27	C9			RET	;Termina.
348.	0A28	11 69 08	XMCFIN		LD DE,\$0869	;Direcciona mensaje.
349.	0A28	3E 00			LD A,\$00	;Codigo mensaje.
350.	0A2D	C9			RET	;Termina.
351.	0A2E	11 75 08	ERROR3		LD DE,\$0875	;Direcciona mensaje.
352.	0A31	3E 05			LD A,\$05	;Codigo mensaje.
353.	0A33	C9			RET	;Termina.
354.	0A34					
355.	09A4				**= \$09A4	
356.	09A4					

APEHO A

Sistema Graficador FAP-100

```

357. 09A4 0E FF      RETD0  LD C,$FF      ;Inicializa contador.
358. 09A6 0D 00      RT1    DEC C        ;Decrementa contador.
359. 09A7 20 01      JR NZ,RT3     ;Salta si no es fin.
360. 09A9 C9         RET        ;Termina.
361. 09AA 57         RT3    LD D,A        ;Toma parametro.
362. 09AB 15         RT2    DEC D        ;Decrementa parametro.
363. 09AC 20 FD      JR NZ,RT2     ;Salta si no es fin.
364. 09AE 18 F6      JR RT1       ;Termina conteo parametro.
365. 09B0           ;
366. 0A63           ;      **= $0A63
367. 0A63           ;
368. 0A63 CD A4 09   NVO    CALL RETD0   ;Genera retardo.
369. 0A66 3E 05     LD A,$05     ;Genera parametro para re-
370. 0A68 32 FB 10  LD (BUFB),A  ;petir bloque.
371. 0A6B 3A 00 11  LD A,(DISP)  ;Toma parametro seleccion
372. 0A6E           ;      ;dispositivo.
373. 0A6E C9       RET        ;Termina.
374. 0A6F 00       NOP        ;
375. 0A70 1E 0A     ENTHP  LD E,$0A     ;Espera respuesta NAK o
376. 0A72 CD 23 09 ENTHP1 CALL ENT     ;AK, dejando pasar todo
377. 0A75 FE 01     CP $01      ;lo que no sea alguna de
378. 0A77 C8       RET Z        ;estas.
379. 0A78 FE 04     CP $04      ;
380. 0A7A C8       RET Z        ;
381. 0A7B 32 01 11 LD ($1101),A ;
382. 0A7E 1D       DEC E        ;
383. 0A7F C8       RET Z        ;
384. 0A80 C3 70 0A JP ENTHP   ;
385. 0A83           ;
386. 0A83           ;
387. 0A83           ;
388. 0A83           ;
389. 0A83           ;
390. 0A83           ;
391. 0A83           ;
392. 0A83           ;
392. 0A83           ;

```

T a b l a d e S i m b o l o s

Nombre	Valor	Linea de Definicion
let7	11B0	6
let8	11B7	7
let9	11CA	8
letec	0136	9
espa	013F	10
esmen	01DC	11
esmen1	189E	12
uartst	0003	13
uartio	0002	14
orgusx	1127	15
orgusy	1129	16
buf1	10FA	17
bufb	10FB	18
tope	10FC	19
disp	1100	20
modo	1102	21



long	112D	22
condir	1124	23
dirl	110B	24
final	0892	25
entack	0971	26
acceta	1CF3	29
acuno	1CFE	33
accos	1D86	82
tracin	1D8E	86
cinuno	1D99	90
cindos	1DEB	124
cintre	1E08	139
finl	1E10	143
xmensa	1E29	158
xtres	1E2C	160
xuno	1E33	163
inicio	0885	171
xmsal	08AA	181
xms1	08AC	182
xms2	08BC	191
xms3	08DD	212
beta	08E4	215
xmsu	08E8	217
xmsu1	08EC	218
xmsu2	08FA	224
xmsfin	0918	242
et20	091D	244
ent	0923	247
alfa	092A	250
alfal	092F	252
errst	0932	254
resp	093A	257
ims	0940	260
repite	094B	265
efa	0961	277
efb	0969	280
xment	09B0	286
xm1	09B2	287
xment1	09D2	304
xment2	0A00	330
xmin	0A0D	336
xmcin	0A10	337
enteot	0A14	339
error1	0A1C	342
error2	0A22	345
xmcfm	0A28	348
error3	0A2E	351
retdo	09A4	357
rt1	09A6	358
rt3	09AA	361
rt2	09AB	362
nvo	0A63	368
enthp	0A70	375
enthp1	0A72	376

APEND A

Sistema Graficador FAP-100

```

1. 0000 ;
2. 0000 ;*** MENU DE MODIFICACIONES ***
3. 0000 ;
4. 0000 ;
5. 1E3B ; ** $1E3B
6. 1E3B LET15 = $121F
7. 1E3B LET16 = $1239
8. 1E3B LET17 = $1254
9. 1E3B LET18 = $1268
10. 1E3B LET19 = $1279
11. 1E3B LET20 = $128E
12. 1E3B LET21 = $12A1
13. 1E3B LET22 = $12B1
14. 1E3B RENPOS = $1144
15. 1E3B LETEC = $0136
16. 1E3B ESPA = $013F
17. 1E3B ESMEN = $01DC
18. 1E3B ESMEN1 = $18BE
19. 1E3B POSC = $C479
20. 1E3B MODADA = $1EB5
21. 1E3B RDOC = $FFFE
22. 1E3B AMP = $FFFD
23. 1E3B DES = $FFFC
24. 1E3B ROT = $FFFB
25. 1E3B ;
26. 1E3B 3E 1A ; MENMOD LD A,$1A ;Manda comando para borrar
27. 1E3D CD 3F 01 CALL ESPA ;pantalla.
28. 1E40 3E 22 LD A,$22 ;Inicializa numero de ren-
29. 1E42 32 44 11 LD (RENPOS),A ;gion de pantalla.
30. 1E45 16 14 LD D,$14 ;Posiciona cursor en pan-
31. 1E47 CD 79 04 CALL POSC ;talla.
32. 1E4A 21 1F 12 LD HL,LET15 ;Despliega mensaje "MENU
33. 1E4D CD BE 18 CALL ESMEN1 ;DE MODIFICACIONES".
34. 1E50 21 39 12 LD HL,LET16 ;Despliega mensaje "1. Mo-
35. 1E53 CD BE 18 CALL ESMEN1 ;dificacion dato a dato".
36. 1E56 21 54 12 LD HL,LET17 ;Despliega mensaje "Esca-
37. 1E59 CD BE 18 CALL ESMEN1 ;lamiento:".
38. 1E5C 21 68 12 LD HL,LET18 ;Despliega mensaje "2. Re-
39. 1E5F CD BE 18 CALL ESMEN1 ;duccion ".
40. 1E62 21 79 12 LD HL,LET19 ;Despliega mensaje "3. Am-
41. 1E65 CD BE 18 CALL ESMEN1 ;plificacion ".
42. 1E68 21 8E 12 LD HL,LET20 ;Despliega mensaje "4.
43. 1E6B CD BE 18 CALL ESMEN1 ;Traslacion ".
44. 1E6E 21 A1 12 LD HL,LET21 ;Despliega mensaje "5. Ro-
45. 1E71 CD BE 18 CALL ESMEN1 ;stacion ".
46. 1E74 21 B1 12 LD HL,LET22 ;Despliega mensaje "6. Sa-
47. 1E77 CD DC 01 CALL ESMEN ;lida ".
48. 1E7A CD 36 01 NADA CALL LETEC ;Recibe opcion elegida.
49. 1E7D FE 31 CP $31 ;Checa si es modificacion
50. 1E7F ; ;dato a dato.

```

51.	1E7F 20 06		JR NZ,MEN1	;Salta si no es.
52.	1E91 CD 85 1E		CALL MODADA	;Llama modificacion dato
53.	1EB4	,		;a dato.
54.	1EB4 C3 3B 1E		JP MENMOD	;Recibe nueva eleccion.
55.	1EB7 FE 32	MEN1	CP #32	;Checa si es Reduccion.
56.	1EB9 20 06		JR NZ,MEN2	;Salta si no es.
57.	1EB8 CD FE FF		CALL Rduc	;Llama reduccion.
58.	1EBE C3 3B 1E		JP MENMOD	;Recibe nueva eleccion.
59.	1E91 FE 33	MEN2	CP #33	;Checa si es Amplificacion.
60.	1E93 20 06		JR NZ,MEN3	;Salta si no es.
61.	1E95 CD FD FF		CALL AMP	;Llama amplificacion.
62.	1E98 C3 3B 1E		JP MENMOD	;Recibe nueva eleccion.
63.	1E98 FE 34	MEN3	CP #34	;Checa si es Translacion.
64.	1E9D 20 06		JR NZ,MEN4	;Salta si no es.
65.	1E9F CD FC FF		CALL DES	;Llama a Desplazamiento.
66.	1EA2 C3 3B 1E		JP MENMOD	;Recibe nueva eleccion.
67.	1EA5 FE 35	MEN4	CP #35	;Checa si es Rotacion.
68.	1EA7 20 06		JR NZ,MEN5	;Salta si no es.
69.	1EA9 CD FB FF		CALL ROT	;Llama a rotacion.
70.	1EAC C3 3B 1E		JP MENMOD	;Recibe nueva eleccion.
71.	1EAF FE 36	MEN5	CP #36	;Checa si es Salida.
72.	1EB1 C2 7A 1E		JP NZ,NADA	;Salta si no es ninguna
73.	1EB4	,		;eleccion.
74.	1EB4 C9		RET	;Termina.
74.	1EB5			

T a b l a d e S i m b o l o s

Nombre	Valor	Linea de Definicion
let15	121F	6
let16	1238	7
let17	1254	8
let18	1268	9
let19	1279	10
let20	128E	11
let21	12A1	12
let22	12B1	13
renpos	1144	14
letec	0136	15
espa	013F	16
esmen	01DC	17
esmen1	188E	18
posc	C479	19
modada	1EB5	20
rduc	FFFFE	21
amp	FFFFD	22
des	FFFFC	23
rot	FFFFB	24
menmod	1E3B	26
nada	1E7A	48
men1	1E87	55
men2	1E91	59
men3	1E9B	63
men4	1EA5	67
men5	1EAF	71

APEND A

Sistema Graficador FAP-100

```

1. 0000 )
2. 0000 )** MODIFICACION DATO A DATO ***
3. 0000 )
4. 0000 )
5. 1EB5 *= $1EB5
6. 1EB5 LET1 = $1145
7. 1EB5 LET2 = $115A
8. 1EB5 LET10 = $11D1
9. 1EB5 LETEC = $0136
10. 1EB5 ESPA = $013F
11. 1EB5 ESMEN1 = $10BE
12. 1EB5 LLENA = $19C4
13. 1EB5 INDIR = $19DD
14. 1EB5 GUADIR = $1A09
15. 1EB5 DATO = $1AB4
16. 1EB5 PLANT = $1A37
17. 1EB5 CHECA = $1AD0
18. 1EB5 FIN = $195D
19. 1EB5 INCREN = $1AD8
20. 1EB5 GUADAS = $1BA6
21. 1EB5 GUACON = $1C3D
22. 1EB5 BIASC = $1A4A
23. 1EB5 POSC = $C479
24. 1EB5 INCR = $C401
25. 1EB5 CRLF = $C53A
26. 1EB5 DIR1 = $110B
27. 1EB5 CONDIR = $1124
28. 1EB5 COL = $1126
29. 1EB5 RENG = $112B
30. 1EB5 RENPOS = $1144
31. 1EB5 ;
32. 1EB5 ;
33. 1EB5 3E 1A MODADA LD A,$1A ;Manda comando para borrar
34. 1EB7 CD 3F 01 CALL ESPA ;pantalla.
35. 1EBA CD C4 19 CALL LLENA ;Prepara espacio para dato.
36. 1EBD 21 45 11 LD HL,LET1 ;Despliega mensaje "DIREC-
37. 1EC0 CD BE 18 CALL ESMEN1 ;CION INICIAL ;".
38. 1EC3 CD DD 19 CALL INDIR ;Introduce direc. inicial.
39. 1EC6 CD 09 1A CALL GUADIR ;Almacena direc. inicial.
40. 1EC9 CD 3A C5 CALL CRLF ;Salta renglon.
41. 1ECC CD 3A C5 CALL CRLF ;Salta renglon.
42. 1ECC CD 3A C5 CALL CRLF ;Salta renglon.
43. 1E02 CD C4 19 CALL LLENA ;Prepara espacio para dato.
44. 1ED5 3E 23 LD A,$23 ;Genera numero de renglon
45. 1ED7 32 44 11 LD <RENPOS>,A ;para posicionar cursor.
46. 1EDA 21 2B 11 LD HL,RENG ;Apunta localidad para al-
47. 1EDD 22 24 11 LD <CONDIR>,HL ;macenar no. renglon a mo-
48. 1EE0 ; ;dificarse.
49. 1EE0 21 D1 11 LD HL,LET10 ;Despliega mensaje "RENGLON
50. 1EE3 CD BE 18 CALL ESMEN1 ;NUMERO? ".

```

APEND A

Sistema Graficador FAP-100

51.	IEE6	21	5A	11	LD HL,LET2	;Despliega numero de entra-
52.	IEE9	CD	BE	18	CALL ESMEN1	;da.
53.	IEEC	3E	11		LD A,\$11	;Prepara no. columna a po-
54.	IEEE					;sicionarse.
55.	IEEE	CD	B4	1A	CALL DATO	;Introduce y almacena dato.
56.	IEF1	CD	66	1F	CALL CALCRE	;Calcula no. renglon a mo-
57.	IEF4					;dificarse.
58.	IEF4	CD	82	1F	CALL ACTCON	;Actualiza apuntador.
59.	IEF7	3E	1A		LD A,\$1A	;Manda comando para borrar
60.	IEF9	CD	3F	01	CALL ESPA	;pantalla.
61.	IEFC	CD	37	1A	CALL PLANT	;Despliega plantilla guia.
62.	IEFF	CD	01	C4	CALL INCR	;Actualiza apuntador de
63.	1F02					;renglones de pantalla.
64.	1F02	CD	0B	1F	NUEVA CALL DESREN	;Despliega informacion ren-
65.	1F05					;gion a modificarse.
66.	1F05	3E	0B		LD A,\$0B	;Prepara no. de columna.
67.	1F07	CD	F2	1F	CALL MODATO	;Modifica dato.
68.	1F0A	3E	16		LD A,\$16	;Prepara no. columna.
69.	1F0C	CD	F2	1F	CALL MODATO	;Modifica dato.
70.	1F0F	3E	24		LD A,\$24	;Prepara no. columna.
71.	1F11	CD	00	20	CALL MODCON	;Modifica control.
72.	1F14	CD	D0	1A	CALL CHECA	;Checa si es fin tabla.
73.	1F17	2B	04		JR Z,SIGUE	;Salta si no es fin.
74.	1F19	CD	5D	19	CALL FIN	;Termina.
75.	1F1C	C9			RET	;
76.	1F1D	FE	19		SIGUE CP \$19	;Checa si es fin modifi-
77.	1F1F					;cacion.
78.	1F1F	20	04		JR NZ,SIGUEN	;Salta si continua.
79.	1F21	CD	1A	20	CALL TERM	;Termina.
80.	1F24	C9			RET	;
81.	1F25	3E	2D		SIGUEN LD A,\$2D	;Prepara no. columna.
82.	1F27	CD	F2	1F	CALL MODATO	;Modifica dato.
83.	1F2A	3E	38		LD A,\$38	;Prepara columna.
84.	1F2C	CD	F2	1F	CALL MODATO	;Modifica dato.
85.	1F2F	3E	46		LD A,\$46	;Prepara columna.
86.	1F31	CD	00	20	CALL MODCON	;Modifica control.
87.	1F34	CD	D0	1A	CALL CHECA	;Checa si es fin tabla.
88.	1F37	2B	04		JR Z,SIGUE1	;Salta si no es fin.
89.	1F39	CD	5D	19	CALL FIN	;Termina.
90.	1F3C	C9			RET	;
91.	1F3D	FE	19		SIGUE1 CP \$19	;Checa si fin modificacion.
92.	1F3F	20	04		JR NZ,SIGUE2	;Salta si continua.
93.	1F41	CD	1A	20	CALL TERM	;Termina.
94.	1F44	C9			RET	;
95.	1F45	CD	3A	C5	SIGUE2 CALL CRLF	;Accesa nuevo renglon.
96.	1F48	CD	01	C4	CALL INCR	;Actualiza contador de
97.	1F48					;renglones pantalla.
98.	1F4B	CD	D8	1A	CALL INCR	;Incrementa numero de
99.	1F4E					;renglon.
100.	1F4E	3A	44	11	LD A,(RENPOS)	;Checa si pantalla llena.
101.	1F51	FE	37		CP \$37	;

APEND A

Sistema Graficador FAP-100

102.	1F53	C2	02	1F	JP NZ,NUEVA	;Salta si no llena.
103.	1F56	3E	1E		LD A,#1E	;Manda cursor a posicion
104.	1F53	CD	3F	01	CALL ESPA	;hoee.
105.	1F53	CD	37	1A	CALL PLANT	;Despliega plantilla guia.
106.	1F5E	16	00		LD D,\$00	;Regreza el cursor a la po-
107.	1F60	CD	79	C4	CALL POSC	sicion anterior.
108.	1F63	C3	02	1F	JP NUEVA	;Prosigue modificacion.
109.	1F66	ED	5B	2B	LD DE,<RENG>	;Toma numero de renglon.
110.	1F6A	21	00	00	LD HL,\$0000	;Prepara registro para pro-
111.	1F6D					ducto multiplicacion x 10.
112.	1F6D	01	6A	00	LD BC,\$000A	;Prepara multiplicador.
113.	1F70	C5			PUSH BC	;Preserva multiplicador.
114.	1F71	19			ADD HL,DE	;Obtiene producto parcial.
115.	1F72	0D			DEC C	;Decrementa contador.
116.	1F73	79			LD A,C	;Toma contador.
117.	1F74	FE	00		CP \$00	;Checa si fin multiplica-
118.	1F76					cion.
119.	1F76	20	F9		JR NZ,REUNO	;Salta si no es fin.
120.	1F78	C1			POP BC	;Retoma factor 10.
121.	1F79	A7			AND A	;Termina calculo de direc-
122.	1F7A	ED	42		SBC HL,BC	cion de dato a modificar-
123.	1F7C					se.
124.	1F7C	3E	20		LD A,\$20	;Prepara numero de renglon
125.	1F7E	32	44	11	LD <RENPOS>,A	;pantalla inicial.
126.	1F81	C9			RET	;Termina.
127.	1F82	ED	5B	0B	LD DE,<DIR1>	;Toma direccion inicial.
128.	1F86	19			ADD HL,DE	;Obtiene direccion final.
129.	1F87	22	24	11	LD <CONDIR>,HL	;Actualiza apuntador.
130.	1F8A	C9			RET	;Termina.
131.	1F8B	ED	4B	2B	LD BC,<RENG>	;Toma numero renglon.
132.	1F8F	CD	2C	20	CALL CONDES	;Despliega numero renglon.
133.	1F92	16	0B		LD D,\$0B	;Prepara numero columna.
134.	1F94	CD	79	C4	CALL POSC	;Posiciona cursor.
135.	1F97	2A	24	11	LD HL,<CONDIR>	;Toma direccion dato a
136.	1F9A					modificarse.
137.	1F9A	CD	36	20	CALL BCHL	;Toma dato.
138.	1F9D	E5			PUSH HL	;Preserva direccion.
139.	1F9E	CD	2C	20	CALL CONDES	;Despliega dato.
140.	1FA1	16	16		LD D,\$16	;Prepara columna.
141.	1FA3	CD	79	C4	CALL POSC	;Posiciona cursor.
142.	1FA6	E1			POP HL	;Retoma direccion.
143.	1FA7	CD	36	20	CALL BCHL	;Toma dato.
144.	1FAA	E5			PUSH HL	;Preserva direccion.
145.	1FAB	CD	2C	20	CALL CONDES	;Despliega dato.
146.	1FAE	16	24		LD D,\$24	;Prepara no. columna.
147.	1FB0	CD	79	C4	CALL POSC	;Posiciona cursor.
148.	1FB3	E1			POP HL	;Retoma direccion.
149.	1FB4	7E			LD A,<HL>	;Toma control.
150.	1FB5	23			INC HL	;Actualiza apuntador.
151.	1FB6	E5			PUSH HL	;Preserva direccion.
152.	1FB7	CB	7F		BIT 7,A	;Checa fin de tabla.

APEND A

Sistema Graficador FAP-100

153.	1FB9	28	03		JR Z,NUM	;Salta si no es fin.
154.	1FBB	E1			POP HL	;Retoma direccion.
155.	1FBC	18	2A		JR NUM2	;Salta si es fin.
156.	1FBE	C6	30		ADD A,\$30	
157.	1FC0	CD	3F	01	CALL ESPA	;Prepara codigo ASCII.
158.	1FC3	16	2D		LD D,\$2D	;Despliega control.
159.	1FC5	CD	79	C4	CALL POSC	;Prepara no. columna.
160.	1FC8	E1			POP HL	;Posiciona cursor.
161.	1FC9	CD	36	20	CALL BCHL	;Retoma direccion.
162.	1FCC	E5			PUSH HL	;Toma dato.
163.	1FCD	CD	2C	20	CALL CONDES	;Preserva direccion.
164.	1FD0	16	38		LD D,\$38	;Despliega dato.
165.	1FD2	CD	79	C4	CALL POSC	;Prepara no. columna.
166.	1FD5	E1			POP HL	;Posiona cursor.
167.	1FD6	CD	36	20	CALL BCHL	;Retoma direccion.
168.	1FD9	E5			PUSH HL	;Toma dato.
169.	1FDA	CD	2C	20	CALL CONDES	;Preserva direccion.
170.	1FDD	16	46		LD D,\$46	;Despliega dato.
171.	1FDF	CD	79	C4	CALL POSC	;Prepara no. columna.
172.	1FE2	E1			POP HL	;Posiciona cursor.
173.	1FE3	7E			LD A,(HL)	;Retoma direccion.
174.	1FE4	CB	7F		BIT 7,A	;Toma control.
175.	1FE6	28	04		JR Z,NUM1	;Checa fin de tabla.
176.	1FE8	3E	58		LD A,\$58	;Salta si no es fin tabla.
177.	1FEA	18	02		JR MANDA1	;Prepara letra X.
178.	1FEC	C6	30		ADD A,\$30	;Manda fin de tabla.
179.	1FEE	CD	3F	01	CALL ESPA	;Prepara codigo ASCII.
180.	1FF1	C9			RET	;Despliega control.
181.	1FF2	32	26	11	LD (COL),A	;Termina.
182.	1FF5	57			LD D,A	;Almacena no. columna.
183.	1FF6	CD	79	C4	CALL POSC	;Prepara no. columna.
184.	1FF9	CD	3B	20	CALL MODA	;Posiciona cursor.
185.	1FFC	CD	A6	1B	CALL GUADAS	;Modifica dato.
186.	1FFF	C9			RET	;Almacena dato.
187.	2000	32	26	11	LD (COL),A	;Termina.
188.	2003	57			LD D,A	;Almacena no. columna.
189.	2004	CD	79	C4	CALL POSC	;Prepara no. columna.
190.	2007	CD	84	20	CALL MODICO	;Posiciona cursor.
191.	200A	FE	19		CP \$19	;Modifica control.
192.	200C	28	04		JR Z,TERCON	;Checa fin modificacion.
193.	200E	CD	3D	1C	CALL GUACON	;Salta si es fin.
194.	2011	C9			RET	;Almacena control.
195.	2012	2A	24	11	LD HL,(CONDIR)	;Termina.
196.	2015	23			INC HL	;Toma apuntador.
197.	2016	22	24	11	LD (CONDIR),HL	;Actualiza apuntador.
198.	2019	C9			RET	;Almacena apuntador.
199.	201A	11	05	00	LD DE,\$0005	;Termina.
200.	201D	CD	00	1A	CALL CHECA	;Prepara registro.
201.	2020	20	06		JR NZ,FIT	;Checa si es fin de tabla.
202.	2022	19			ADD HL,DE	;Salta si fin de tabla.
203.	2023					;Accesa siguiente pareja de datos.

APEND A

Sistema Graficador FAP-100

204.	2023	22	24	11		LD <CONDIR>,HL	;Actualiza apuntador.
205.	2026	18	F5			JR TERUNO	;Prosigue busqueda final.
206.	2028	CD	5D	19	FIT	CALL FIN	;Genera y almacena para-
207.	2028						;metros tabla.
208.	2028	C9				RET	;Termina.
209.	202C	CD	4A	1A	CONDES	CALL BIASC	;Obtiene equivalente dec-
210.	202F						;cimal en base a codigos
211.	202F						;ASCII.
212.	202F	21	5A	11		LD HL,LET2	;Despliega dato en panta-
213.	2032	CD	BE	18		CALL ESMEN1	;lla.
214.	2035	C9				RET	;Termina.
215.	2036	4E			BCHL	LD C,<HL>	;Toma parte baja de dato.
216.	2037	23				INC HL	;Actualiza apuntador.
217.	2038	46				LD B,<HL>	;Toma parte alta de dato.
218.	2039	23				INC HL	;Actualiza apuntador.
219.	203A	C9				RET	;Termina.
220.	203B	2A	24	11	MODA	LD HL,<CONDIR>	;Toma apuntador de tabla.
221.	203E	CD	36	20		CALL BCHL	;Toma dato.
222.	2041	CD	4A	1A		CALL BIASC	;Prepara formato ASCII.
223.	2044	CD	36	01	DATUNO	CALL LETEC	;Recibe digito.
224.	2047	FE	20			CP #20	;Checa fin de modificacion.
225.	2049	C8				RET Z	;Termina si es fin.
226.	204A	FE	2D			CP #2D	;Checa si es negativo.
227.	204C	20	05			JR NZ,LAOTRA	;Salta si positivo.
228.	204E	FD	77	00		LD <IY+0>,A	;Establece signo negativo.
229.	2051	18	21			JR DADOS	;Despliega dato.
230.	2053	FE	2B		LAOTRA	CP #2B	;Checa si es positivo.
231.	2055	20	06			JR NZ,C30	;Salta si no es positivo.
232.	2057	FD	36	00	20	LD <IY+0>,\$20	;Establece signo positivo.
233.	2058	18	17			JR DADOS	;Despliega dato.
234.	205D	FE	30		C30	CP #30	;Establece filtro decimal
235.	205F	FA	44	20		JP M,DATUNO	;para dato introducido
236.	2062	FE	3A			CP #3A	;desde teclado.
237.	2064	F2	44	20		JP P,DATUNO	
238.	2067	21	5C	11		LD HL,\$115C	;Prepara espacio para
239.	206A	11	5B	11		LD DE,\$115B	;nuevo digito.
240.	206D	01	03	00		LD BC,\$0003	
241.	2070	ED	B0			LDIR	
242.	2072	2B				DEC HL	
243.	2073	77				LD <HL>,A	;Almacena digito.
244.	2074	21	5A	11	DADOS	LD HL,LET2	;Despliega nuevo valor de
245.	2077	CD	BE	18		CALL ESMEN1	;dato.
246.	207A	3A	26	11		LD A,<COL>	;Toma no. columna.
247.	207D	57				LD D,A	;Prepara no. columna.
248.	207E	CD	79	C4		CALL POSC	;Posiciona cursor.
249.	2081	C3	44	20		JP DATUNO	;Recibe nuevo digito.
250.	2084	7E			MODICO	LD A,<HL>	;Toma control.
251.	2085	FD	21	5A	11	LD IY,LET2	;Direcciona espacio para
252.	2089						;dato.
253.	2089	CB	7F			BIT 7,A	;Checa si fin de tabla.
254.	208B	20	02			JR NZ,COUNO	;Salta si fin de tabla.



APEND A

Sistema Graficador FAP-100

255.	208D	C6 30		ADD A,#30	;Prepara codigo ASCII.
256.	208F	FD 77 03	COUND	LD (IY+3),A	;Almacena control.
257.	2092	CD 36 01	COCUAT	CALL LETEC	;Recibe nuevo control.
258.	2095	FE 20		CP #20	;Checa si fin modificacion.
259.	2097	FE 20		JR Z,CODOS	;Salta si fin modificacion.
260.	2099	FE 18		CP #18	;Checa si se genera un nuevo fin de tabla.
261.	209B				
262.	209B	28 24		JR Z,COTRES	;Salta si fin de tabla.
263.	209D	FE 19		CP #19	;Checa si fin modificacion total.
264.	209F				
265.	209F	C8		RET Z	;Termina si es cierto.
266.	20A0	FE 30		CP #30	;Establece filtro para recibir unicamente palabras de control validas.
267.	20A2	FA 92 20		JP M,COCUAT	
268.	20A5	FE 33		CP #33	
269.	20A7	F2 92 20		JP P,COCUAT	
270.	20AA	FD 77 03		LD (IY+3),A	;Almacena palabra valida.
271.	20AD	CD 3F 01		CALL ESPA	;Despliega nuevo control.
272.	20B0	3A 26 11		LD A,<COL>	;Toma no. columna.
273.	20B3	57		LD 0,A	;Prepara no. columna.
274.	20B4	CD 79 C4		CALL POSC	;Posiciona cursor.
275.	20B7	18 09		JR COCUAT	;Recibe nuevo control.
276.	20B9	FD CB 03	7E CODOS	BIT 7,<IY+3>	;Checa si es fin de tabla.
277.	20BD	C9		RET Z	;Termina si es cierto.
278.	20BE	3E 19		LD A,#19	;Prepara codigo para fin de modificacion total.
279.	20C0				
280.	20C0	C9		RET	;Termina.
281.	20C1	3E 58	COTRES	LD A,#58	;Prepara codigo fin tabla.
282.	20C3	CD 3F 01		CALL ESPA	;Despliega codigo final.
283.	20C6	C9		RET	;Termina.
283.	20C7				

T a b l a d e S i m b o l o s

Nombre	Valor	Linea de Definicion
let1	1145	6
let2	115A	7
let10	11D1	8
letec	0136	9
espa	013F	10
esmen1	18BE	11
llena	19C4	12
indir	19D0	13
guadir	1A09	14
dato	1AB4	15
plant	1A37	16
checa	1AD0	17
fin	195D	18
incres	1A08	19
guadas	1BA6	20
guacon	1C3D	21
biasec	1A4A	22
posc	C479	23
incr	C401	24
ctrlf	C53A	25
dir1	1108	26
condir	1124	27
col	1126	28

reng	112B	29
renpos	1144	30
modada	1EB5	33
nueva	1F02	64
sigue	1F10	76
siguen	1F25	81
sigue1	1F30	91
sigue2	1F45	95
calcre	1F66	109
reuno	1F71	114
actcon	1F82	127
desren	1F8B	131
num	1FBE	156
manda	1FC0	157
num2	1FE8	176
num1	1FEC	178
mandal	1FEE	179
modato	1FF2	181
modcon	2000	187
tercon	2012	195
term	201A	199
teruno	201D	200
fit	2028	206
condes	202C	209
bchl	2036	215
moda	203B	220
datuno	2044	223
laotra	2053	230
c30	205D	234
dados	2074	244
modico	2084	250
couno	208F	256
cocuat	2092	257
codos	20B9	276
cotres	20C1	281

APEND A

Sistema Graficador FAP-100

```

1. 0000 ;
2. 0000 ;*** REDUCCION, AMPLIFICACION Y DESPLAZAMIENTO ***
3. 0000 ;
4. 0000 ;
5. 20C7 * = $20C7
6. 20C7 LET1 = $1145
7. 20C7 LET2 = $115A
8. 20C7 LET7 = $11B0
9. 20C7 LET9 = $11CA
10. 20C7 LET12 = $11E9
11. 20C7 LET13 = $11F5
12. 20C7 LET14 = $1201
13. 20C7 LETEC = $0136
14. 20C7 ESPA = $013F
15. 20C7 ESMEN = $01DC
16. 20C7 ESMEN1 = $18BE
17. 20C7 GUADAS = $18A6
18. 20C7 LLENA = $19C4
19. 20C7 INDIR = $19DD
20. 20C7 QUADIR = $1A09
21. 20C7 NUEDAT = $1CD7
22. 20C7 NUECON = $1CE4
23. 20C7 SUMIYS = $1CED
24. 20C7 CONY = $1BF8
25. 20C7 DIV = $1C58
26. 20C7 1148 = $1CC1
27. 20C7 POSC = $C479
28. 20C7 INCR = $C401
29. 20C7 CRLF = $C53A
30. 20C7 ORGUSX = $1127
31. 20C7 ORGUSY = $1129
32. 20C7 ORGX = $1135
33. 20C7 ORGY = $1137
34. 20C7 FACTX = $1131
35. 20C7 FACTY = $1133
36. 20C7 DIR1 = $1109
37. 20C7 CONDIR = $1124
38. 20C7 COL = $1126
39. 20C7 DIVOR = $112F
40. 20C7 OP = $1139
41. 20C7 RENPOS = $1144
42. 20C7 ;
43. 20C7 ;
44. 20C7 3E 00 RDUC LD A,$00
45. 20C9 32 39 11 LD (OP),A ;Genera codigo identifica-
46. 20CC CD 0F 21 CALL ESCALA ;cion y lo almacena.
47. 20CF C9 RET ;Transforma tabla.
48. 20D0 3E 01 AMP LD A,$01 ;Termina.
49. 20D2 32 39 11 LD (OP),A ;Genera codigo identifica-
50. 20D5 CD 0F 21 CALL ESCALA ;cion y lo almacena.
;Transforma tabla.

```

APEND A

Sistema Graficador FAP-100

51.	20D8	C9		RET	;Termina.
52.	20D9	3E 02	DES	LD A,#02	;Define codigo de identifi-
53.	20DB	32 39 11		LD <OP>,A	;ficacion.
54.	20DE	CD 0F 21		CALL ESCALA	;Realiza transformacion.
55.	20E1	01 05 00		LD BC,#0005	;Prepara factor de escala-
56.	20E4	ED 43 2F 11		LD <DIVOR>,BC	;miento.
57.	20E8	2A 31 11		LD HL,<FACTX>	;Toma factor eje X.
58.	20EB	E5		PUSH HL	;Preserva factor.
59.	20EC	CD 5B 1C		CALL DIV	;Realiza escalamiento.
60.	20EF	D1		POP DE	;Retoma signo factor.
61.	20F0	CD C1 1C		CALL 1148	;Realiza compensacion.
62.	20F3	EB		EX DE,HL	;Preserva factor escalado.
63.	20F4	2A 35 11		LD HL,<ORGX>	;Toma coordenada X origen.
64.	20F7	19		ADD HL,DE	;Actualiza coordenada.
65.	20F8	22 35 11		LD <ORGX>,HL	;Almacena coordenada.
66.	20FB	2A 33 11		LD HL,<FACTY>	;Toma factor eje Y.
67.	20FE	E5		PUSH HL	;Preserva factor.
68.	20FF	CD 5B 1C		CALL DIV	;Realiza escalamiento.
69.	2102	D1		POP DE	;Retoma signo factor.
70.	2103	CD C1 1C		CALL 1148	;Realiza compensacion.
71.	2106	EB		EX DE,HL	;Preserva factor escalado.
72.	2107	2A 37 11		LD HL,<ORGY>	;Toma coordenada Y origen.
73.	210A	19		ADD HL,DE	;Actualiza coordenada.
74.	210B	22 37 11		LD <ORGY>,HL	;Almacena coordenada.
75.	210E	C9		RET	;Termina.
76.	210F	CD 34 21	ESCALA	CALL REQPAR	;Introduce parametros.
77.	2112	CD 36 01	SCUNO	CALL LETEC	;Recibe seleccion de tabla
78.	2115				;a modificarse.
79.	2115	FE 4E		CP #4E	;Checa si modifica tabla
80.	2117				;maquina.
81.	2117	28 17		JR Z,SCDOS	;Salta si se modifica.
82.	2119	FE 53		CP #53	;Checa si modifica tabla
83.	211B				;usuario.
84.	211B	20 F5		JR NZ,SCUNO	;Salta si no recibe contes-
85.	211D				;tacion.
86.	211D	CD 86 21		CALL ESCUSU	;Transforma tabla usuario.
87.	2120	CD A1 22		CALL DEFORI	;Define origen.
88.	2123	FD 2A 0B 11		LD IY,<DIR1>	;Toma direccion inicial.
89.	2127	FD 6E FA		LD L,<IY-6>	;Toma parte baja direccion
90.	212A				;final.
91.	212A	FD 66 FB		LD H,<IY-5>	;Toma parte alta direccion
92.	212D				;final.
93.	212D	22 0B 11		LD <DIR1>,HL	;Actualiza dir. inicial.
94.	2130	CD D9 21	SCDOS	CALL ESCMAQ	;Transforma tabla maquina.
95.	2133	C9		RET	;Termina.
96.	2134	3E 1A	REQPAR	LD A,#1A	;Manda comando para borrar
97.	2136	CD 3F 01		CALL ESPA	;pantalla.
98.	2139	3E 20		LD A,#20	;Prepara numero de renglon
99.	213B	32 44 11		LD <RENPOS>,A	;de pantalla.
100.	213E	3E 0B		LD A,#0B	;Prepara no. columna.
101.	2140	32 26 11		LD <COL>,A	

APEND. A

Sistema Graficador FAP-100

102.	2143	21	E9	11	LD HL,LET12	;Despliega mensaje "FACTOR
103.	2146	CD	BE	18	CALL ESMEN1	;X? "
104.	2149	CD	3A	22	CALL INFAC	;Introduce factor.
105.	214C	21	31	11	LD HL,FACTX	;Direcciona espacio para
106.	214F	22	24	11	LD (CONDIR),HL	;factor.
107.	2152	CD	A6	1B	CALL GUADAS	;Almacena factor.
108.	2155	CD	3A	C5	CALL CRLF	;Salta renglon.
109.	2158	CD	3A	C5	CALL CRLF	;Salta renglon.
110.	215B	CD	01	C4	CALL INCR	;Actualiza no. renglon.
111.	215E	CD	01	C4	CALL INCR	;
112.	2161	21	F5	11	LD HL,LET13	;Despliega mensaje "FACTOR
113.	2164	CD	BE	18	CALL ESMEN1	;Y? "
114.	2167	CD	3A	22	CALL INFAC	;Introduce factor.
115.	216A	21	33	11	LD HL,FACTY	;Direcciona espacio para
116.	216D	22	24	11	LD (CONDIR),HL	;factor.
117.	2170	CD	A6	1B	CALL GUADAS	;Almacena factor.
118.	2173	CD	3A	C5	CALL CRLF	;Salta renglon.
119.	2176	CD	3A	C5	CALL CRLF	;Salta renglon.
120.	2179	CD	01	C4	CALL INCR	;Actualiza no. renglon.
121.	217C	CD	01	C4	CALL INCR	;
122.	217F	21	01	12	LD HL,LET14	;Despliega mensaje "MODI-
123.	2182	CD	BE	18	CALL ESMEN1	;FICA TABLA USUARIO?
124.	2185				;	;(S/N) "
125.	2185	C9			RET	;Termina.
126.	2186	CD	3A	C5	CALL CRLF	;Salta renglon.
127.	2189	3E	1A		LD A,\$1A	;Manda comando para borrar
128.	218B	CD	3F	01	CALL ESPA	;pantalla.
129.	218E	21	45	11	LD HL,LET1	;Despliega mensaje "DIREC-
130.	2191	CD	BE	18	CALL ESMEN1	;CION INICIAL: "
131.	2194	CD	C4	19	CALL LLENA	;Prepara espacio para dato.
132.	2197	CD	DD	19	CALL INDIR	;Introduce dir. inicial.
133.	219A	CD	09	1A	CALL GUADIR	;Almacena dir. inicial.
134.	219D	CD	3A	C5	CALL CRLF	;Salta renglon.
135.	21A0	21	B0	11	LD HL,LET7	;Despliega mensaje
136.	21A3	CD	DC	01	CALL ESMEN	;"TRANS. "
137.	21A6	FD	2A	0B	LD IY,(DIR1)	;Apunta inicio tabla.
138.	21AA	2A	31	11	LD HL,(FACTX)	;Toma factor eje X.
139.	21AD	22	2F	11	LD (DIVOR),HL	;Prepara factor.
140.	21B0	FD	6E	00	LD L,(IY+0)	;Toma parte baja dato.
141.	21B3	FD	66	01	LD H,(IY+1)	;Toma parte alta dato.
142.	21B6	CD	81	22	CALL DEFOP	;Transforma dato.
143.	21B9	CD	D7	1C	CALL HUEDAT	;Almacena dato.
144.	21BC	2A	33	11	LD HL,(FACTY)	;Toma factor eje Y.
145.	21BF	22	2F	11	LD (DIVOR),HL	;Prepara factor.
146.	21C2	FD	6E	02	LD L,(IY+2)	;Toma parte baja dato.
147.	21C5	FD	66	03	LD H,(IY+3)	;Toma parte alta dato.
148.	21C8	CD	81	22	CALL DEFOP	;Transforma dato.
149.	21CB	CD	D7	1C	CALL HUEDAT	;Almacena dato.
150.	21CE	CD	E4	1C	CALL NUECON	;Copia control.
151.	21D1	CB	7F		BIT 7,A	;Checa si fin de tabla.
152.	21D3	C0			RET N2	;Termina si es fin tabla.

ESCUSU

USDOS

APERD A

Sistema Graficador FAP-100

153.	2104	CD	ED	1C		CALL SUMIYS	;Accesa siguiente pareja
154.	2107						;de datos.
155.	2107	18	D1			JR USDOS	;Prosigue transformacion.
156.	2109	CD	3A	C5	ESCHAQ	CALL CRLF	;Salta renglon.
157.	21DC	21	B0	11		LD HL,LET7	;Despliega mansaje
158.	21DF	CD	DC	01		CALL ESMEN	; "TRANS. "
159.	21E2	FD	2A	0B 11		LD IY,(DIR1)	;Actualiza apuntador de
160.	21E6	FD	22	24 11		LD (CONDIR),IY	;tabla.
161.	21EA	ED	5B	35 11	SIMAQ	LD DE,(ORGX)	;Toma coordenada X origen.
162.	21EE	D5				PUSH DE	;Preserva coordenada.
163.	21EF	2A	31	11		LD HL,(FACTX)	;Toma factor eje X.
164.	21F2	22	2F	11		LD (DIVOR),HL	;Prepara factor.
165.	21F5	FD	6E	00		LD L,(IY+0)	;Toma parte baja dato.
166.	21F8	A7				AND A	;Translada dato a origen
167.	21F9	ED	52			SBC HL,DE	;absoluto.
168.	21FB	CD	D7	22		CALL DEFOPM	;Transforma dato.
169.	21FE	D1				POP DE	;Desplaza dato a origen
170.	21FF	19				ADD HL,DE	;relativo inicial.
171.	2200	CD	D7	1C		CALL NUEDAT	;Almacena dato.
172.	2203	ED	5B	37 11		LD DE,(ORGY)	;Toma coordenada Y origen.
173.	2207	D5				PUSH DE	;Preserva coordenada.
174.	2208	2A	33	11		LD HL,(FACTY)	;Toma factor eje Y.
175.	2208	22	2F	11		LD (DIVOR),HL	;Prepara factor.
176.	220E	FD	6E	02		LD L,(IY+2)	;Toma parte baja dato.
177.	2211	FD	6E	03		LD H,(IY+3)	;Toma parte alta dato.
178.	2214	A7				AND A	;Translada dato a origen
179.	2215	ED	52			SBC HL,DE	;absoluto.
180.	2217	CD	D7	22		CALL DEFOPM	;Transforma dato.
181.	221A	D1				POP DE	;Desplaza dato a origen
182.	221B	19				ADD HL,DE	;relativo inicial.
183.	221C	CD	D7	1C		CALL NUEDAT	;Almacena dato.
184.	221F	CD	E4	1C		CALL NUECON	;Copia palabra control.
185.	2222	CB	7F			BIT 7,A	;Checa si fin tabla.
186.	2224	20	06			JR NZ,MAFIN	;Salta si fin tabla.
187.	2226	CD	ED	1C		CALL SUMIYS	;Accesa siguiente pareja
188.	2229						;de datos.
189.	2229	C3	EA	21		JP SIMAQ	;Posigue transformacion.
190.	222C	21	CA	11	MAFIN	LD HL,LET9	;Despliega mensaje
191.	222F	CD	DC	01		CALL ESMEN	; "LISTO &".
192.	2232	CD	36	01	NOCR	CALL LETEC	;Recibe comando.
193.	2235	FE	0D			CP #0D	;Checa si es comando
194.	2237						;regreso.
195.	2237	20	F9			JR NZ,NOCR	;Salta si no es.
196.	2239	C9				RET	;Termina.
197.	223A	CD	C4	19	INFAC	CALL LLENA	;Prepara espacio para dato.
198.	223D	FD	36	04 31		LD (IY+4),#31	;Genera valor de entrada.
199.	2241	21	5A	11	FACUHO	LD HL,LET2	;Despliega valor de entrada
200.	2244	CD	BE	18		CALL ESMEN1	;da para factor.
201.	2247	3A	26	11		LD A,(COL)	;Toma no. columna.
202.	224A	57				LD D,A	;Prepara no. columna.
203.	224B	CD	79	C4		CALL POSC	;Posiciona cursor.

APEND A

Sistema Graficador FAP-100

204.	224E	CD	36	01	FACDOS	CALL LETEC	;Recibe digito.
205.	2251	FE	0D			CP \$0D	;Checa si fin factor.
206.	2253	CB				RET Z	;Termina si es fin.
207.	2254	FE	2D			CP \$2D	;Checa si es negativo.
208.	2256	29	05			JR NZ,POST	;Salta si es positivo.
209.	2258	FD	77	00		LD (<IV+0),A	;Establece signo negativo.
210.	225B	18	E4			JR FACUNO	;Recibe siguiente digito.
211.	225D	FE	2B		POST	CP \$2B	;Checa si es positivo.
212.	225F	20	06			JR NZ,FAC	;Salta si no es positivo.
213.	2261	FD	36	00	20	LD (<IV+0),#20	;Establece signo positivo.
214.	2265	18	DA			JR FACUNO	;Recibe siguiente digito.
215.	2267	FE	30		FAC	CP \$30	;Establece filtro decimal
216.	2269	FA	4E	22		JP M,FACDOS	;para la recepcion de los
217.	226C	FE	3A			CP \$3A	digitos.
218.	226E	F2	4E	22		JP P,FACDOS	;
219.	2271	21	5C	11		LD HL,#115C	;Prepara espacio para nue-
220.	2274	11	5B	11		LD DE,#115B	vo digito.
221.	2277	01	03	00		LD BC,#0003	;
222.	227A	ED	B0			LDIR	;
223.	227C	2B				DEC HL	;
224.	227D	77				LD (<HL),A	;Almacena nuevo digito.
225.	227E	C3	41	22		JP FACUNO	;Prosigue captura.
226.	2281	3A	39	11	DEFOP	LD A,<OP>	;Toma codigo identifica-
227.	2284						cion de operacion.
228.	2284	CB	47			BIT 0,A	;Checa si amplificacion.
229.	2286	28	0D			JR Z,DEFUNO	;Salta si no lo es.
230.	2288	E5				PUSH HL	;Copia dato.
231.	2289	D1				POP DE	;
232.	228A	21	00	00		LD HL,#0000	;Prepara reg. producto.
233.	228D	ED	4B	2F	11	LD BC,<DIVOR>	;Toma factor.
234.	2291	CD	F8	1B		CALL CONV	;Realiza transformacion.
235.	2294	C9				RET	;Termina.
236.	2295	CB	4F		DEFUNO	BIT 1,A	;Checa si reduccion.
237.	2297	28	04			JR Z,DEFDOS	;Salta si lo es.
238.	2299	CD	F7	22		CALL DESPLA	;Realiza translacion.
239.	229C	C9				RET	;Termina.
240.	229D	CD	5B	1C	DEFDOS	CALL DIV	;Realiza reduccion.
241.	22A0	C9				RET	;Termina.
242.	22A1	3A	39	11	DEFORI	LD A,<OP>	;Toma codigo de identifica-
243.	22A4						cion de operacion.
244.	22A4	FE	02			CP \$02	;Checa si es translacion.
245.	22A6	28	01			JR Z,SIORIG	;Salta si es translacion.
246.	22A8	C9				RET	;Termina.
247.	22A9	01	05	00	SIORIG	LD BC,#0005	;Prepara factor escala-
248.	22AC	ED	43	2F	11	LD (<DIVOR>,BC	miento.
249.	22B0	2A	31	11		LD HL,<FACTX>	;Toma factor eje X.
250.	22B3	E5				PUSH HL	;Preserva factor.
251.	22B4	CD	5B	1C		CALL DIV	;Realiza escalamiento.
252.	22B7	D1				POP DE	;Recupera signo factor.
253.	22B8	CD	C1	1C		CALL 1149	;Realiza compensacion.
254.	22BB	EB				EX DE,HL	;Preserva factor escalado.

APEND A

Sistema Graficador FAP-100

255.	22BC	2A	27	11		LD HL,(ORGUSX)	;Toma coordenada X origen
256.	22BF						;usuario.
257.	22BF	19				ADD HL,DE	;Actualiza coordenada.
258.	22C0	22	27	11		LD (ORGUSX),HL	;Almacena coordenada.
259.	22C3	2A	33	11		LD HL,(FACTY)	;Toma factor eje Y.
260.	22C6	E5				PUSH HL	;Preserva factor.
261.	22C7	CD	5B	1C		CALL DIV	;Realiza escalamiento.
262.	22CA	D1				POP DE	;Retoma signo factor.
263.	22CB	CD	C1	1C		CALL 1148	;Realiza compensacion.
264.	22CE	EB				EX DE,HL	;Preserva factor escalado.
265.	22CF	2A	29	11		LD HL,(ORGUSY)	;Toma coordenada Y origen
266.	22D2						;usuario.
267.	22D2	19				ADD HL,DE	;Actualiza coordenada.
268.	22D3	22	29	11		LD (ORGUSY),HL	;Almacena coordenada.
269.	22D6	C9				RET	;Termina.
270.	22D7	3A	39	11	DEFOPM	LD A,(OP)	;Toma codigo identifica-
271.	22DA						cion de operacion.
272.	22DA	CB	47			BIT 0,A	;Checa si amplificacion.
273.	22DC	28	0D			JR Z,DEF1M	;Salta si no es.
274.	22DE	E5				PUSH HL	;Copia dato.
275.	22DF	D1				POP DE	
276.	22E0	21	00	00		LD HL,#0000	;Prepara reg. producto.
277.	22E3	ED	4B	2F	11	LD BC,(DIVOR)	;Prepara factor.
278.	22E7	CD	F8	1B		CALL CONV	;Realiza amplificacion.
279.	22EA	C9				RET	;Termina.
280.	22EB	CB	4F		DEF1M	BIT 1,A	;Checa si desplazamiento.
281.	22ED	28	04			JR Z,DEF2M	;Salta si no lo es.
282.	22EF	CD	FD	22		CALL DESPM	;Realiza translacion.
283.	22F2	C9				RET.	;Termina.
284.	22F3	CD	5B	1C	DEF2M	CALL DIV	;Realiza reduccion.
285.	22F6	C9				RET	;Termina.
286.	22F7	ED	5B	2F	11	LD DE,(DIVOR)	;Toma factor.
287.	22FB	19			DESPLA	ADD HL,DE	;Realiza desplazamiento.
288.	22FC	C9				RET	;Termina.
289.	22FD	E5			DESPM	PUSH HL	;Preserva dato origen.
290.	22FE	2A	2F	11		LD HL,(DIVOR)	;Toma factor.
291.	2301	E5				PUSH HL	;Preserva factor.
292.	2302	01	05	00		LD BC,#0005	;Prepara factor escala-
293.	2305	ED	43	2F	11	LD (DIVOR),BC	miento.
294.	2309	CD	5B	1C		CALL DIV	;Escala factor.
295.	230C	D1				POP DE	;Retoma signo factor.
296.	230D	CD	C1	1C		CALL 1148	;Realiza compensacion.
297.	2310	EB				EX DE,HL	;Preserva factor escalado.
298.	2311	E1				POP HL	;Toma dato.
299.	2312	19				ADD HL,DE	;Realiza transformacion.
300.	2313	C9				RET	;Termina.
300.	2314						

T a b l a d e S i m b o l o s

Nombre	Valor	Linea de Definicion
let1	1145	6
let2	115A	7
let7	11B0	8
let9	11CA	9
let12	11E9	10
let13	11F5	11



let14	1201	12
letec	0136	13
espa	013F	14
esmen	01DC	15
esmen1	18BE	16
guadas	1BA6	17
llena	19C4	18
indir	190D	19
guadir	1A09	20
nuedat	1CD7	21
nuecon	1CE4	22
sumiy5	1CED	23
conv	1BF8	24
div	1C5B	25
1148	1CC1	26
posc	C479	27
incr	C401	28
cr1f	C53A	29
orgusx	1127	30
orgusy	1129	31
orgx	1135	32
orgy	1137	33
factx	1131	34
facty	1133	35
dirl	110B	36
condir	1124	37
col	1126	38
divor	112F	39
op	1139	40
renpos	1144	41
rduc	20C7	44
amp	20D0	48
des	20D9	52
escala	210F	76
scuno	2112	77
scdos	2130	94
reqpar	2134	96
escusu	2186	126
usdos	21AA	138
escmaq	21D9	156
simaq	21EA	161
mafin	222C	190
nocr	2232	192
infac	223A	197
facuno	2241	199
facdos	224E	204
post	225D	211
fac	2267	215
defop	2281	226
defuno	2295	236
defdos	229D	240
defori	22A1	242
siorig	22A9	247
defopm	22D7	270
def1m	22EB	280
def2m	22F3	284
despla	22F7	286
despm	22FD	289

APEND A

Sistema Graficador FAP-100

```
1. 0000 ;
2. 0000 ;***
3. 0000 ;
4. 0000 ;
5. 2314
6. 2314 LET1 *= $2314
7. 2314 LET7 = $1145
8. 2314 LET9 = $1180
9. 2314 LET14 = $11CA
10. 2314 LET32 = $138D
11. 2314 LETEC = $0136
12. 2314 ESPA = $013F
13. 2314 ESMEM = $01DC
14. 2314 ESMEM1 = $188E
15. 2314 LLENA = $19C4
16. 2314 INDIR = $19DD
17. 2314 GUADIR = $1A09
18. 2314 GUADAS = $1BA6
19. 2314 COMP2 = $1B55
20. 2314 CONY = $1BF8
21. 2314 DIV = $1B5B
22. 2314 1148 = $1CC1
23. 2314 NUEDAT = $1CD7
24. 2314 NUECON = $1CE4
25. 2314 SUMIY5 = $1CED
26. 2314 DIV16 = $1C70
27. 2314 INFAC = $223A
28. 2314 INCR = $C401
29. 2314 CRLF = $C53A
30. 2314 DIR1 = $110B
31. 2314 ABSX = $1117
32. 2314 ABSY = $1119
33. 2314 DIFX = $1113
34. 2314 DIFY = $1115
35. 2314 DIVOR = $112F
36. 2314 SEN = $113C
37. 2314 COS = $113E
38. 2314 ORGUSX = $1127
39. 2314 ORGUSY = $1129
40. 2314 RAD = $1122
41. 2314 ORGX = $1135
42. 2314 ORGY = $1137
43. 2314 CONDIR = $1124
44. 2314 GRAD = $113A
45. 2314 COL = $1126
46. 2314 RENFOS = $1144
47. 2314 ;
48. 2314 ;
49. 2314 CD 36 23 ROT CALL REQANG ;Introduce angulo y calcula
50. 2317 ; ; la seno y coseno.
```

APEHD A

## Sistema Graficador FAP-100

51.	2317	CD	36	01	ROTUNO	CALL LETEC	;Recibe contestacion sobre
52.	231A						;modificar tabla usuario.
53.	231A	FE	4E			CP #4E	;Checa si solo tabla maq.
54.	231C	28	14			JR Z,SOMA	;Salta si solo tabla maq.
55.	231E	FE	53			CP #53	;Checa si tambien la de
56.	2320						;usuario.
57.	2320	20	F5			JR NZ,ROTUNO	;Salta si no es respuesta.
58.	2322	CD	6D	23		CALL ROTUSU	;Rota tabla usuario.
59.	2325	FD	2A	0B 11		LD IY,<DIR1>	;Apunta inicio de tabla.
60.	2329	FD	6E	FA		LD L,<IY-6>	;Toma direccion final de
61.	232C	FD	66	FB		LD H,<IY-5>	;tabla usuario.
62.	232F	22	0B	11		LD <DIR1>,HL	;Actualiza inicio tabla.
63.	2332	CD	AA	23	SOMA	CALL ROTMAQ	;Rota tabla de maquina.
64.	2335	C9				RET	;Termina.
65.	2336	3E	1A		REQANG	LD A,#1A	;Manda comando para borrar
66.	2338	CD	3F	01		CALL ESPA	;pantalla.
67.	233B	3E	20			LD A,#20	;Prepara no. de renglon de
68.	233D	32	44	11		LD <RENPOS>,A	;pantalla.
69.	2340	3E	0A			LD A,#0A	;Prepara no. columna de
70.	2342	32	26	11		LD <COL>,A	;pantalla.
71.	2345	21	8D	13		LD HL,LET32	;Despliega mensaje "ANGU-
72.	2348	CD	BE	18		CALL ESMEN1	;LO? "
73.	234B	CD	3A	22		CALL INFAC	;Recibe angulo rotacion.
74.	234E	21	3A	11		LD HL,GRAD	;Apunta a espacio reserva-
75.	2351	22	24	11		LD <CONDIR>,HL	;do para angulo rotacion.
76.	2354	CD	A6	1B		CALL GUADAS	;Almacena angulo.
77.	2357	CD	3A	C5		CALL CRLF	;Salta renglon.
78.	235A	CD	01	C4		CALL INCR	;Incrementa apuntador de
79.	235D						;renglon.
80.	235D	CD	1F	24		CALL SENCOS	;Calcula seno y coseno.
81.	2360	CD	3A	C5		CALL CRLF	;Salta renglon.
82.	2363	CD	01	C4		CALL INCR	;Incrementa apuntador de
83.	2366						;renglon.
84.	2366	21	01	12		LD HL,LET14	;Despliega mensaje "MODI-
85.	2369	CD	BE	18		CALL ESMEN1	;FICA TABLA USUARIO?
86.	236C						;(<S/N> "
87.	236C	C9				RET	;Termina.
88.	236D	3E	1A		ROTUSU	LD A,#1A	;Manda comando para borrar
89.	236F	CD	3F	01		CALL ESPA	;pantalla.
90.	2372	21	45	11		LD HL,LET1	;Despliega mensaje "DIREC-
91.	2375	CD	BE	18		CALL ESMEN1	;CION INICIAL: "
92.	2378	CD	C4	19		CALL LLENA	;Prepara espacio para dato.
93.	237B	CD	DD	19		CALL INDIR	;Recibe dir. inicial.
94.	237E	CD	09	1A		CALL GUADIR	;Almacena dir. inicial.
95.	2381	CD	3A	C5		CALL CRLF	;Salta renglon.
96.	2384	21	B0	11		LD HL,LET7	;Despliega mensaje
97.	2387	CD	DC	01		CALL ESMEN	; "TRANS. "
98.	238A	FD	2A	0B 11		LD IY,<DIR1>	;Apunta inicio tabla.
99.	238E	FD	CB	05 4E	SIROTA	BIT 1,<IY+5>	;Checa si es linea.
100.	2392	28	08			JR Z,LINEA	;Salta si es linea.
101.	2394	CD	76	24		CALL DIVORG	;Escala origen usuario.

APEND A

Sistema Graficador FAP-100

102.	2397	CD	AC	24		CALL ROTCIR	;Rota circulo.
103.	239A	18	03			JR REPITE	;Prosigue transformacion.
104.	239C	CD	17	25	LINEA	CALL OBSECO	;Rota linea.
105.	239F	CD	E4	1C	REPITE	CALL NUECON	;Copia palabra control.
106.	23A2	CB	7F			BIT 7,A	;Checa si fin de tabla.
107.	23A4	C8				RET Z	;Termina.
108.	23A5	CD	ED	1C		CALL SUMIYS	;Accesa siguientes datos.
109.	23A8	18	E4			JR SIROTA	;Continua transformacion.
110.	23AA	CD	3A	C5	ROTHAQ	CALL CRLF	;Salta renglon.
111.	23AD	21	80	11		LD HL,LET7	;Despliega mensaje
112.	23B0	CD	DC	01		CALL ESMEN	; "TRANS. "
113.	23B3	FD	2A	08	11	LD IY,<DIR1>	;Apunta inicio de tabla.
114.	23B7	FD	22	24	11	LD <CONDIR>,IY	;Actualiza apuntador.
115.	23BB	CD	47	25	SIROMA	CALL XY1ORG	;Translada dato a origen
116.	23BE						;absoluto.
117.	23BE	FD	CB	04	7E	BIT 7,<IY+4>	;Checa si es fin tabla.
118.	23C2	20	38			JR NZ,LINMAQ	;Termina con linea.
119.	23C4	FD	CB	04	4E	BIT 1,<IY+4>	;Checa si es circulo.
120.	23C8	28	32			JR Z,LINMAQ	;Salta si no es.
121.	23CA	CD	ED	1C		CALL SUMIYS	;Accesa centro circulo.
122.	23CD	CD	47	25		CALL XY1ORG	;Translada a origen abso-
123.	23D0						;luto.
124.	23D0	CD	91	25		CALL RESIYS	;Accesa punto inicial de
125.	23D3						;circulo.
126.	23D3	FD	6E	05		LD L,<IY+5>	;Toma coordenada X centro
127.	23D6	FD	66	06		LD H,<IY+6>	
128.	23D9	22	17	11		LD <ABSX>,HL	;Almacena coordenada.
129.	23DC	FD	6E	07		LD L,<IY+7>	;Toma coordenada Y centro.
130.	23DF	FD	66	08		LD H,<IY+8>	
131.	23E2	22	19	11		LD <ABSY>,HL	;Almacena coordenada.
132.	23E5	21	00	00		LD HL,\$0000	;Prepara valores de com-
133.	23E8	22	13	11		LD <DIFX>,HL	;paracion.
134.	23EB	22	15	11		LD <DIFY>,HL	
135.	23EE	CD	AC	24		CALL ROTCIR	;Rota circulo.
136.	23F1	CD	91	25		CALL RESIYS	;Apunta punto inicial
137.	23F4						;circulo.
138.	23F4	CD	6E	25		CALL XY1ORG	;Translada a origen re-
139.	23F7						;lativo inicial.
140.	23F7	CD	ED	1C		CALL SUMIYS	;Apunta centro circulo.
141.	23FA	18	03			JR RESORI	;Salta a trasladar
142.	23FC						;centro.
143.	23FC	CD	17	25	LINMAQ	CALL OBSECO	;Rota linea.
144.	23FF	CD	6E	25	RESORI	CALL XY1ORG	;Translada a origen rela-
145.	2402						;tivo inicial.
146.	2402	2A	24	11		LD HL,<CONDIR>	;Toma apuntador.
147.	2405	CD	E4	1C		CALL NUECON	;Copia control.
148.	2408	CB	7F			BIT 7,A	;Checa si fin de tabla.
149.	240A	20	05			JR NZ,ROTDOS	;Salta si es fin.
150.	240C	CD	ED	1C		CALL SUMIYS	;Accesa siguiente pareja
151.	240F						;de puntos.
152.	240F	18	AA			JR SIROMA	;Continua transformacion.

APEND A

Sistema Graficador FAP-100

153.	2411	21	CA	11	ROTDOS	LD HL, LET9	;Despliega mensaje
154.	2414	CD	DC	01		CALL ESMEN	; "LISTO &".
155.	2417	CD	36	01	ROTTRE	CALL LETEC	;Busca comando retorno.
156.	241A	FE	0D			CP #0D	;Checa si es comando.
157.	241C	F0	F9			JR NZ,ROTTRE	;Salta si no es.
158.	241E	C9				RET	;Termina.
159.	241F	01	64	00	SENCOS	LD BC, #0064	;Prepara factor de escala.
160.	2422	ED	43	2F 11		LD (DIVOR), BC	;
161.	2425	ED	5B	3A 11		LD DE, (GRAD)	;Toma no. de grados.
162.	242A	01	38	00		LD BC, #0038	;Prepara factor de escala.
163.	242D	CD	9C	25		CALL RECUR	;Obtiene equivalencia en
164.	2430						;intervalo 0-200.
165.	2430	E5				PUSH HL	;Preserva equivalencia.
166.	2431	21	00	00		LD HL, #0000	;Inicializa sen 0=0 y
167.	2434	E5				PUSH HL	;cos 0 = 1000.
168.	2435	22	3C	11		LD (SEN), HL	;
169.	2438	21	E8	03		LD HL, #03E8	;
170.	243B	22	3E	11		LD (COS), HL	;
171.	243E	ED	5B	3C 11	SIREC	LD DE, (SEN)	;Obtiene:
172.	2442	01	03	00		LD BC, #0003	;
173.	2445	CD	9C	25		CALL RECUR	;cos' = cos - C * sen
174.	2448	EB				EX DE, HL	;
175.	2449	2A	3E	11		LD HL, (COS)	;
176.	244C	A7				AND A	;
177.	244D	ED	52			SBC HL, DE	;
178.	244F	E5				PUSH HL	;Preserva cos'.
179.	2450	ED	5B	3E 11		LD DE, (COS)	;Obtiene:
180.	2454	01	03	00		LD BC, #0003	;
181.	2457	CD	9C	25		CALL RECUR	;sen' = sen + C * cos
182.	245A	EB				EX DE, HL	;
183.	245B	2A	3C	11		LD HL, (SEN)	;
184.	245E	19				ADD HL, DE	;
185.	245F	22	3C	11		LD (SEN), HL	;Almacena sen'
186.	2462	E1				POP HL	;Retoma cos'.
187.	2463	22	3E	11		LD (COS), HL	;Almacena cos'.
188.	2466	D1				POP DE	;Retoma contador.
189.	2467	13				INC DE	;Incrementa contador.
190.	2468	E1				POP HL	;Retoma equivalencia.
191.	2469	E5				PUSH HL	;Preserva equivalencia
192.	246A	D5				PUSH DE	;y contador.
193.	246B	A7				AND A	;Checa si es angulo de-
194.	246C	ED	52			SBC HL, DE	;seado.
195.	246E	28	03			JR Z, SCFIN	;Salta si lo es.
196.	2470	F2	3E	24		JP P, SIREC	;Prosigue calculo.
197.	2473	E1			SCFIN	POP HL	;Retoma equivalencia y
198.	2474	E1				POP HL	;
199.	2475	C9				RET	;Termina.
200.	2476	01	05	00	DIVORC	LD BC, #0005	;Prepara factor de escala-
201.	2479	ED	43	2F 11		LD (DIVOR), BC	;miento.
202.	247D	FD	6E	05		LD L, (Y+5)	;Toma coordenada X
203.	2480	FD	6E	06		LD H, (Y+6)	;centro.

APEND A

Sistema Graficador FAP-100

204.	2483	E5			PUSH HL		;Preserva signo.
205.	2484	CD	5B	1B	CALL DIV		;Escala coordenada.
206.	2487	D1			POP DE		;Retoma signo.
207.	2488	CD	C1	1C	CALL 1148		;Realiza compensacion.
208.	248B	22	17	11	LD <ABSX>,HL		;Almacena coordenada.
209.	248E	FD	6E	07	LD L,<IY+7>		;Toma coordenada Y
210.	2491	FD	66	08	LD H,<IY+8>		;centro.
211.	2494	E5			PUSH HL		;Preserva signo.
212.	2495	CD	5B	1B	CALL DIV		;Escala coordenada.
213.	2498	D1			POP DE		;Retoma signo.
214.	2499	CD	C1	1C	CALL 1148		;Realiza compensacion.
215.	249C	22	19	11	LD <ABSX>,HL		;Almacena coordenada.
216.	249F	2A	27	11	LD HL,<ORGUSX>		;Toma coordenada X
217.	24A2						;origen usuario.
218.	24A2	22	13	11	LD <DIFX>,HL		;Almacena coordenada.
219.	24A5	2A	29	11	LD HL,<ORGUSY>		;Toma coordenada Y
220.	24A8						;origen usuario.
221.	24A8	22	15	11	LD <DIFY>,HL		;Almacena coordenada.
222.	24AB	C9			RET		;Termina.
223.	24AC	2A	17	11	LD HL,<ABSX>	ROTCIR	;Toma coordenada X centro.
224.	24AF	ED	4B	13 11	LD BC,<DIFX>		;Toma coordenada X origen.
225.	24B3	A7			AND A		;Checa si coordenada X
226.	24B4	ED	42		SBC HL,BC		;coincide con origen.
227.	24B6	20	1A		JR NZ,NOORG		;Salta si no coincide.
228.	24B8	2A	19	11	LD HL,<ABSX>		;Toma coordenada Y centro.
229.	24BB	ED	4B	15 11	LD BC,<DIFY>		;Toma coordenada Y origen.
230.	24BF	A7			AND A		;Checa si coordenada Y
231.	24C0	ED	42		SBC HL,BC		;coincide con origen.
232.	24C2	20	0E		JR NZ,NOORG		;Salta si no coincide.
233.	24C4	CD	ED	1C	CALL SUMIY5		;Accesa siguiente pareja
234.	24C7	2A	24	11	LD HL,<CONDIR>		;de puntos y actualiza
235.	24CA	11	09	00	LD DE,#0009		;apuntador de tabla,
236.	24CD	19			ADD HL,DE		;
237.	24CE	22	24	11	LD <CONDIR>,HL		;
238.	24D1	C9			RET		;Termina.
239.	24D2	FD	6E	02	LD L,<IY+2>	NOORG	;Toma coordenada Y punto
240.	24D5	FD	66	03	LD H,<IY+3>		;inicial.
241.	24D8	FD	5E	07	LD E,<IY+7>		;Toma coordenada Y centro.
242.	24DB	FD	56	08	LD D,<IY+8>		;
243.	24DE	A7			AND A		;Calcula radio.
244.	24DF	ED	52		SBC HL,DE		;
245.	24E1	22	22	11	LD <RAD>,HL		;Almacena radio.
246.	24E4	CD	17	25	CALL OBSECO		;Transforma punto inicial.
247.	24E7	CD	E4	1C	CALL NUECON		;Copia palabra control.
248.	24EA	FD	6E	00	LD L,<IY+0>		;Toma coordenada X punto
249.	24ED	FD	75	05	LD <IY+5>,L		;inicial y lo copia en
250.	24F0	FD	6E	01	LD L,<IY+1>		;coordenada X centro.
251.	24F3	FD	75	06	LD <IY+6>,L		;
252.	24F6	FD	6E	02	LD L,<IY+2>		;Toma coordenada Y punto
253.	24F9	FD	66	03	LD H,<IY+3>		;inicial.
254.	24FC	ED	5B	22 11	LD DE,<RAD>		;Toma radio.

APER0 A

Sistema Graficador FAP-100

255.	2500	A7		AND A	;Calcula coordenada Y
256.	2501	ED	52	SBC HL,DE	;centro.
257.	2503	FD	75 07	LD (IY+7),L	;Almacena coordenada.
258.	2506	FD	74 08	LD (IY+8),H	;
259.	2509	2A	24 11	LD HL,(CONDIR)	;Toma apuntador.
260.	250C	11	04 00	LD DE,#0004	;Actualiza apuntador.
261.	250F	19		ADD HL,DE	;
262.	2510	22	24 11	LD (CONDIR),HL	;
263.	2513	CD	ED 1C	CALL SUMIYS	;
264.	2516	C9		RET	;Termina.
265.	2517	ED	5B 3C 11	LD DE,(SEN)	;Almacena sen.
266.	2518	CD	C4 25	CALL YFUN	;Realiza y*sen(ang)/1000.
267.	251E	E5		PUSH HL	;Preserva resultado.
268.	251F	ED	5B 3E 11	LD DE,(COS)	;Toma coseno.
269.	2523	CD	BA 25	CALL XFUN	;Obtiene x*cos(ang)/1000.
270.	2526	D1		POP DE	;Retoma resultado previo.
271.	2527	A7		AND A	;Realiza x*cos - y*sen.
272.	2528	ED	52	SBC HL,DE	;
273.	252A	E5		PUSH HL	;Preserva resultado.
274.	252B	ED	5B 3E 11	LD DE,(COS)	;Toma coseno.
275.	252F	CD	C4 25	CALL YFUN	;Obtiene y*cos(ang)/1000.
276.	2532	E5		PUSH HL	;Preserva resultado.
277.	2533	ED	5B 3C 11	LD DE,(SEN)	;Toma seno.
278.	2537	CD	BA 25	CALL XFUN	;Obtiene x*cos(ang)/1000.
279.	253A	D1		POP DE	;Retoma resultado previo.
280.	253B	19		ADD HL,DE	;Realiza y*cos + x*sen.
281.	253C	D1		POP DE	;Retoma x'.
282.	253D	EB		EX DE,HL	;Intercambia x', y'.
283.	253E	D5		PUSH DE	;Preserva y'.
284.	253F	CD	D7 1C	CALL NUEDAT	;Almacena x'.
285.	2542	E1		POP HL	;Retoma y almacena y'.
286.	2543	CD	D7 1C	CALL NUEDAT	;
287.	2546	C9		RET	;Termina.
288.	2547	ED	5B 35 11	LD DE,(ORGX)	;Toma coordenada X origen.
289.	2548	FD	6E 00	LD L,(IY+0)	;Toma coordenada X punto
290.	254E	FD	66 01	LD H,(IY+1)	;inicial.
291.	2551	A7		AND A	;Desplaza hacia origen.
292.	2552	ED	52	SBC HL,DE	;
293.	2554	FD	75 00	LD (IY+0),L	;Almacena coordenada
294.	2557	FD	74 01	LD (IY+1),H	;trasladada.
295.	255A	ED	5B 37 11	LD DE,(ORGY)	;Toma coordenada Y origen.
296.	255E	FD	6E 02	LD L,(IY+2)	;Toma coordenada Y punto
297.	2561	FD	66 03	LD H,(IY+3)	;inicial.
298.	2564	A7		AND A	;Desplaza hacia origen.
299.	2565	ED	52	SBC HL,DE	;
300.	2567	FD	75 02	LD (IY+2),L	;Almacena coordenada
301.	256A	FD	74 03	LD (IY+3),H	;trasladada.
302.	256D	C9		RET	;Termina.
303.	256E	ED	5B 35 11	LD DE,(ORGX)	;Toma coordenada X origen.
304.	2572	FD	6E 00	LD L,(IY+0)	;Toma coordenada X punto
305.	2575	FD	66 01	LD H,(IY+1)	;inicial.

APEND A

Sistema Graficador FAP-100

306.	2578	19		ADD HL,DE	;Translada hacia origen
307.	2579				;relativo inicial.
308.	2579	FD	75 00	LD (IY+0),L	;Almacena coordenada
309.	257C	FD	74 01	LD (IY+1),H	;transladada.
310.	257F	ED	5B 37	LD DE,(ORGY)	;Toma coordenada Y origen.
311.	2583	FD	6E 02	LD L,(IY+2)	;Toma coordenada Y punto
312.	2586	FD	66 03	LD H,(IY+3)	;inicial.
313.	2589	19		ADD HL,DE	;Translada hacia origen
314.	258A				;relativo inicial.
315.	258A	FD	75 02	LD (IY+2),L	;Almacena coordenada
316.	258D	FD	74 03	LD (IY+3),H	;transladada.
317.	2590	C9		RET	;Termina.
318.	2591	FD	2B	DEC IY	;Realiza
319.	2593	FD	2B	DEC IY	;IY = IY - 5.
320.	2595	FD	2B	DEC IY	;
321.	2597	FD	2B	DEC IY	;
322.	2599	FD	2B	DEC IY	;
323.	2598	C9		RET	;Termina.
324.	259C	D5		PUSH DE	;Preserva signo de dato.
325.	259D	CB	7A	BIT 7,D	;Checa signo.
326.	259F	28	07	JR Z,RECUR1	;Salta si es positivo.
327.	25A1	D5		PUSH DE	;Copia dato.
328.	25A2	E1		POP HL	;
329.	25A3	CD	55 1B	CALL COMP2	;Obtiene complemento a 2.
330.	25A6	E5		PUSH HL	;Copia dato.
331.	25A7	D1		POP DE	;
332.	25A8	21	00 00	LD HL,\$0000	;Prepara reg. producto.
333.	25AB	CD	F8 1B	CALL CONV	;Realiza multiplicacion.
334.	25AE	D1		POP DE	;Retoma signo.
335.	25AF	CB	7A	BIT 7,D	;Checa signo.
336.	25B1	28	03	JR Z,RECUR2	;Salta si es positivo.
337.	25B3	CD	55 1B	CALL COMP2	;Obtiene complemento a 2.
338.	25B6	CD	5B 18	CALL DIV	;Realiza division.
339.	25B9	C9		RET	;Termina.
340.	25BA	FD	4E 00	LD C,(IY+0)	;Toma coordenada X.
341.	25BD	FD	46 01	LD B,(IY+1)	;
342.	25C0	CD	CE 25	CALL OPERA	;Obtiene x*sen o cos/1000.
343.	25C3	C9		RET	;Termina.
344.	25C4	FD	4E 02	LD C,(IY+2)	;Toma coordenada Y.
345.	25C7	FD	46 03	LD B,(IY+3)	;
346.	25CA	CD	CE 25	CALL OPERA	;Obtiene y*sen o cos/1000.
347.	25CD	C9		RET	;Termina.
348.	25CE	CD	DE 25	CALL MULT16	;Realiza* x o y*sen o cos.
349.	25D1	CD	41 26	CALL PREDIV	;Divide resultado entre
350.	25D4	01	E8 03	LD BC,\$03E8	;1000.
351.	25D7	ED	43 2F	LD (DIVOR),BC	;
352.	25DB	C3	70 1C	JP DIV16	;
353.	25DE	C5		PUSH BC	;Preserva dato.
354.	25DF	D5		PUSH DE	;
355.	25E0	3E	10	LD A,\$10	;Prepara contador.
356.	25E2	08		EX AF,AF'	;Preserva contador.



APEND A

Sistema Graficador FAP-100

357.	25E3	CB	78		BIT 7,B		;Checa signo de dato.
358.	25E5	28	07		JR Z,MULT01		;Salta si es positivo.
359.	25E7	CS			PUSH BC		;Copia dato.
360.	25E8	E1			POP HL		;Copia dato.
361.	25E9	CD	55 1B		CALL COMP2		;Obtiene complemento a 2.
362.	25EC	ES			PUSH HL		;Copia dato.
363.	25ED	C1			POP BC		;Copia dato.
364.	25EE	D1		MULT01	POP DE		;Retoma multiplicador.
365.	25EF	7A			LD A,D		;Toma signo.
366.	25F0	F5			PUSH AF		;Preserva signo y estatus.
367.	25F1	CB	7F		BIT 7,A		;Checa signo.
368.	25F3	28	07		JR Z,MULT02		;Salta si es positivo.
369.	25F5	D5			PUSH DE		;Copia multiplicador.
370.	25F6	E1			POP HL		;Copia multiplicador.
371.	25F7	CD	55 1B		CALL COMP2		;Obtiene complemento a 2.
372.	25FA	ES			PUSH HL		;Copia resultado.
373.	25FB	D1			POP DE		;Copia resultado.
374.	25FC	F1		MULT02	POP AF		;Retoma dato y estatus.
375.	25FD	08			EX AF,AF'		;Retoma contador.
376.	25FE	21	00 00		LD HL,\$0000		;Inicializa producto parcial.
377.	2601						;Inicializa producto parcial.
378.	2601	CB	43		BIT 0,E		;Checa primer bit multiplicador.
379.	2603	28	01		JR Z,CORRE		;Salta si es cero.
380.	2603	28	01		ADD HL,BC		;Adiciona con multiplicando.
381.	2605	09					;Adiciona con multiplicando.
382.	2606						;Adiciona con multiplicando.
383.	2606	CB	38		SRL E		;Realiza corrimiento de registro de 2n bits, que almacena al producto parcial y al multiplicador.
384.	2608	CB	42		BIT 0,D		;Realiza corrimiento de registro de 2n bits, que almacena al producto parcial y al multiplicador.
385.	260A	28	02		JR Z,CORRED		;Salta si no es fin.
386.	260C	CB	FB		SET 7,E		;Almacena al producto parcial y al multiplicador.
387.	260E	CB	3A		SRL D		;Almacena al producto parcial y al multiplicador.
388.	2610	CB	45		BIT 0,L		;Almacena al producto parcial y al multiplicador.
389.	2612	28	02		JR Z,CORREL		;Almacena al producto parcial y al multiplicador.
390.	2614	CB	FA		SET 7,D		;Almacena al producto parcial y al multiplicador.
391.	2616	CB	3D		SRL L		;Almacena al producto parcial y al multiplicador.
392.	2618	CB	44		BIT 0,H		;Almacena al producto parcial y al multiplicador.
393.	261A	28	02		JR Z,CORREH		;Almacena al producto parcial y al multiplicador.
394.	261C	CB	FD		SET 7,L		;Almacena al producto parcial y al multiplicador.
395.	261E	CB	3C		SRL H		;Almacena al producto parcial y al multiplicador.
396.	2620	3D			SRL H		;Almacena al producto parcial y al multiplicador.
397.	2621	FE	00		DEC A		;Decrementa contador.
398.	2623	20	DC		CP #00		;Checa si fin operacion.
399.	2625	08			JR NZ,BITDOR		;Salta si no es fin.
400.	2626	C1			EX AF,AF'		;Retoma signo y estatus.
401.	2627	A8			POP BC		;Retoma signo dato.
402.	2628				XOR B		;Checa si signos iguales o diferentes.
403.	2628	CB	7F				;Checa si signos iguales o diferentes.
404.	262A	CB			BIT 7,A		;Checa si signos iguales o diferentes.
405.	262B	ES			RET Z		;Termina si iguales.
406.	262C			COMP16	PUSH HL		;Guarda parte alta producto.
407.	262C	EB					;Guarda parte alta producto.
					EX DE,HL		;Toma parte baja.

APEND A

Sistema Graficador FAP-100

408.	262D	CD 55 1B	CALL COMP2	;Obtiene complemento a
409.	2630			;2 de parte baja.
410.	2630	EB	EX DE,HL	;Almacena resultado.
411.	2631	E1	POP HL	;Retoma parte alta.
412.	2632	30 06	JR NC,MULT03	;Salta si solo se requiere
413.	2634			;comp. a 1 de parte al-
414.	2634			;ta.
415.	2634	D5	PUSH DE	;Preserva parte baja.
416.	2635	CD 55 1B	CALL COMP2	;Obtiene complemento a 2
417.	2638			;de parte alta.
418.	2638	D1	POP DE	;Retoma parte baja.
419.	2639	C9	RET	;Termina.
420.	263A	7D	MULT03 LD A,L	;Toma byte alto.
421.	263B	2F	CPL	;Lo invierte.
422.	263C	6F	LD L,A	;Lo almacena.
423.	263D	7C	LD A,H	;Toma byte bajo.
424.	263E	2F	CPL	;Lo invierte.
425.	263F	67	LD H,A	;Lo almacena.
426.	2640	C9	RET	;Termina.
427.	2641	E5	PREDIV PUSH HL	;Preserva parte alta.
428.	2642	CB 7C	BIT 7,H	;Checa signo.
429.	2644	28 03	JR Z,PRED01	;Salta si es positivo.
430.	2646	CD 2B 26	CALL COMP16	;Obtiene complemento a 2.
431.	2649	7A	PRED01 LD A,D	;Estructura los dos re-
432.	264A	CB 3F	SRL A	;gistros de 16 bits, pero
433.	264C	CB 3F	SRL A	;utilizando solamente los
434.	264E	CB 3F	SRL A	;12 primeros bits de ca-
435.	2650	CB 3F	SRL A	;da uno de ellos, como
436.	2652	65	LD H,L	;lo requiere al algoritmo
437.	2653	CB 25	SLA L	;de division.
438.	2655	CB 25	SLA L	
439.	2657	CB 25	SLA L	
440.	2659	CB 25	SLA L	
441.	265B	85	ADD A,L	
442.	265C	6F	LD L,A	
443.	265D	CB 3C	SRL H	
444.	265F	CB 3C	SRL H	
445.	2661	CB 3C	SRL H	
446.	2663	CB 3C	SRL H	
447.	2665	C1	POP BC	;Preserva el signo para
448.	2666	F1	POP AF	;que no se pierda al re-
449.	2667	C5	PUSH BC	;greso de la subrutina.
450.	2668	F5	PUSH AF	
451.	2669	C9	RET	;Termina.
451.	266A			

T a b l a d e S i m b o l o s

Nombre	Valor	Linea de Definicion
let1	1145	6
let7	1180	7
let9	11CA	8
let14	1201	9
let32	138D	10
letec	0136	11
espa	013F	12
esmen	01DC	13

esmen1	188E	14
ilena	19C4	15
indir	19D0	16
guadir	1A09	17
guadas	1BA6	18
comp2	1B55	19
conv	1BF8	20
div	1B5B	21
l148	1CC1	22
nuedat	1CD7	23
nuecon	1CE4	24
sumiy5	1CED	25
div16	1C70	26
infac	223A	27
incr	C401	28
crlf	C53A	29
dir1	110B	30
absx	1117	31
absy	1119	32
difx	1113	33
dify	1115	34
divor	112F	35
sen	113C	36
cos	113E	37
orgusx	1127	38
orgusy	1129	39
rad	1122	40
orgx	1135	41
orgy	1137	42
condir	1124	43
grad	113A	44
col	1126	45
renpos	1144	46
rot	2314	49
rotuno	2317	51
soma	2332	63
reqang	2336	65
rotusu	236D	88
sirota	238E	99
linea	239C	104
repite	239F	105
rotmaq	23AA	110
siroma	23BB	115
linmaq	23FC	143
resori	23FF	144
rotdos	2411	153
roffre	2417	155
sencos	241F	159
sirec	243E	171
scfin	2473	197
divorg	2476	200
rotcir	24AC	223
noorg	24D2	239
obseco	2517	265
xylorg	2547	288
xylorg	256E	303
resiy5	2591	318
recur	259C	324
recur1	25A8	332
recur2	25B6	338

xfun	25BA	340
yfun	25C4	344
opera	25CE	348
mult16	25DE	353
mult01	25EE	364
mult02	25FC	374
bitdor	2601	378
corre	2606	383
corred	260E	387
correl	2616	391
correh	261E	395
comp16	262B	405
mult03	263A	420
prediv	2641	427
pre01	2649	431

APEND A

Sistema Graficador FAP-100

```

1. 0000
2. 0000
3. 0000
4. 0000
5. 266A
6. 266A LET23 = $266A
7. 266A LET24 = $12BB
8. 266A LET25 = $12E2
9. 266A LET26 = $12F5
10. 266A LET27 = $130A
11. 266A LET28 = $131E
12. 266A LET29 = $1335
13. 266A LET30 = $1348
14. 266A LET31 = $1362
15. 266A LETEC = $0136
16. 266A ESPA = $013F
17. 266A ESMEN = $01DC
18. 266A ESMEN1 = $18BE
19. 266A POSC = $C479
20. 266A GENTA = $18D7
21. 266A MENMOD = $1E38
22. 266A GRAF12 = $14F4
23. 266A ACCETA = $1CF3
24. 266A TRACIN = $1D8E
25. 266A RENPOS = $1144
26. 266A
27. 266A
28. 266A 3E 1A MENU LD A,$1A
29. 266C CD 3F 01 CALL ESPA ;Manda comando para borrar
30. 266F 3E 20 LD A,$20 ;pantalla.
31. 2671 32 44 11 LD <RENPOS>,A ;Prepara numero de renglon
32. 2674 16 18 LD D,$18 ;de pantalla.
33. 2676 CD 79 C4 CALL POSC ;Prepara no. columna.
34. 2679 21 8B 12 LD HL,LET23 ;Posiciona cursor.
35. 267C CD BE 18 CALL ESMEN1 ;Despliega mensaje "SISTE-
36. 267F MA GRAFICADOR 12 BITS
37. 267F 21 6C 13 LD HL,LET31 ;FAP-100".
38. 2682 CD BE 18 CALL ESMEN1 ;Despliega mensaje "G. R.
39. 2685 21 E2 12 LD HL,LET24 ;Espinosa P. anero/86".
40. 2688 CD BE 18 CALL ESMEN1 ;Despliega mensaje "MENU
41. 268B 21 F5 12 LD HL,LET25 ;PRINCIPAL ".
42. 268E CD BE 18 CALL ESMEN1 ;Despliega mensaje "0. Ge-
43. 2691 21 0A 13 LD HL,LET26 ;nerar tabla ".
44. 2694 CD BE 18 CALL ESMEN1 ;Despliega mensaje "1.
45. 2697 21 1E 13 LD HL,LET27 ;Cargar Tabla ".
46. 269A CD BE 18 CALL ESMEN1 ;Despliega mensaje "2. Mo-
47. 269D 21 35 13 LD HL,LET28 ;dificar tabla ".
48. 26A0 CD BE 18 CALL ESMEN1 ;Despliega mensaje "3.
49. 26A3 21 4B 13 LD HL,LET29 ;Graficar tabla ".
50. 26A6 CD BE 18 CALL ESMEN1 ;Despliega mensaje "4. Al-
;macenar tabla ".

```

APEND A

Sistema Graficador FAP-100

51.	26A9	21	62	13		LD HL,LET30	;Despliega mensaje "5.
52.	26AC	CD	DC	01		CALL ESMEN	;Salida ".
53.	26AF	C9				RET	;Termina.
54.	26B0	CD	6A	26	UN0	CALL MENU	;Despliega menu principal.
55.	26B3	CD	36	01	PIDE	CALL LETEC	;Recibe opcion elegida.
56.	26B6	FE	30			CP #30	;Checa si generar tabla.
57.	26B8	20	05			JR NZ,UN01	;Salta si no es.
58.	26BA	CD	D7	18		CALL GENTA	;Invoca generacion tabla.
59.	26BD	18	F1			JR UN0	;Recibe nueva eleccion.
60.	26BF	FE	31		UN01	CP #31	;Checa si cargar tabla.
61.	26C1	20	05			JR NZ,UN02	;Salta si no es.
62.	26C3	CD	F3	1C		CALL ACCETA	;Invoca accesa tabla.
63.	26C6	18	E8			JR UN0	;Recibe nueva eleccion.
64.	26C8	FE	32		UN02	CP #32	;Checa si modificar tabla.
65.	26CA	20	05			JR NZ,UN03	;Salta si no es.
66.	26CC	CD	38	1E		CALL MENMOD	;Invoca menu de modifica-
67.	26CF						ciones.
68.	26CF	18	DF			JR UN0	;Recibe nueva eleccion.
69.	26D1	FE	33		UN03	CP #33	;Checa si graficar tabla.
70.	26D3	20	05			JR NZ,UN04	;Salta si no es.
71.	26D5	CD	F4	14		CALL GRAF12	;Invoca graficas 12 bits.
72.	26D8	18	D6			JR UN0	;Recibe nueva eleccion.
73.	26DA	FE	34		UN04	CP #34	;Checa si almacenar tabla.
74.	26DC	20	05			JR NZ,UN05	;Salta si no es.
75.	26DE	CD	8E	1D		CALL TRACIN	;Invoca transmite a cinta.
76.	26E1	18	CD			JR UN0	;Recibe nueva eleccion.
77.	26E3	FE	35		UN05	CP #35	;Checa si es salida.
78.	26E5	20	CC			JR NZ,PIDE	;Salta si no es eleccion
79.	26E7						;valida.
80.	26E7	CF				RST #08	;Termina sistema grafica-
81.	26E8						;dor 12 bits FAP-100.
82.	26E8						

T a b l a d e S i m b o l o s

Nombre	Valor	Linea de Definicion
let23	1288	6
let24	12E2	7
let25	12F5	8
let26	130A	9
let27	131E	10
let28	1335	11
let29	1348	12
let30	1362	13
let31	136C	14
letec	0136	15
espa	013F	16
esmen	01DC	17
esment	18BE	18
posc	C479	19
genta	18D7	20
menmod	1E38	21
graf12	14F4	22
acceta	1CF3	23
tracin	1DBE	24
renpos	1144	25

menu	266A	28
uno	2680	54
pide	2683	55
uno1	26BF	60
uno2	26C8	64
uno3	26D1	69
uno4	26DA	73
uno5	26E3	77

## APENDICE B

Localidades de servicio para sistema graficador FAP-100.

NOMBRE	DIRECCION	FUNCION
BUFL	10FAH	Se utiliza para reconocer si el dispositivo con el que se establece comunicación con protocolo X-MODEM, genera o no respuesta.
BUFB	10FBH	Almacena parámetro para reconocer cuantas veces se ha repetido un bloque en comunicación con protocolo X-MODEM.
TOPE	10FC-FDH	En estas localidades se almacena la dirección de la última localidad que se transmite, o que se utiliza para almacenar datos, en comunicación X-MODEM.



DISP	1100H	Almacena el código con el que se selecciona un determinado dispositivo para comunicación X-MODEM.
MODO	1102H	El valor almacenado en esta localidad, determina el formato en que se transmiten los datos en comunicación X-MODEM (binario o ASCII).
DIR1	110B-OAH	Almacena la dirección inicial de un determinado archivo de datos.
BUFY	110C-ODH	Buffer de salida para coordenada X de dato formateado y listo para ser graficado.
BUFY	110E-OFH	Buffer de salida para coordenada Y de dato formateado y listo para ser graficado.
PLUMA	1110H	Almacena la palabra de control perteneciente a un par de datos unidos por alguno de los algoritmos de graficado.

BUF	1111-12H	Localidades para almacenar temporalmente un dato próximo a ser formateado.
DIFX	1113-14H	Se utiliza para almacenar la diferencia entre las coordenadas X de dos puntos. Se utiliza como parámetro para definir octante en algoritmo de rectas.
DIFY	1115-16H	Se utiliza para almacenar la diferencia entre las coordenadas Y de dos puntos. Se utiliza como parámetro para definir octante en algoritmo de rectas.
ABSX	1117-18H	Almacena el valor absoluto de la diferencia sobre el eje X de dos puntos.
ABSY	1119-1AH	Almacena el valor absoluto de la diferencia sobre el eje Y de dos puntos.
OCT	111BH	Palabra de control que define el octante en que se tiene que realizar

el trazo de los algoritmos de graficado.

BRX            111C-1DH      Almacena la diferencia de las coordenadas X de dos puntos para utilizarla como contador de etapas en los algoritmos de graficado.

BRY            111E-1FH      Almacena la diferencia de las coordenadas Y de dos puntos para utilizarla como contador de etapas en los algoritmos de graficado.

DIS            1120-21H      Discriminante utilizado en los algoritmos de graficado para definir dirección del trazo.

RAD            1122-23H      Almacena el valor del radio de un círculo para su trazado por el algoritmo implementado para ese fin.

CONDIR        1124-25H      El contenido de esta localidad es la última dirección utilizada por un archivo de datos.

COL            1126H          Número de columna en el que se posiciona al cursor de la terminal

de video.

ORGUSX	1127-28H	Coordenada X del origen relativo perteneciente a la tabla de usuario.
ORGUSY	1129-2AH	Coordenada Y del origen relativo perteneciente a la tabla de usuario.
RENG	112B-2CH	Número de renglon desplegado en pantalla durante la generación o modificación dato a dato de un archivo.
LONG	112D-2EH	Almacena temporalmente la longitud de un determinado archivo de datos.
DIVOR	112F-30H	Contiene el divisor utilizado durante el algoritmo de división.
FACTX	1131-32H	Factor de transformación para eje X durante la ejecución de los programas de translación, reducción y amplificación.
FACTY	1133-34H	Factor de transformación para eje Y durante la ejecución de los programas de translación, reducción

y amplificación.

ORGX	1135-36H	Coordenada X de origen relativo perteneciente a tabla de máquina de un archivo.
ORGY	1137-38H	Coordenada Y de origen relativo perteneciente a tabla de máquina de un archivo.
OP	1139H	Define la operación a ejecutarse durante los programas de transformación. 00= división, 01= multiplicación y 02= suma.
GRAD	113A-3BH	Almacena el ángulo de rotación utilizado para la alternativa de transformación de este mismo nombre.
SEN	113C-3DH	Valor de la función seno del ángulo de rotación.
COS	113E-3FH	Valor de la función coseno del ángulo de rotación.
***	1140-43H	Zona libre.

RENPOS	1144H	Número de renglon en que se posiciona el cursor de la terminal de video.
LET1	1145-59H	Mensaje:"DIRECCION INICIAL: "".
LET2	115A-5FH	Area de memoria reservada para dato proveniente del teclado.
LET3	1160-66H	Mensaje:"REGLON".
LET4	1167-89H	Mensaje:" COOR X COOR Y CONTROL".
LET5	118A-9CH	Mensaje:"DIRECCION FINAL: "".
LET6	119D-AFH	Mensaje:"LONGITUD BUFFER: "".
LET7	11B0-B6H	Mensaje:"TRANS.".
LET8	11B7-C9H	Mensaje:"CR PARA TRANSMITIR".
LET9	11CA-DOH	Mensaje:"LISTO&".
LET10	11D1-E2H	Mensaje:"REGLON NUMERO? "".
LET11	11E3-E8H	Mensaje:"NORM.".

LET12	11E9-F4H	Mensaje:"FACTOR X? "
LET13	11F5-1200H	Mensaje:"FACTOR Y? "
LET14	1201-1EH	Mensaje:"MODIFICA TABLA USUARIO? (S/N)":
LET15	121F-37H	Mensaje:"MENU DE MODIFICACIONES".
LET16	1238-53H	Mensaje: "1. Modificacion dato a dato".
LET17	1254-67H	Mensaje:"Escalamiento : "
LET18	1268-78H	Mensaje:"2. Reduccion".
LET19	1279-8DH	Mensaje:"3. Amplificacion".
LET20	128E-A0H	Mensaje:"4. Translacion".
LET21	12A1-BOH	Mensaje:"5. Rotacion".
LET22	12B1-BAH	Mensaje:"6. Salida".
LET23	12BB-E1H	Mensaje:"SISTEMA GRAFICADOR 12 BITS FAP-100".

LET24	12E2-F4H	Mensaje:"MENU PRINCIPAL".
LET25	12F5-1309H	Mensaje:"0. Generar tabla".
LET26	130A-1DH	Mensaje:"1. Cargar tabla".
LET27	131E-34H	Mensaje:"2. Modificar tabla".
LET28	1335-4AH	Mensaje:"3. Graficar tabla".
LET29	134B-61H	Mensaje:"4. Almacenar tabla".
LET30	1362-6BH	Mensaje:"5. Salida".
LET31	136C-8CH	Mensaje: "G. R. Espinosa P. enero/86".
LET32	138D-96H	Mensaje:"ANGULO?--".

NOTA. Las áreas de memoria ocupadas por los letreros (LETxx), tienen almacenadas en la última localidad que ocupan un ODH, para que sean compatibles con ESMEN y ESMEN1. Además, todos están estructurados a base de los códigos ASCII de cada uno de los caracteres que



los forman.

NOTA. & equivale a sonido de campana.

## APENDICE C

### Algoritmo para división entera binaria.

El origen de este de algoritmo, se debe al hecho de que en muchas computadoras, la división es una operación considerablemente más lenta que la multiplicación, por lo que su fin principal, es el de acelerar el proceso para la realización de la primera.

Dentro del sistema graficador, adicionalmente a esto, la utilización de este método, nace de la restricción de la microcomputadora huesped, la cual es incapaz de ejecutar este tipo de operación.

El algoritmo que ahora se presenta, se basa en el hecho de que la división es la operación inversa a la multiplicación, ya que se tiene que

$$\text{dividendo} = \text{divisor} * \text{cociente}$$

de donde

dividendo / divisor = cociente

tomando en cuenta que la división no es conmutativa, excepto cuando dividendo = divisor.

Así, mientras que la multiplicación es una operación de corrimiento-adición del multiplicando, la división es una operación de corrimiento-substracción del divisor, en donde el resultado de una substracción determina la siguiente operación, por lo que la división tiene una inherente dependencia sobre la operación previa, hecho que ocurre serialmente para todos los subsecuentes ciclos.

La división binaria es considerablemente simple, ya que únicamente se tienen dos alternativas de selección (0 o 1). Si el divisor es más pequeño que el residuo parcial, el bit correspondiente del cociente es 1 y una substracción es realizada. Si es más grande, el bit correspondiente del cociente es 0 y la substracción no se lleva a cabo. El cociente se va definiendo desde el bit más significativo al menos, mientras que el residuo se determina automáticamente al obtener el cociente.

En base a esto, un primer método que se puede implementar, es el conocido como " división secuencial de corrimiento-substracción/adición ".

La lógica que implementa éste, es la de posicionar el número de bits de que consta el divisor, con los más significativos del dividendo (o residuo parcial) y realizar una substracción. Si el resultado es positivo (dividendo

mayor que divisor), un 1 es colocado en el bit correspondiente del cociente. Con esto, el residuo parcial obtenido (resultado de la resta), se recorre a la izquierda junto con el siguiente bit del dividendo y con el divisor posicionado correctamente, el proceso se repite.

Si el resultado de la sustracción es negativo (divisor mayor que dividendo), un 0 es colocado en el cociente, y el dividendo (o residuo parcial) es reestablecido adicionando nuevamente al divisor, con lo que se recorre a la izquierda para tener acceso al siguiente bit, y el proceso se repite.

Si se supone que existe una probabilidad igual para generar un dígito de cociente 0 o 1, entonces  $n$  sustracciones y  $n/2$  adiciones son requeridas para un divisor de  $n$  bits, además de  $n$  corrimientos de 1 bit a la izquierda.

El método conocido como "Division secuencial de corrimiento-sustracción/adición sin reestablecimiento", permite reducir este número de operaciones, eliminando las adiciones, con lo que el proceso se acelera aún más.

Este toma como base que, cuando el resultado de la sustracción es positivo, entonces el dividendo (A) es recorrido a la izquierda y el divisor (B) es sustraído nuevamente, con lo que la operación que se realiza es

$$2A - B$$

mientras que si el resultado es negativo, el residuo parcial es realmacenado realizando la suma  $A+B$ , y esto es recorrido a la izquierda para sustraer B otra vez, lo que es equivalente

a realizar la operacion

$$2(A + B) - B = 2A + B$$

Con esto, en el método modificado solo se tienen dos tipos de operaciones,  $2A+B$  y  $2A-B$ , y el bit del cociente es colocado como en la división con reestablecimiento, esto es, si el signo de A (residuo parcial) es positivo, el bit correspondiente es 1, en caso contrario es 0.

Para su funcionamiento, la primera vez el dividendo es recorrido y el divisor es restado. También, si el último bit del cociente es cero, entonces el residuo parcial previo debe ser reestablecido para obtener el residuo final correcto. Con esto, en división sin reestablecimiento, entre  $n$  o  $n+1$  corrimientos-substracción/adición son requeridos.

Durante la implementación en computadora digital de este método, la substracción del divisor se puede realizar adicionando el complemento a dos de éste, y el resultado de la operación se obtiene en el registro de banderas. Además, el espacio destinado para el dividendo, puede ser ocupado simultaneamente por el cociente, ya que a medida que los bits del primero se dejan de utilizar, se crea proporcionalmente los del segundo.

A continuación, se expone un ejemplo numérico de este método, en donde las longitudes de los números corresponden a las utilizadas dentro del sistema graficador.

A.Q = 0000 0000 0010 0111 0000 1111 (+9999)

B = 0000 0000 0101 (+5)

Bneg + 1 = 1111 1111 1011 (-5)

0000 0000 0010	0111 0000 1111	
0000 0000 0100	1110 0001 111	1er. corrimiento
1111 1111 1011		
-----		
1111 1111 1111		resta
1111 1111 1111	1110 0001 1110	Q coloca Q0=0
1111 1111 1111	1100 0011 110	2do. corrimiento
0000 0000 0101		
-----		
0000 0000 0100		adiciona
0000 0000 0100	1100 0011 1101	Q coloca Q0=1
0000 0000 1001	1000 0111 101	3er. corrimiento
1111 1111 1011		
-----		
0000 0000 0100		resta
0000 0000 0100	1000 0111 1011	Q coloca Q0=1
0000 0000 1001	0000 1111 011	4o. corrimiento
1111 1111 1011		
-----		
0000 0000 0100		resta
0000 0000 0100	0000 1111 0111	Q coloca Q0=1
0000 0000 1000	0001 1110 111	5o. corrimiento
1111 1111 1011		
-----		
0000 0000 0011		resta
0000 0000 0011	0001 1110 1111	Q coloca Q0=1
0000 0000 0110	0011 1101 111	6o. corrimiento
1111 1111 1011		
-----		
0000 0000 0001		resta
0000 0000 0001	0011 1101 1111	Q coloca Q0=1
0000 0000 0010	0111 1011 111	7o. corrimiento
1111 1111 1011		
-----		
1111 1111 1101		resta

1111 1111 1101	0111 1011 1110	Q	coloca Q0=0
1111 1111 1010	1111 0111 110		8o. corrimiento
0000 0000 0101			
-----			
1111 1111 1111			suma
1111 1111 1111	1111 0111 1100	Q	coloca Q0=0
1111 1111 1111	1110 1111 100		9o. corrimiento
0000 0000 0101			
-----			
0000 0000 0100			suma
0000 0000 0100	1110 1111 1001	Q	coloca Q0=1
0000 0000 1001	1101 1111 001		10o. corrimiento
1111 1111 1011			
-----			
0000 0000 0100			resta
0000 0000 0100	1101 1111 0011	Q	coloca Q0=1
0000 0000 1001	1011 1110 011		11o. corrimiento
1111 1111 1011			
-----			
0000 0000 0100			resta
0000 0000 0100	1011 1110 0111	Q	coloca Q0=1
0000 0000 1001	0111 1100 111		12o. corrimiento
1111 1111 1011			
-----			
0000 0000 0100			resta
0000 0000 0100	0111 1100 1111		coloca Q0=1
R	Q		
(+4)	(+1999)		

Dentro del sistema graficador, la implementación de este algoritmo se hizo por medio de la subrutina DIV, y tomando en cuenta las limitaciones de circuiteria propia del microprocesador Z-80, el espacio para cada uno de los elementos de la division queda de la siguiente forma:

El dividendo de 24 bits se implementa dentro de dos registros de 16 bits, ocupando los 12 bits menos significativos de cada uno de ellos.

El divisor de 12 bits, es implementado en un tercer

registro de 16 bits, utilizando los 12 menos significativos de este.

Todos los bits no utilizados en los tres registros anteriores, son colocados a cero siempre que se realice

El listado completo de esta subrutina se puede consultar en el apéndice A de este trabajo.



## APENDICE D

### Algoritmo de adición múltiple para multiplicación.

La implementación de este algoritmo, se basa en el hecho de que la operación multiplicación, es un proceso de adiciones sucesivas, donde el valor de los sumandos es siempre el mismo, y corresponde al determinado por el del multiplicando. Además, el número de sumando es el valor proporcionado por el del multiplicador.

Esto es, cuando se requiere llevar a cabo la operación multiplicación entre dos números  $n$  y  $m$ , donde  $m$  es el multiplicando y  $n$  el multiplicador, el proceso se puede sustituir de la siguiente manera

$$m \cdot n = m + m + m + \dots + m$$

donde el número de sumandos es  $n$ .

El resultado obtenido por cualquiera de las dos formas, es el mismo.

El ejemplo numérico que a continuación se expone, aclara

el significado de esta analogía.

Se desea obtener el producto de los números 4 y 8. Si se realiza por el primer método, este es 32. Si se utiliza el segundo método, la suma adquiere la siguiente estructura:

$$4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 = 32$$

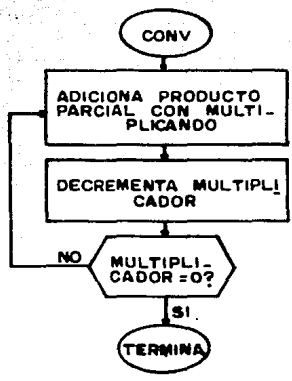
donde, como se puede comprobar, el resultado es el mismo.

Al igual que la multiplicación, el algoritmo es conmutativo, es decir, tomando en cuenta a los números  $n$  y  $m$ , que tanto uno como el otro puede tomar el papel de multiplicando o multiplicador.

Dentro del sistema graficador, este algoritmo se implementa por medio de la subrutina CONV, la cual requiere para su funcionamiento, que el multiplicando se almacene en el registro DE y el multiplicador en el BC. El producto lo genera almacenado en el HL.

El espacio que dispone para almacenar el valor de este último, establece los límites de funcionamiento de esta subrutina, pues como el registro HL tiene una longitud de 16 bits, entonces el producto solo se podrá ejecutar tomando al multiplicando como un número de 8 bits y al multiplicador de la misma longitud, o en su defecto, tomando al primero con una longitud de 12 bits, y al segundo con 4. La primera forma se utiliza para almacenar datos, y la última en la alternativa de amplificación.

El diagrama a bloques de esta subrutina es el que se muestra a continuación, y el listado completo de la misma se puede consultar en el apéndice A de este trabajo.



## APENDICE E

Algoritmo para obtención funciones seno y coseno.

Uno de los problemas graves que se presenta en la implementación de la alternativa de transformación para rotar una gráfica, es que las ecuaciones para lograr ésta, se basan en la obtención de las funciones seno y coseno del ángulo de rotación, operaciones que la microcomputadora huésped es incapaz de realizar.

Para esto, se realizó la implementación del algoritmo que en este apéndice se presenta. Este, es diseñado específicamente para ser programado en lenguaje de bajo nivel, ya que el resultado final se basa únicamente en operaciones de adición y substracción.

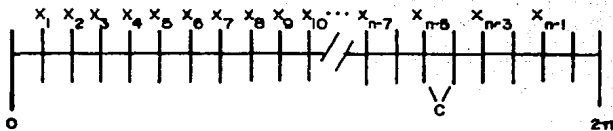
El algoritmo utiliza un método recursivo que toma como valores iniciales los resultados obtenidos para la función seno y la función coseno con un ángulo igual a cero. Esto es:

Para  $\theta = 0$ , tenemos que

$$\cos \theta = 1 \quad \dots(c1)$$

$$\text{sen } \theta = 0 \quad \dots(c2)$$

El ángulo requerido por el algoritmo debe ser dado en radianes forzosamente y dentro de un intervalo de 0 a  $2\pi$ . Además, este intervalo no es continuo, sino que esta compuesto por datos discretos con un intervalo entre ellos constante, como se muestra a continuación



en donde

$$x_i - x_{(i-1)} = C \quad \dots(c3)$$

por lo que C es la magnitud del intervalo entre valores.

Para su funcionamiento, entre más pequeño sea el valor de C, el resultado es más preciso, y debido a que se supone este así, se hace la consideración de que

$$\cos C = 1 \quad \dots(c4)$$

$$\text{sen } C = C \quad \dots(c5)$$

Tomando lo anterior en cuenta, la implementación del algoritmo se lleva a cabo como se muestra a continuación.

Se tiene de c3 que

$$X_i = X(i-1) + C \quad \dots(c6)$$

por lo que

$$\text{sen}(X_i) = \text{sen}(X(i-1) + C) \quad \dots(c7)$$

$$\text{cos}(X_i) = \text{cos}(X(i-1) + C) \quad \dots(c8)$$

pero por identidades trigonométricas se sabe que

$$\text{sen}(A + B) = \text{sen}A \text{cos}B + \text{cos}A \text{sen}B$$

$$\text{cos}(A + B) = \text{cos}A \text{cos}B - \text{sen}A \text{sen}B$$

y aplicando estas a las ecuaciones c7 y c8, se genera el siguiente resultado

$$\text{sen}(X(i-1)+C) = \text{sen}(X(i-1))\text{cos}(C) + \text{cos}(X(i-1))\text{sen}(C) \quad \dots(c9)$$

$$\text{cos}(X(i-1)+C) = \text{cos}(X(i-1))\text{cos}(C) - \text{sen}(X(i-1))\text{sen}(C) \quad \dots(c10)$$

de donde, si se sustituyen las ecuaciones c4 y c5 en c9 y c10, entonces se obtiene

$$\text{sen}(X(i-1)+C) = \text{sen}(X(i-1)) + C\text{cos}(X(i-1)) \quad \dots(c11)$$

$$\text{cos}(X(i-1)+C) = \text{cos}(X(i-1)) - C\text{sen}(X(i-1)) \quad \dots(c12)$$

pero tomando en cuenta a c7 y c8, el resultado final es

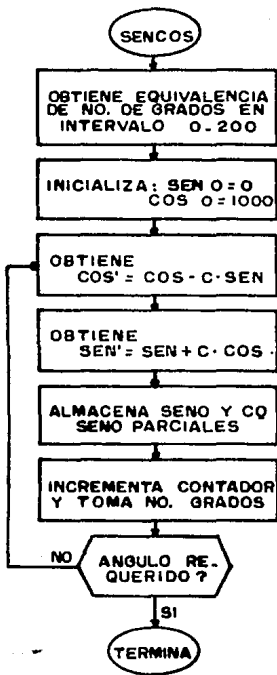
$$\text{sen } (X_1) = \text{sen } (X_{(i-1)}) + C \cos (X_{(i-1)}) \quad \dots(c13)$$

$$\text{cos } (X_1) = \text{cos } (X_{(i-1)}) - C \text{sen } (X_{(i-1)}) \quad \dots(c14)$$

que son las ecuaciones por medio de las cuales se obtiene el valor de un determinado ángulo en función del valor del ángulo anterior a el, partiendo del valor inicial  $\theta = 0$ .

Para el desarrollo de éste dentro del sistema graficador, se modificó de tal forma que se permitiera hacer uso de un valor de C mayor que la unidad y además que el valor del ángulo pudiera darse en grados. Esto trajo como consecuencia la implementación de un algoritmo de multiplicación, ya que el de adición múltiple resulto insuficiente (ver apéndice F).

Así, el ángulo en radianes se maneja en un rango de 0 a 200, esto es, que 200 equivale a  $2\pi$  y 0 a 0. Además el resultado de las funciones seno y coseno se expandió de manera que éste se da con una variación de -1000 a 1000 en lugar de -1 a 1, y finalmente se implemento el escalamiento adecuado para que el usuario proporcione el ángulo en grados, siguiendo la equivalencia de que 360 grados es igual a 200. Con esto, el diagrama a bloques de la subrutina SENCOS, por medio de la cual se implementó el algoritmo, es el que se muestra a continuación. El listado completo de ella se muestra en el apéndice A de este trabajo.





## APENDICE F

### Algoritmo para multiplicación entera binaria.

En aritmética entera binaria, existen métodos sencillos de multiplicación de dos números. Uno de ellos, es el que se implementó en la subrutina CONV, donde el multiplicando se suma el número de veces contenido en el multiplicador. La limitante de este tipo de procedimiento, es que el producto generado, no debe tener una longitud mayor de 16 bits.

Para la implementación del programa de rotación, se requiere sin embargo, de la multiplicación de un número de 12 bits por otro de igual longitud, lo que implica como resultado un número de 24 bits, por lo que su implementación con el algoritmo de adición múltiple no es posible.

Para lograr llevar a cabo este tipo de operación, se implementó el método conocido como corrimiento-adición secuencial, el cual permite ejecutar el producto de dos números de hasta 16 bits de longitud cada uno. Con este método, una multiplicación de dos números de  $n$  bits

$$A = a(n-1) \ a(n-2) \ a(n-3) \ \dots \ a_0$$

$$\text{y } B = b(n-1) \ b(n-2) \ b(n-3) \ \dots \ b_0$$

da como resultado un número de  $2n$  bits

$$P = p(2n-1) \ p(2n-2) \ p(2n-3) \ \dots \ p_0$$

El algoritmo se basa en el estado del bit menos significativo del multiplicador en un momento determinado.

Si los elementos se almacenan de manera que el multiplicando se almacene en un registro de  $n$  bits, y el multiplicador junto con el producto (inicializado en cero) en un registro de  $2n$  bits, como se muestra a continuación

MULTIPLICANDO	PRODUCTO PARCIAL	MULTIPLICADOR
$A_{n-1} \ \dots \ A_1 \ A_0$	$P_{n-1} \ \dots \ P_1 \ P_0$	$B_{n-1} \ \dots \ B_1 \ B_0$

entonces el algoritmo se basará para su decisión en el estado que muestre el bit menos significativo del multiplicador, bajo las siguientes reglas:

- Si el bit menos significativo del multiplicador es cero, el registro  $2n$  es recorrido a la derecha una posición y el signo del multiplicando es copiado en el bit más significativo de este registro.
- Si el bit menos significativo del multiplicador es

uno, el multiplicando es adicionado a los n bits más significativos del registro 2n, y entonces es recorrido a la derecha una posición, con lo que el signo del multiplicando se copia en el bit más significativo del registro 2n.

A continuación se muestra un ejemplo numérico de la multiplicación de dos números de 16 bits cada uno.

multiplicando (+9999)	producto parcial	multiplicador (+1000)
0010011100001111	0000000000000000	0000001111101000
0=corrimento		
copia signo	0000000000000000	0000000111110100
0=corrimento		
copia signo	0000000000000000	0000000011110100
0=corrimento		
copia signo	0000000000000000	0000000001111101
1=suma-corrimento	0010011100001111	
	-----	
	0010011100001111	
	0010011100001111	0000000001111101
copia signo	0001001110000111	1000000000111110
0=corrimento		
copia signo	0000100111000011	1100000000011111
1=suma-corrimento	0010011100001111	
	-----	
	0011000011010010	
	0011000011010010	1100000000011111

```

copia signo 0001100001101001 0110000000001111
1=suma-corrimiento
0010011100001111
-----
001111101111000
001111101111000 0110000000001111

copia signo 000111110111100 0011000000000111
1=suma-corrimiento
0010011100001111
-----
0100011011001011
0100011011001011 0011000000000111

copia signo 0010001101100101 1001100000000011
1=suma-corrimiento
0010011100001111
-----
0100101001110100
0100101001110100 1001100000000011

copia signo 0010010100111010 0100110000000001
1=suma-corrimiento
0010011100001111
-----
0100110001001001
0100110001001001 0100110000000001

copia signo 0010011000100100 1010011000000000
0=corrimento

copia signo 0010011000100100 0100110000000000
0=corrimento

copia signo 0001001100010010 0101001100000000
0=corrimento

copia signo 0000100110001001 0010100110000000
0=corrimento

copia signo 0000001011000100 1001010011000000
0=corrimento

copia signo 0000001001100010 0100101001100000
0=corrimento
    
```

O=corrimento

copia signo 0000000100110001 0010010100110000

O=corrimento

copia signo 0000000010011000 1001001010011000

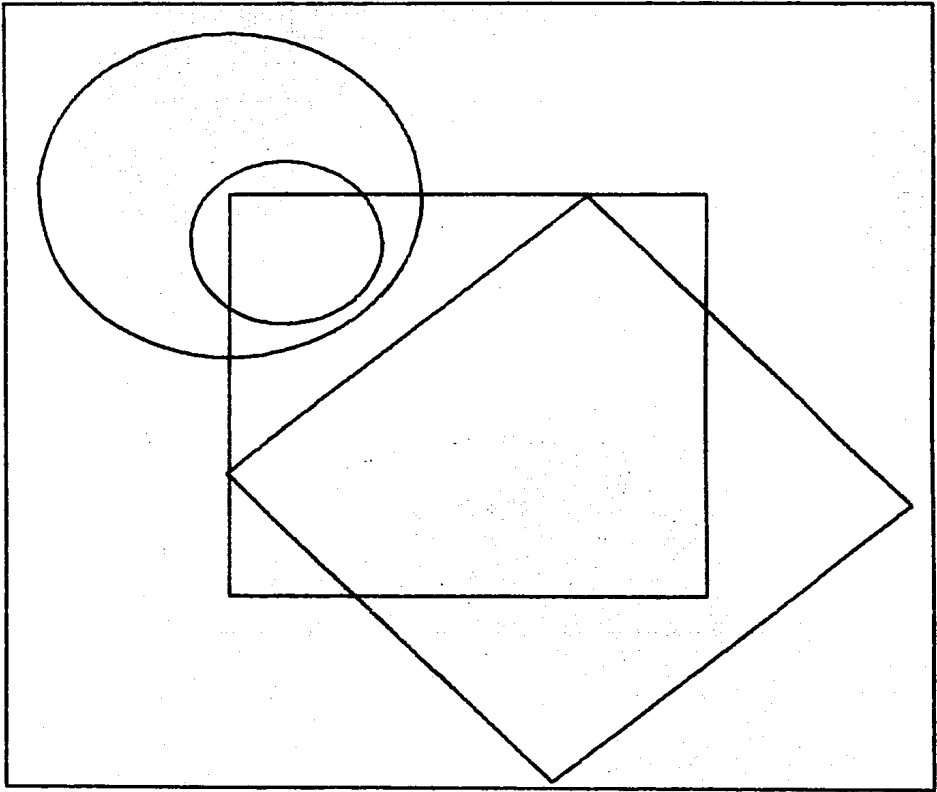
producto  
(+9999000)

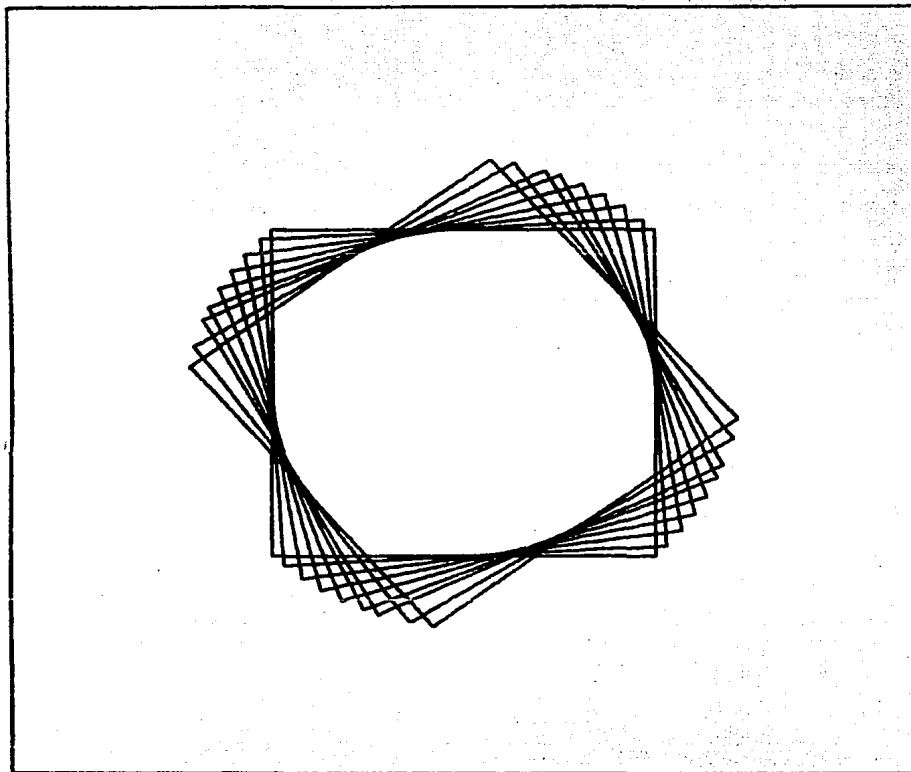
Dentro del sistema de graficado, este algoritmo se implementó por medio de MULT16, la cual requiere para su funcionamiento que el multiplicando se le proporcione en el registro BC, mientras que el multiplicador en el DE. El producto se obtiene a lo largo de HL y DE.

El listado completo de esta subrutina se puede consultar en el apéndice A de este trabajo.

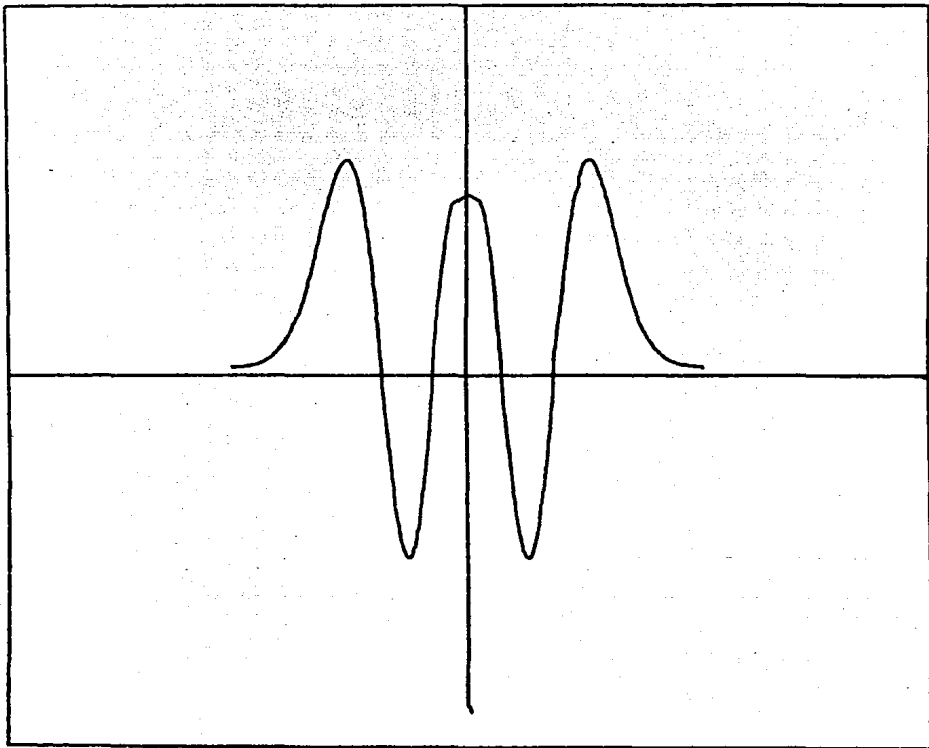
# APENDICE G

Ejemplos de gráficas.

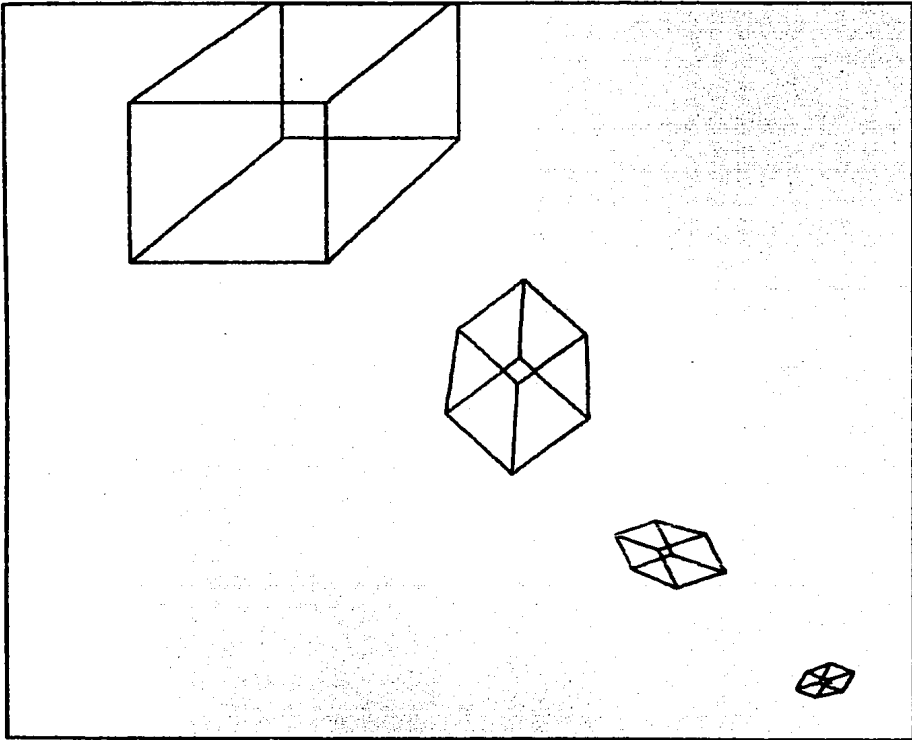


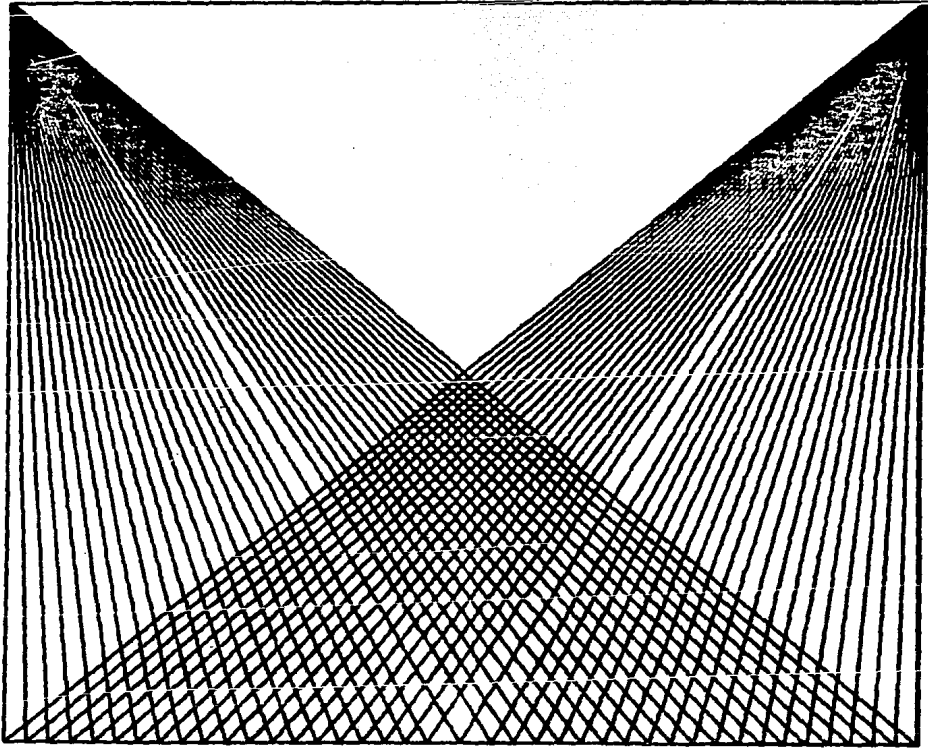


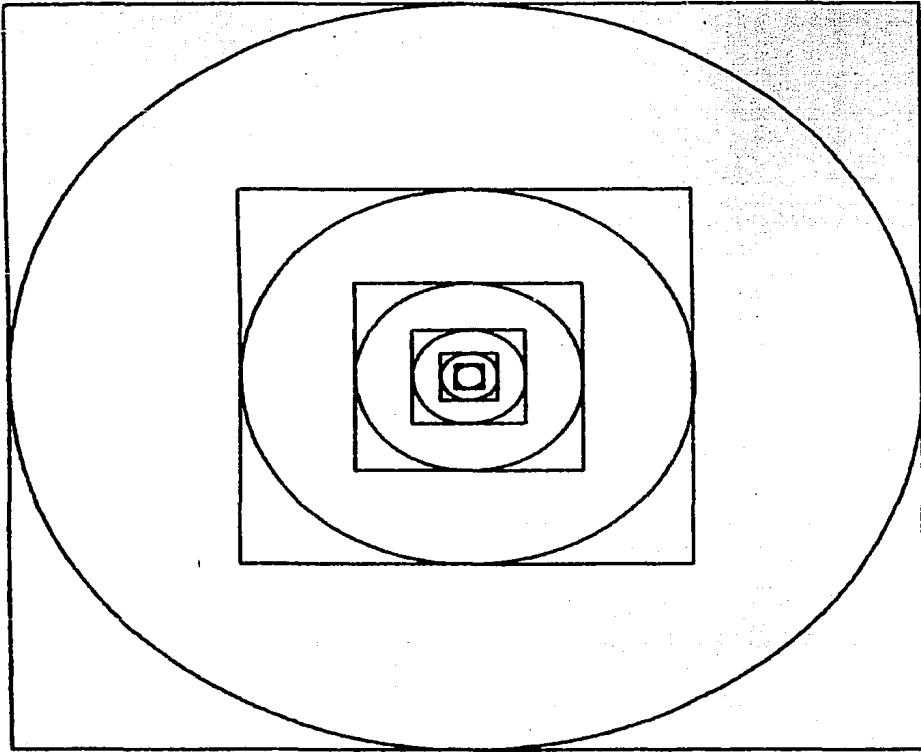


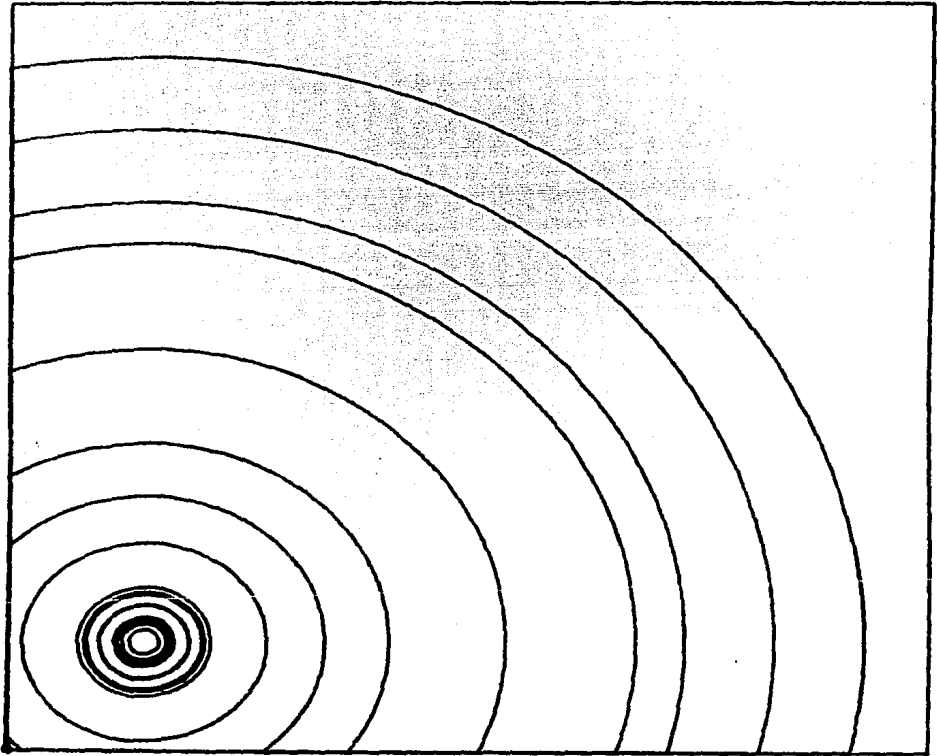


NOTA. Los datos con que se construyó esta gráfica, fueron proporcionados por los alumnos de la ESFM Concepción Mejía García y Jorge Aguilar Hernández.









## BIBLIOGRAFIA

1. Foley J. D.: 1982. "Fundamentals of Interactive Computer Graphics". ed. Addison Wesley. Philippines.
2. Computervision. 1981. "The CAD / CAM Handbook". ed. Computervision. USA.
3. Ramírez R. A., Rodea G. O.: 1984. Tesis de licenciatura "Diseño y Construcción de una microcomputadora basada en el bus IEEE 696. UNAM. México, D. F.
4. Zilog. 1982. "Z-80 Data Book". ed. Zilog. USA.
5. National. 1978. "Memory Data Book". ed. National Semiconductor Inc. USA.
6. Libes S., Garetz M. 1981. "Interfacing to S-100/ IEEE - 696 Microcomputer". ed. Osborne/Mc Graw Hill. USA.
7. National. 1981. "Logic Data Book". ed. National Semiconductor Inc. USA.
8. Ciarcia S. 1981. "Build your own Z-80 computer". ed. Mc Graw Hill. USA.

9. National. 1980. "Data conversion/adquisition Data Book". ed. National Semiconductor Inc. USA.
10. Motorola. 1978. "Motorola Linear Data Book". ed. Motorola. USA.
11. Texas Instruments. 1979/80. "Manual de semiconductores de silicio. Tipos preferidos en Latinoamerica": ed. Texas Instruments. México D. F.
12. National. 1978. "Voltage Regulator Data Book". ed. National Semiconductor Inc. USA.
13. Sproull Robert F., Newman W. M. 1984. "Principles of - Interactive Computer Graphics". 2nd. ed., ed. Mc Graw Hill. USA.
14. Leventhal L. A. 1979. "Z-80 assembly language programming". ed. Mc Graw Hill. USA.
15. Cavanagh J. J. F. 1984. "Digital computer arithmetic. Design and implementation". ed. Mc Graw Hill. USA.
16. Harrington S. 1983. "Computer Graphics. A programming approach". ed. Mc Graw Hill. USA.
17. Grappel R. 1979. "An easy way to calculate sines and - cosines". Artículo incluido en: "Numbers in Theory and Practice". Byte Publications, Inc. USA.
18. Brodlie. 1985. "Mathematical Methods in Computer Gra - phics and Design". ed. Academic Press. USA.

19. Hornitz, Sahni. 1984. "Fundamentals of Computer Algorithms". ed. Computer Science Press. USA.
20. Harris. 1985. "Computer Graphics and Applications". ed. Chapman and Hall. USA.
21. Auslander, Sagues. 1985. "Microprocessors for measurement and control". ed Mc Graw Hill. USA.