

ESCUELA UNIVERSITARIA DE TELECOMUNICACION DE LAS PALMAS  
DE GRAN CANARIA

TITULO: APLICACIONES DE DISEÑOS GRAFICOS EN SISTEMA  
CAD, USANDO COMO PERIFERICO DE SALIDA UN TRAZADOR DE  
GRAFICOS ADAPTADO COMO PLOTTER.

AUTOR:

TUTOR:

MARCELA A. RODRIGUEZ DOMINGUEZ

SEBASTIAN SUAREZ GIL

## INTRODUCCION

La finalidad de este proyecto ha sido la construcción de figuras en tres dimensiones y dibujadas con el Plotter X-Y PM 8043.

Para conseguir el objetivo final de este proyecto se ha descrito dos partes.

Una primera parte formado por dos programas bien diferentes .

El primer programa es el EXA y sus correspondientes módulos . Cuya misión es la de generar figuras en tres dimensiones . Está realizado en BASIC con el terminal HEWLETT - PACKARD HP-3000.

El segundo programa es el tridimensional que esta compuesto de un menú donde estan los ficheros correspondientes a las figuras . Y el propio programa tridimensional que se basa en la interpretación de los ficheros generados por el programa EXA . El programa tridimensional está realizado en PLM 80 por el MDS-221.

Una segunda parte formada por un Hardware que conecta el MDS 221 y el Plotter X-Y PM 8043. Utilizando la salida PUNCH del MDS-221 que se adapto para que funcionará como puerto de salida.

PARTE I

## INDICE

PAGINA

INTRODUCCION

PARTE I

DISPOSITIVOS PERIFERICOS.....	001
TECLADOS.....	001
1. GENERALIDADES.....	001
2. PULSADORES.....	002
2.1 Pulsadores mecánicos convencionales.....	003
2.2 Pulsadores mecánicos de lámina flexible..	003
2.3 Pulsadores mecánicos de bovedillas.....	004
2.4 Pulsadores reed.....	004
2.5 Pulsadores capacitivos.....	005
2.6 Pulsadores de núcleo magnético.....	006
2.7 Pulsadores de efecto de Hall.....	007
3. TECLAS.....	010
3.1 Rotulación.....	011
3.2 Perfil.....	014
VISUALIZADORES TRC.....	016

1.	IDEAS SOBRE EL VISUALIZADOR TRC.....	016
1.1	Funcinamiento del TRC.....	017
1.2	Señales de control de un visualizador TRC.....	020
1.3	Fuciones a realizar por el visualizador TRC.....	023
	CASSETTES.....	028
1.	INTRODUCCION.....	028
	DISCOS FLEXIBLES.....	031
1.	INTRODUCCION.....	031
2.	CARACTERISTICAS DE LOS DISCOS.....	032
2.1	Capacidad de almacenamiento.....	033
	IMPRESORAS.....	035
1.	LAS CARACTERISTICAS DE LAS IMPRESORAS.....	035
2.	IMPRESORAS VERSION TERMINAL Y TELEIMPRESORAS..	035
3.	CLASIFICACION DE LAS IMPRESORAS.....	035
3.1	Impresoras de impacto.....	035
3.2	Versión de matriz de puntos.....	036
3.3	Versión de caracteres de línea de tiempo..	036
4.	CARACTERISTICAS DE LAS IMPRESORAS.....	036

5.	LOS CARACTERES DE LAS IMPRESORAS DE TIEMPO DE IMPACTO.....	037
6.	IMPRESORAS DE TIEMPO DE IMPACTO.....	045
7.	IMPRESORAS DE TIEMPO DE NO IMPACTO.....	049
8.	IMPRESORAS DE MATRIZ DE PUNTOS.....	052
9.	CARACTERES DE LA IMPRESORAS CON MATRIZ DE PUNTOS.....	054
10.	LINEA Y TIEMPO DE IMPACTO DE LAS IMPRESORAS DE MATRIZ DE PUNTOS.....	058
11.	MATRIZ DE PUNTOS PARA IMPRESORAS DE NO IMPACTO.....	059
11.1	Impresoras t�rmicas.....	059
11.2	Impresoras electrosensitivas.....	064
11.3	Impresoras electrost�ticas.....	064
11.4	Impresoras de inyecci�n.....	066
12.	TRATAMIENTO DEL PAPEL.....	069
	TRANSFORMACIONES Y PROYECCIONES EN TRES DIMENSIONES.....	077
1.	INTRODUCCION.....	077
2.	CAMBIO DE ESCALA EN TRES DIMENSIONES.....	078
3.	ROTACION EN TRES DIMENSIONES.....	080
4.	REFLEXION EN TRES DIMENSIONES.....	082
5.	TRASLACION EN TRES DIMENCIONES.....	084

6.	PROYECCION ISOMETRICA.....	085
7.	PROYECCION DIMETRICA.....	087

PARTE II

1.	INTERFACE DE CONEXION ENTRE EL SISTEMA DE DESARROLLO Y EL PLOTTER.....	089
1.1	DESCRIPCION GENERAL.....	089
1.2	DESCRIPCION DE LOS INTEGRADOS.....	090
1.3	EXPLICACION DEL HARDWARE.....	090
1.4	ESQUEMA DEL HARDWARE	
1.5	PRESUPUESTO	
2.	PROGRAMA EXA.....	097
2.1	ANALISIS DEL PROGRAMA EXA.....	097
2.2	CARACTERISTICAS GENERALES.....	097
2.3	MODULO 1. PROGRAMA EXA.....	098
2.4	MODULO 2. CONFIGURACION DEL DIBUJO.....	104
2.5	MODULO 3. TRASLACION DEL DIBUJO.....	109
2.6	MODULO 4. ESTRECHAMIENTO DEL DIBUJO.....	115
2.7	MODULO 5. ROTACION DEL DIBUJO.....	122
2.8	MODULO 6. REFLEXION DEL DIBUJO.....	129
2.9	MODULO 7. PROYECCION DIMETRICA DEL DIBUJO.....	136
2.10	MODULO 8. PROYECCION ISOMETRICA DEL	

DIBUJO.....	142
2.11 MODULO 9. PROYECCION CONICA DEL DIBUJO...	148
3. PROGRAMA TRIDIMENSIONAL.....	155
3.1 ANALISIS DEL PROGRAMA TRIDIMENSIONAL.....	155
3.2 CARACTERISTICAS GENERALES.....	155
3.3 PROGRAMA MENU.....	155
3.4 PROGRAMA TRIDIMENSIONAL.....	156
3.5 LISTADO DEL PROGRAMA EXA. Y SUS MODULOS	
3.6 LISTADO DE FICHEROS	
3.7 LISTADO DEL PROGRAMA TRIDIMENSIONAL .	

MENU



## DISPOSITIVOS PERIFERICOS

Los dispositivos periféricos utilizados son: Teclados , Visualisadores TRC , Cassettes , Discos Flexibles , Impresoras .

### TECLADOS

#### 1. GENERALIDADES

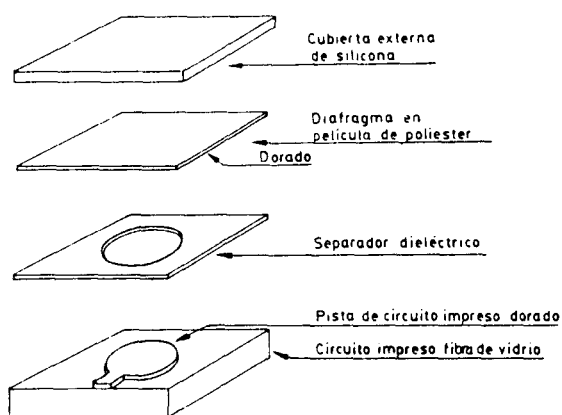
Se denomina "teclado" al género de periféricos de entrada , constituidos por un conjunto de botones pulsadores , de tal modo que cada botón se corresponda con determinado carácter , función , instrucción o idea .

El tipo de teclas sí como su número y distribución vendrán determinados por la aplicación correcta que se desea realizar , por lo que no existen modelos genéricos , sino desarrollos específicos .

El número de teclados conectados a un determinado sistema es sumamente variable , oscilando entre cero ( tal como en sistemas muy simples de instrumentación o automatización ) y varias decenas ( tal como en sistemas multiterminal de recogida y consulta de bancos de datos ); no obstante , en sistemas basados en microprocesadores lo más común es disponer de una unidad

, desde donde se suministran al sistema las informaciones básicas en cuanto a selección y control de programas de variables.

Físicamente, el teclado acostumbra a ir asociado a otro periféricos de salida, tal como una impresora, una pantalla



o un visualizador, con lo que el operador obtiene una comunicación bidireccional con el sistema.

## 2. PULSADORES

El componente básico de un teclado es el pulsador individual.

Generalmente, cada tecla controla un simple interruptor que permanece abierto mientras el pulsador está en descanso y cierra cuando el pulsador es

deprimido.

Existe un sin número de variaciones , relacionándose más adelante los tipos más usuales.

El concepto de conmutación de estado sólido se aplica a veces al describir elementos sin contacto , aunque hablando con propiedad solamente se puede aplicar a una limitada variedad entre la que se cuentan los de efecto Hall o elementos fotosensibles.

### 2.1 PULSADORES MECANICOS CONVENCIONALES

Los interruptores mecánicos son utilizados básicamente por su economía (fig.8.1).

El movimiento mecánico actúa directamente sobre los contactos.

Permiten , sin grandes complicaciones , configuraciones de contactos múltiples.

### 2.2 PULSADORES MECANICOS DE LAMINA FLEXIBLE

Este tipo de interruptor (fig.8.2) , formados por una serie de láminas sobrepuestas , se basa en la deflexión de un diafragma flexible , dorado por su cara inferior , que permite establecer contacto con un circuito impreso a través de aperturas practicadas en un separador dieléctrico.

Una cubierta de silicona protege los contactos contra los contaminantes. Algunas versiones más económicas

emplean láminas flexibles de silicona conductora , que sustituyen la cubierta protectora y el diafragma dorado. En algunos casos se emplea una base serigrafiada de tinta conductora como sustrato.

### 2.3 PULSADORES MECANICOS DE BOVEDILLAS

Estos pulsadores (fig.8.3) emplean unos discos metálicos embutidos en forma de casquete esférico , que cuando son deprimidos pasan a una condición invertida , establece el contacto requerido.

Emiten un clásico chasquido audible que advierte de su correcta operación.

### 2.4 PULSADORES REED

Están formados por contactos inmersos en una atmósfera inerte , sellados por una cápsula de vidrio hermética (fig.8.4).

El movimiento de la tecla desplaza un pequeño imán permanente que provoca el cierre de los contactos; dada la acción indirecta sobre estos últimos , no se transmiten sobrecargas mecánicas que provoquen fatiga y desgastes prematuros. Por lo anterior y por la hermeticidad del encapsulado que impide la contaminación de los contactos , este tipo de pulsadores ofrecen un vida útil unas 5 veces superior al clásico pulsador mecánico.

## 2.5 PULSADORES CAPACITIVOS

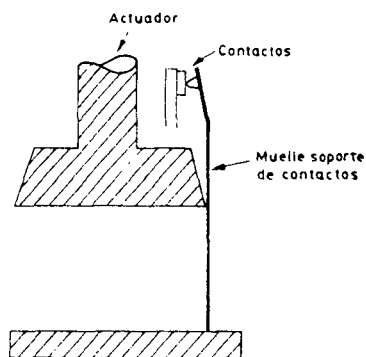


Figura 8.2 Pulsador de lámina flexible.

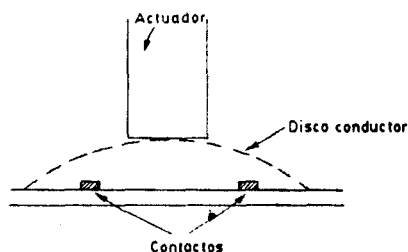


Figura 8.3 Pulsador de bovedilla.

Este tipo de pulsadores emplea un cambio en la capacidad de un condensador para entregar una salida (fig.8.5).

Emplea dos superficies vecinas sobre un mismo circuito impreso, estando una de ellas excitada por la señal alterna de un oscilador; si se aproxima paralelamente una placa conductora sobre ambas superficies, se provoca un acoplamiento entre ellas, con lo que aparece una fracción de la señal alterna en

la salida.

La señal de salida del pulsador debe ser convenientemente amplificada y convertida a niveles lógicos.

Existen múltiples variantes tales como los que utilizan contactos sensitivos, sin ningún elemento móvil. Otros diseños emplean bovedillas metálicas concavas como elemento de acoplamiento.

Los pulsadores capacitativos ofrecen la elevada fiabilidad de los interruptores sin contactos móviles.

Dados los bajos niveles de señal entregada por estos pulsadores se presenta una acusada sensibilidad a interferencias y unos serios condicionamientos en la estructura metálica soporte, trazado de pistas en el circuito impreso y electrónica de amplificación, detección y conversión. Por todo ello sólo aparecen disponibles formando parte de teclados completos producidos por fabricantes especializados.

## 2.6 PULSADORES DE NUCLEOS MAGNETICOS

El elemento conmutador es un núcleo toroidal de ferrita empleado como transformador (fig.8.6).

El núcleo es atravesado normalmente por dos hilos: uno, energizado a alta frecuencia se emplea como primario; el segundo como normal del pulsador, donde el núcleo está saturado por el campo del imán

permanente. La depresión de la tecla desplaza el imán, desaturando el núcleo y generando una señal. Esta última es amplificada, por medio de un núcleo magnético y trasladada a niveles lógicos.

Su fiabilidad es comparable a los pulsadores

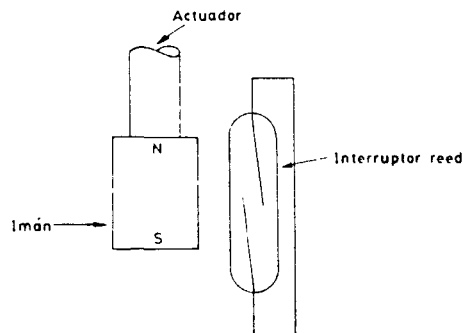


Figura 8.4 Pulsador de láminas (reed).

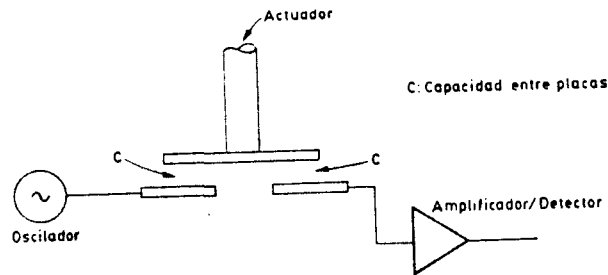


Figura 8.5 Pulsador capacitivo.

capacitivos, presentando, del mismo modo que aquellos, una elevada criticidad en el trazado de las pistas del circuito impreso, por lo que generalmente se ofrece como parte integrante de un conjunto completo.

## 2.7 PULSADORES DE EFECTO HALL

Los sensores de efecto Hall (fig.8.7) están formados por una pastilla semiconductor recorrida por una corriente continua , y un campo magnético perpendicular a ella que provoca una deformación de las líneas equipotenciales sobre la superficie del semiconductor , apareciendo una tensión de salida proporcional al producto de la corriente de polarización por la intensidad del campo magnético aplicado.

La conmutación se obtiene al aproximar un imán permanente al sensor , que desarrolla una tensión de salida que es amplificada y convertida en digital.

Generalmente el conjunto formado por el sensor , amplificador , disparador de Schmitt , monoestable opcional y paso de salida , forma un circuito integrado monolítico (fig.8.8) asociado a cada pulsador.

Se distinguen dos tipos fundamentales :estático y dinámico. Los pulsadores estáticos conducen a su salida , mientras exista campo magnético a su entrada . Los pulsadores dinámicos conducen durante cierto período (típicamente 20 Ms) cuando el campo de entrada supera el nivel de conmutación , pero no lo hacen durante el resto del tiempo que dicho campo permanezca a nivel elevado , ni durante el alejamiento del imán ; para ello incorporan un monoestable que dispara en el flanco de subida de la señal magnética.



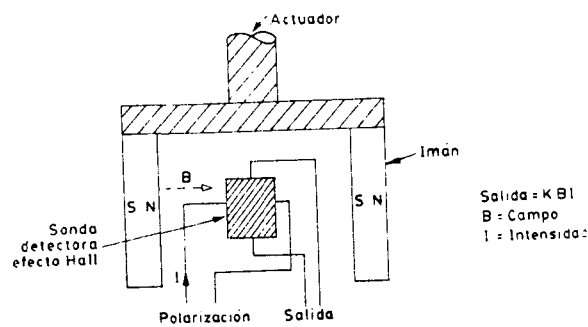
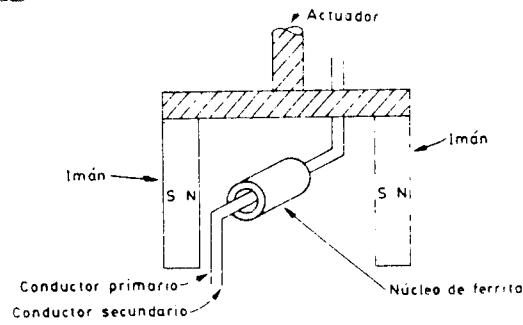


Figura 8.7 Pulsador de efecto Hall.

La salida puede estar formada por un transistor de colector abierto, simple o doble, o bien por una puerta lógica "Y" aceptando señales externas de validación y sincronismo.

Dada la ausencia de contactos, la baja impedancia de todas las señales de interconexión y la sensibilidad a polvo, suciedad y contaminantes, este tipo de pulsadores ofrece la mayor fiabilidad (esencialmente duración infinita), sólo limita por el desgaste del elemento móvil y el resorte de retorno. Este último, en algunos casos, es sustituido por un sistema magnético de retorno, que proporciona simultáneamente una

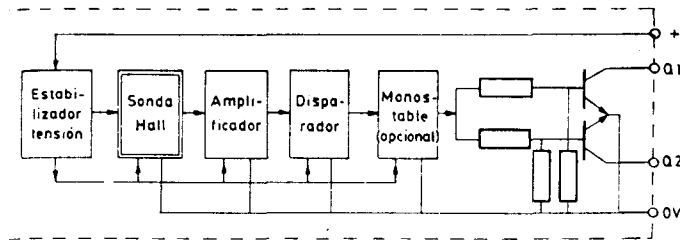


Figura 8.8 Esquema de bloques del detector Hall.

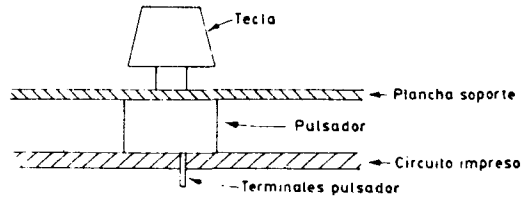


Figura 8.9 Conjunto pulsador más tecla.

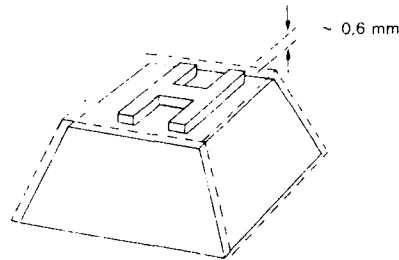


Figura 8.10 Primera fase de una tecla de doble inyección.

realimentación al tácto.

### 3. TECLAS

El elemento unitario componente de un teclado, es el formado por el subconjunto: pulsador más tecla, que a menudo es denominado simplemente tecla (fig.8.9).

La tecla o capuchón, aún cuando no cumple ninguna función eléctrica, es absolutamente imprescindible con el fin de identificar adecuadamente cada pulsador, así como ofrecer una correcta superficie de actuación apropiada a las características fisiológicas de manos y dedos del operador.

### 3.1 ROTULACION

Con el fin de obtener una correcta identificación de cada uno de los distintos pulsadores que componen un teclado es preciso rotular con los gráficos apropiados cada una de las teclas .

El grafismo debe facilitar una rápida localización de la tecla idónea por lo que debe ser claro y esquemático. Los grafismos correspondientes a las teclas asociadas a caracteres alfanuméricos no presentan evidentemente ningún problema; en cambio , las diversas funciones de control han exigido el desarrollo de una serie de grafismos y abreviaturas un tanto criptográficas para el no iniciado.

Para fijar sobre la tecla su correspondiente grafismo se utilizan distintas técnicas , siendo las más usuales: serigrafía , pantografiado , doble inyección y empleo de teclas transparentes rotulables.

Los métodos serigráficos y pantográficos son similares a los empleados para marcado de paneles metálicos , circuito impresos , etc , por lo que no procede mayor ampliación .

Las teclas rotuladas por doble inyección (fig.8.10) están construidas en dos etapas. En primer lugar se inyecta una pieza con unas protuberancias en forma de cordón que sigue el perfil del grafismo deseado . A

continuación esta primera pieza es

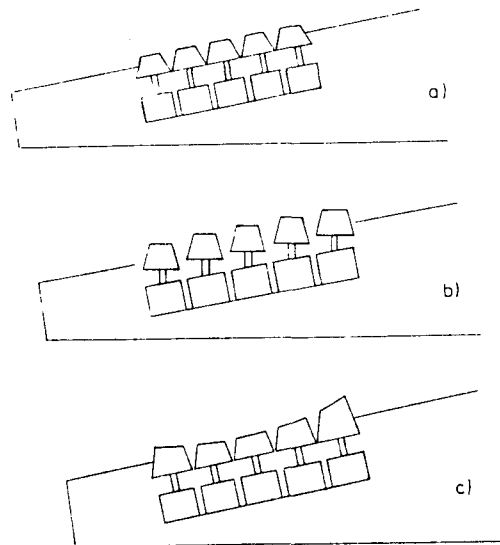


Figura 8.11 Perfiles de teclados: a) inclinado; b) escalonado; c) esculpido.

utilizada como inserto en un segundo molde con el perfil externo de la tecla . Este segundo molde recibe una inyección de plástico con distinto colorante del inserto , por lo que aparece finalmente como un afloramiento de la cresta de la primera inyección , definiendo por tanto , por lo medio de un color distinto , el rótulo deseado .

Las rotulaciones serigraficas y pantografiadas adolecen del defecto de ser sumamente sensible a la erosión causada por el roce mecánico con los dedos del operador y de los agentes químicos , entre los que deben contarse como más importantes el sudor humano y los líquidos limpiadores . Esta sensibilidad queda

totalmente compensada mediante el empleo de doble inyección , por cuanto el plástico utilizado es insoluble a los agentes químicos ambientales y la eventual erosión mecánica provoca un desgaste paralelo tanto en la primera inyección (grafismo) como en la segunda (cuerpo) por lo que mientras esta erosión no desgaste más allá de medio milímetro de plástico , tecla aparecerá como perfectamente legible.

Las ventajas mencionadas de las teclas doblemente inyectadas tienen su contrapartida bajo el punto de vista económico , de tal modo que el costo de los utillajes sólo puede ser amortizado bajo unas condiciones de cantidad y exigencias de calidad , que no siempre se cumplen ;para solventar estas situaciones (fundamentalmente las series reducidas o prototipos ) se han desarrollado unas teclas especiales con las cubiertas total o parcialmente transparente . Esta ventana (o la cubierta completa , según sea el caso) es desmontable , permite la inserción de leyendas rotuladas o impresas sobre papel en forma de etiquetas , facilitando de forma excepcional la obtención de teclados especiales en un tiempo mínimo.

Como complemento a los sistemas de rotulación de teclas deben considerarse otros sistemas de identificación , como el empleo de plástico de distintos colores para diferenciar teclas de especial

interiores; teclas luminosas mediante diodos LED mono o bicolor o bien mediante lamparitas que iluminan por transparencia rótulos translucidos; distintas dimensiones de la usual cuadrada de 19mm (3/4) , que se denomina usualmente de una unidad , existiendo comúnmente teclas de 1 , 1 1/4 , 1 1/2 , 1 3/4 , 2 , 3 , 6 , 8 y 9 unidades de longitud , así como figuras distintas del cuadrilátero , siendo relativamente normales formas de "L"

### 3.2 PERFIL

El conjunto formado por la superficie superior de cada una de las teclas puede adoptar distintas configuraciones basadas en las preferencias y aplicaciones deseadas.

Los teclados suelen montarse formando un ángulo de 10° a 13° con la horizontal.

La interacción entre orientación de teclas e inclinación general del teclado dará lugar a las configuraciones antes indicadas , y que pueden resumirse en tres tipos fundamentales (fig.8.11).

- Inclinado ("Sloped")
  
- Escalonado ("Stepped")
  
- Esculpido ("Sculptured")

El teclado inclinado (fig.8.11a) está formada por un conjunto de pulsadores y teclas totalmente verticales en origen , que posteriormente se giran , adoptando la inclinación general del teclado , con lo que las superficies superiores de las teclas son coplanarias (salvando la ligera concavidad asociada a cada tecla). El teclado escalonado (fig.8.11b) está formado por un conjunto de pulsadores y teclas no alineados originalmente , sino formando un ángulo entre  $10^{\circ}$  a  $20^{\circ}$  (sea por cambio de orientación del vástago del pulsador , sea por modificación del perfil de la tecla). Al girar el conjunto , dándose la inclinación general , las superficies superiores de las teclas quedan horizontales , formando filas a distintas alturas , o escalones.

El teclado esculpido (fig.8.11c) presenta una cierta similitud con el escalonado , con la diferencia de que los ángulos iniciales de pulsadores y teclas no son comunes a la totalidad de elementos , sino función de la fila donde están situados , de tal modo que las filas superiores forman ángulos más acentuados que el general del teclado , mientras que las inferiores no sólo lo desacentúan , sino que llegan a adoptar pendiente negativa. El conjunto adopta con todo ello un perfil curvado.

Los teclados inclinados y escalonados , según

diversos estudios , no aportan sustanciales beneficios en cuanto a velocidad , seguridad o comodidad del operador , por lo que se adoptan indistintamente : la costumbre asocia los teclados inclinados a las maquinas de escribir , por lo que según la aplicación tienda más hacia uno u otro sentido . se adopta la solución más próxima .

El teclado esculpido está justificado por razones económicas , de tal modo que las distancias entre las llemas de los dedos y las superficies de apoyo de las teclas se uniformizan , presentando asimismo cada fila de teclas el ángulo de ataque más apropiado. Con todo ello se mejoran velocidad , fiabilidad confort del operador , dando por tanto mejores rendimientos en aplicaciones de trabajo pesado. Se popularizó inicialmente a partir de los equipos perforadores de tarjetas , y se ha extendido su uso en todos los equipos perforadores de tarjetas , y se ha extendido su uso en todos los ámbitos de la informática de alto nivel.

## VISUALIZADORES TRC

### 1. IDEAS SOBRE EL VISUALIZADOR TRC

En un sistema microcomputador uno de los elementos más costosos es el terminal utilizado por el operador



del sistema , ya que el caso de utilizar un terminal teleimpresor tal como un teletipo , el precio de este puede ser del orden de magnitud del precio del sistema.

Una forma de solventar este problema consiste en utilizar como terminal un teclado y un visualizador de tubo de rayos catódicos (TRC) , pudiendo ser este el TRC del receptor de televisión del propio usuario.

Además , la reciente introducción de circuitos integrados LSI controladores de TRC hace que el diseño de un terminal de este tipo sea sencillo y su coste bajo.

Ahora describiremos el funcionamiento de una unidad controladora de TRC , las posibles arquitecturas de diseño que se pueden utilizar , y los circuitos integrados controladores que existen actualmente.

Además se describe el funcionamiento de un visualizador TRC para posteriormente describir las principales funciones que debe realizar como terminal visualizador de datos de un sistema.

### 1.1 FUNCIONAMIENTO DEL TRC

Tal como se ha mencionados antes , como visualizador TRC puede utilizarse el de un receptor de televisión comercial , cuyo diagrama de bloques aparece en la figura 9.1.

En la recepción normal de un programa de TV la señal

captada por la antena pasa al sintonizador y después al amplificador de frecuencia intermedia, a la salida de este detector es la señal de video; la señal de salida de este detector es la señal de video, con una amplitud aproximada de 2 voltios pico a pico. Esta señal es filtrada para eliminar la frecuencia portadora de video y después es introducida en el amplificador de video; dentro de ese amplificador la señal sigue tres caminos. Por el primero de ellos llega la señal de video al TRC después de haber sido eliminada la frecuencia portadora de sonido y de haber sido amplificada; por el segundo, parte de la señal llega a la etapa de sonido y al altavoz, y por el tercero, los impulsos de sincronismo son separados de la señal de video y sirven para sincronizar la imagen en la pantalla.

En el caso de utilizar un televisor como terminal de datos de un sistema microcomputador hay dos puntos por los que se puede introducir la señal proporcionada por el sistema al televisor: en la entrada de la antena, con una señal modulada convenientemente, o directamente a la salida del detector de video, bloqueando las etapas previas del televisor.

El primer sistema tiene la ventaja de que no debe alterarse para nada el receptor de televisión, pero tiene los inconvenientes de que debe modularse previamente la señal portadora de la misma frecuencia

que la de algún canal de televisión, y de que debido al poco ancho de banda de los sintonizadores de radiofrecuencia y del amplificador de frecuencia intermedia (3.5 Mhz aproximadamente), queda limitado el número de caracteres que pueden representarse en la pantalla.

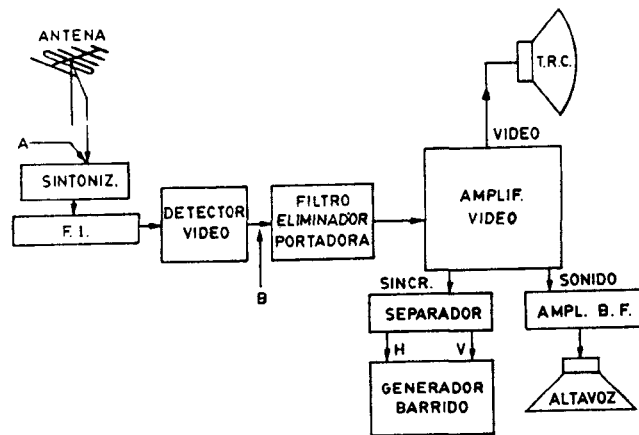


Figura 9.1 Diagrama de bloques de un receptor de TV.

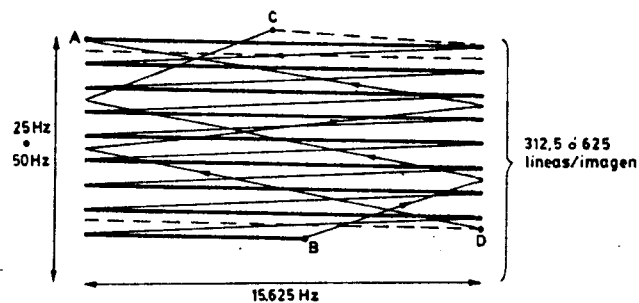


Figura 9.2 Barrido de la pantalla de un receptor de TV.

Si la señal se introduce a la salida del detector de video se resuelven los dos problemas anteriores, pero entonces deberá modificarse algo el televisor ya que

normalmente no lleva un conector externo para poder introducir una señal de video.

Es asimismo aconsejable desconectar las etapas de alta frecuencia del televisor.

En el caso de utilizar como terminal un monitor digital de televisión, la solución a adoptar es obviamente la segunda al no disponer de las etapas de alta frecuencia. Estos monitores tienen la ventaja de que han sido diseñados especialmente para la visualización de caracteres alfanuméricos y el número de caracteres que se pueden visualizar con ellos es el doble o más que los de un televisor comercial.

## 1.2 SEÑALES DE CONTROL DE UN VISUALIZADOR TRC

En un visualizador TRC dos generadores de rampa controlan el haz de electrones que inciden sobre la pantalla, modulado en intensidad por la señal video.

La primera rampa controla el haz horizontalmente. Su período total es de 64 Ms en el sistema europeo de TV (al que nos referimos en lo sucesivo), del que aproximadamente la cuarta parte del tiempo se emplea para el retorno del haz.

La segunda rampa controla el haz verticalmente. Su período es de 20 ms, con lo que se definen 312 líneas horizontales en la pantalla (fig.9.2).

Este tipo de barrido es el llamado directo y es el

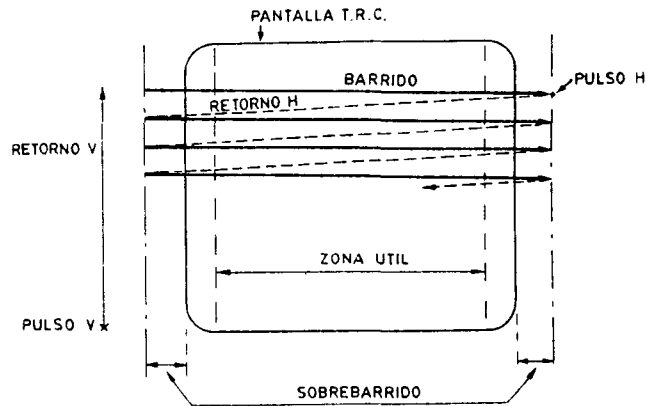


Figura 9.3 Forma de utilización de la pantalla de un visualizador TRC

que suele utilizarse en los monitores de TV comercial es el de entrelazados, en el que cada imagen se representa mediante dos cuadros consecutivos de 312 líneas y media cada uno, entrelazando las líneas del segundo cuadro con las del primero, lo que nos da un total de 625 líneas por imagen. En lo sucesivo solamente nos referiremos al sistema de barrido directo debido a que su resolución es suficiente y a que las señales de control se simplifican notablemente.

Las señales de control de un TRC son tres:

Sincronismo horizontal o señal H. Señal de impulsos cuyos flancos de subida provocan el retorno de derecha a izquierda de la pantalla del haz de electrones. La amplitud mínima de estos impulsos depende del monitor de TV empleado; la máxima, no es crítica. Durante el tiempo de retorno del haz éste se inhibe. Normalmente el barrido horizontal de la pantalla es mayor que el tamaño físico de la misma, y la longitud de una línea de caracteres es menor (fig.9.3). El periodo del barrido horizontal es de 64 Ms y la tolerancia es un 10% como máximo.

Sincronismo vertical o señal V. Señal de impulsos cuyos flancos de subida provocan el retorno del haz de abajo a arriba de la pantalla. El haz de electrones también se inhibe durante este tiempo. Debe estar sincronizado con la señal H, y su periodo es de 20ms.

Modulación de la intensidad del haz o señal Z. Es una señal normalmente binaria en el caso de visualizadores alfanuméricos y controla la intensidad del haz electrónico. Supondremos que su estado lógico bajo un intensidad mínima o nula.

La figura 9.4 nos muestra un ejemplo simplificado de las tres señales mencionadas anteriormente en el caso de representación de caracteres alfanuméricos. Si empleamos como monitor un receptor de TV e introducimos la señal después del detector de video, la señal adecuada es la

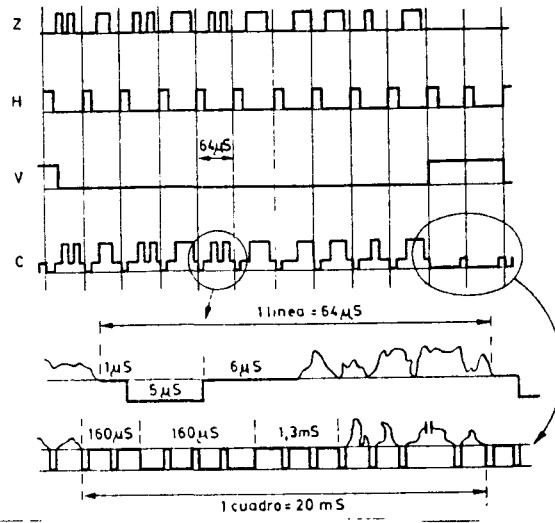


Figura 9.4 Señales de control de un visualizador TRC

<i>N.º líneas de caract.</i>	<i>N.º caract. por línea</i>	<i>N.º total caracteres</i>	<i>N.º líneas TV usadas</i>	<i>Ancho de banda mínimo</i>
8	16	128	96	0.7 MHz
16	32	512	192	2.8 MHz
16	64	1024	192	5.6 MHz
24	80	1920	288	10.4 MHz

Tabla 9.1

señal C de la figura 9.4, que se obtiene sumando adecuadamente las señales H, V y Z. Si la señal la introducimos por el conector de la antena, la señal C debe modularse mediante una portadora cuya frecuencia sea la de algún canal comercial de TV.

### 1.3 FUNCIONES A REALIZAR POR EL VISUALIZADOR TRC

En el caso de utilizar un visualizador TRC como terminal de un sistema microcomputador, las funciones que debe realizar son las siguientes:

Representación de un texto alfanumérico. La finalidad principal de un visualizador TRC consiste en obtener una

representación en la pantalla de un texto que tenemos almacenado en codificación binaria en una memoria digital.

La placa controladora del visualizador TRC deberá tener por tanto un generador de las dos señales de sincronismo H y V y un circuito controlado por dicho generador capaz de leer el texto almacenado en la memoria y transformarlo en la señal Z, que es la moduladora del haz electrónico.

Si el sincronismo vertical tiene un período de 20 ms deberemos leer cada 20 ms el texto de la memoria y representarlo en la pantalla, con lo que se conseguirá una imagen estable.

Formateado de la pantalla (fig.9.5). Un aspecto muy importante a tener en cuenta en el diseño es el formato que tendrá el texto en la pantalla: en qué zona de la pantalla habrá imagen, cuántas líneas de texto contendrá la pantalla, cuántos caracteres contendrá cada línea, cuántos puntos tendrá cada carácter, y cuántos diferentes podrán representarse.

Para que un visualizador TRC sea utilizable como terminal alfanumérico debe cumplir una serie de condiciones mínimas tales como las siguientes:

- a) Representar 8 líneas de 16 caracteres cada una.
- b) Utilizar formato de caracteres de 5\*7.



c) Poder representar 64 tipos de caracteres diferentes.

d) El texto representado ira desfilando de abajo a arriba a medida de que se produzca nueva información para visualizar.

El formato mínimo de 8 líneas de 16 caracteres cada una puede corresponder a un visualizador de 12 pulgadas. Para visualizadores de grandes dimensiones (26 pulgadas , por ejemplo) puede llegarse a un formato de hasta 40 líneas de 34 caracteres cada una.

Formatos normalizados usados son los de la tabla 9.1 , en la que además se indica el ancho de banda mínimo necesario para el amplificador de video. El barrido se supone del tipo directo , sin entrelazado. Todos estos formatos pueden adaptarse a la norma europea de TV de 625 líneas , de las que solamente son usadas 312.

La diferencia entre el número de líneas TV usadas y el número total de líneas de una imagen (312) corresponde a líneas que no se utilizan para la visualización y que están situados en las partes superiores e inferior de la imagen.

Funciones especiales. Si utilizamos como memoria del texto a visualizar parte modificar el texto que queremos visualizar basta modificar el contenido de la memoria bajo supervisión del mismo microcomputador , la forma normal de funcionamiento es enviar carácter tras

carácter al controlador del visualizador para que éste los incluya en su memoria local si los caracteres son representables, o realice ciertas funciones especiales si el carácter es una orden de control.

Ciertas funciones especiales son consecuencia inmediata de la llegada de un carácter de control, tal como el retorno de línea. Otras funciones especiales deben realizarse automáticamente como resultado del estado actual del sistema, tales como el desplazamiento del cursor, el no incluir más caracteres en una línea cuando ésta está llena, etc.

Todas las funciones pueden realizarse mediante hardware especializado o mediante la gestión de la pantalla por un microprocesador dedicado especialmente a ello.

Como funciones especiales principales merecen citarse las siguientes:

- Retorno de línea.
- Salto de línea.
- Tabulación vertical.
- Tabulación horizontal.
- Borrado de un carácter.
- Borrado de todo un texto.
- Desplazamiento del cursor.
- Control de fondo.
- Escala de grises.

- Posibilidad de representación gráfica.
- Posibilidad de utilización de un "lápiz luminoso".

En los ejemplos de realizaciones prácticas que se dan en otro apartado se indicará cómo realizar algunas de estas funciones.

Si el visualizador es autónomo e independiente del sistema microcomputador deberá estar adaptado a las normas de transmisión de datos serie RS232 o bucle de corriente de 20 mA , o ser compatible con la lógica TTL en caso de transmisión de datos en paralelo.

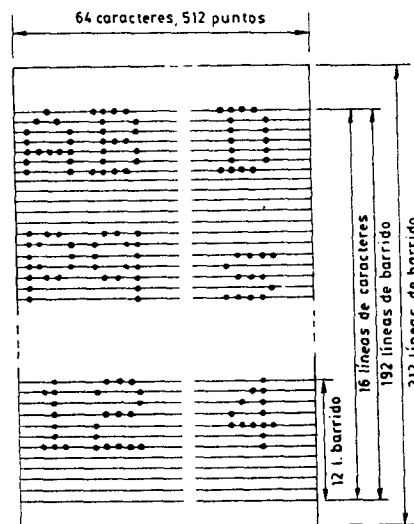


Figura 9.5 Un ejemplo de formateo de la pantalla. Formato: 16 líneas de caracteres: 64 caracteres por línea; 12 barridos/línea caracteres; Matriz 5 x 7, con 8 puntos/carácter; se precisan 512 puntos/línea para representar los 64 caracteres de una línea; se precisan 192 líneas de barrido horizontal para representar las 16 líneas de caracteres; si los 512 puntos se representan en 4/5 línea (51,2  $\mu$ s) la frecuencia para los puntos es de 10 MHz precisándose una Bw del amplificador de video de 5 MHz.

## CASSETTES

### 1. INTRODUCCION

Una cassette es una caja de plástico en cuyo interior se encuentra bobinada una cinta magnética en la que se puede registrar información de principio a fin de una forma secuencial. La señal eléctrica a registrar en la cinta debe tener una frecuencia comprendida dentro de ciertos límites, que dependen de la calidad de los amplificadores del sistema registrador/reproductor, y del soporte magnético de la cinta. Estas consideraciones permiten distinguir tres tipos de transportes (conjunto electromecánico que permite registrar y posteriormente reproducir una señal eléctrica):

- 1) Audio o analógico sencillos (50 Hz-5 KHz)
- 2) Audio o analógico profesionales (20 Hz-15 KHz)
- 3) Digitales (20 Hz-30 KHz)

Estos transportes, además de diferir en la anchura de banda de frecuencia que admiten, difieren en la estabilidad de la velocidad de la cinta y en la distorsión final de la señal.

Aunque hemos definido uno de los tipos como digital, ello no implica que sea el único capaz de almacenar información digital. De hecho, dado que su precio se aproxima mucho al de los transportes para minidiscomos

flexibles , los transportes digitales apenas se usan. Además , el procedimiento de digitales es el mismo que para los discos flexibles. Por todo ello , en este capítulo nos centramos en el diseño de un interfaz para un transporte de cassette analógico.

Dado el reducido ancho de banda y la poca estabilidad del arrastre de la cinta en los transportes analógicos , no se puede pretender registrar en ellos la información en forma digital , mediante modulación de impulsos , como sucede en los transportes digitales ( cintas o discos magnéticos ). En los transportes analógicos la señal binaria a registrar se debe codificar empleando modulación de amplitud o de frecuencia. Problemas de ajuste de volumen han hecho descartar los sistemas modulados en amplitud y todos los procedimientos empleados , "normalizados " a nivel internacional , emplean modulación de frecuencia. En el apartado 10.4 (formato de grabación ) se describe los procedimientos más frecuentes empleados.

Este capítulo se divide en dos partes. En la primera se hace descripción general del método de diseño empleado. En la segunda se describe la realización del sistema de interconexión en tres versiones que van desde un sistema con el mínimo de circuitería y máximo programa , hasta un sistema realizado casi completamente con circuitos y un programa mínimo. El

método de diseño descrito está orientado tanto a la realización de la parte programada (software) como de la parte física (hardware) del sistema. Básicamente consiste en describir detalladamente las operaciones a realizar por el sistema, empleando un pseudolenguaje fácilmente traducible a ensamblador y/o a lenguaje de transferencia de registros (RTL).

De esta forma la primera fase del diseño, en la que se concreta el algoritmo que debe ejecutar el sistema, sirve tanto si se va a realizar con un programa como si se va a realizar con un circuito. La transformación de una parte del sistema de programa a circuito sólo afecta a la última fase del diseño y consiste en una simple traducción de la descripción obtenida en la primera fase, que puede ser más o menos automática según los casos.

## DISCOS FLEXIBLES

### 1. INTRODUCCION

Los discos flexibles constituyen uno de los periféricos para almacenar información más interesante de los sistemas microcomputadores . Características que han influido notablemente en su aceptación y utilización han sido entre otras las de bajo precio , mayor fiabilidad que las cintas magnéticas y un tiempo de acceso de información relativamente pequeño comparado con las velocidades de tratamiento de esa información por los microprocesadores . Los primeros discos flexibles utilizados fueron producidos en el mercado por IBM en el año 1970 , es decir , antes de que aparecieran los microprocesadores . El modelo se denominó 23FD y contenía un disco flexible de 8 pulgadas con una sola cara y con código de grabación en simple densidad . Posteriormente , en 1975 , Shugart usó un código de doble densidad M2fm en su modelo SA800 con lo que duplicó la cantidad de información almacenada. En el año 1976 , la misma Shugart introdujo un disco flexible de 5 1/4 pulgadas que recibió el nombre de minidisco.

El capítulo está dividido en siete apartados . En el primero tratamos de las características de los discos , capacidades de almacenamiento y transportes físico de los discos . En el siguiente descubrimos los códigos

empleados para la grabación. En el tercero exponemos diferentes formatos de pista. En el cuarto tratamos brevemente de las funciones que deben ser implementadas entre el sistema microcomputador y el transporte de discos. En el siguiente describimos las características generales de algunos controladores integrados. Los dos últimos apartados describen la realización de dos interfaces concretas: la primera para simple densidad y la segunda para doble densidad.

## 2. CARACTERISTICAS DE LOS DISCOS

Los discos están constituidos por dos capas, una de ellas hace de sustrato y la otra, constituida a base de un óxido magnético de aproximadamente 7 Mm de espesor, que es donde se graba la información. Ambas capas están unidas y cerradas en un sobre de plástico. Como hemos dicho anteriormente, existen dos tamaños de discos estandarizados de 8 a 5 1/4 pulgadas.

En la figura 11.1 se muestra la forma de los discos así como los valores de algunas dimensiones.

Los discos flexibles se distinguen entre sí por el tamaño ( 8 y 5 1/4 pulgadas ), el número de caras utilizables ( 1 o 2 ), el código de grabación ( simple o de doble densidad ) y el formato de grabación ( sectorizados por hardware o por software ).

En los discos sectorizados por software existe un



único agujero en el disco que al pasar por el agujero correspondiente , realizado en el sobre de plástico , es detectado por la circuitería de forma que define el principio de cada pista . Los discos sectorizados por hardware poseen un agujero físico por cada sector y dos agujeros próximos indicativos del principio de pista . El paso de dichos agujeros es detectado , normalmente , por el conjunto fotodiodo-fototransistor.

Existe sobre el disco una marca que se hace servir para protegerlo contra escritura ; los discos de 8 pulgadas quedan protegidos al dejar la marca visible y con los minidisos ocurre lo contrario.

## 2.1 CAPACIDAD DE ALMACENAMIENTO

La capacidad de almacenamiento de los discos depende del tamaño , código de grabación , formato de grabación y si está sectorizado por hardware o por software . En la tabla 11.1 se resume algunos valores típicos.

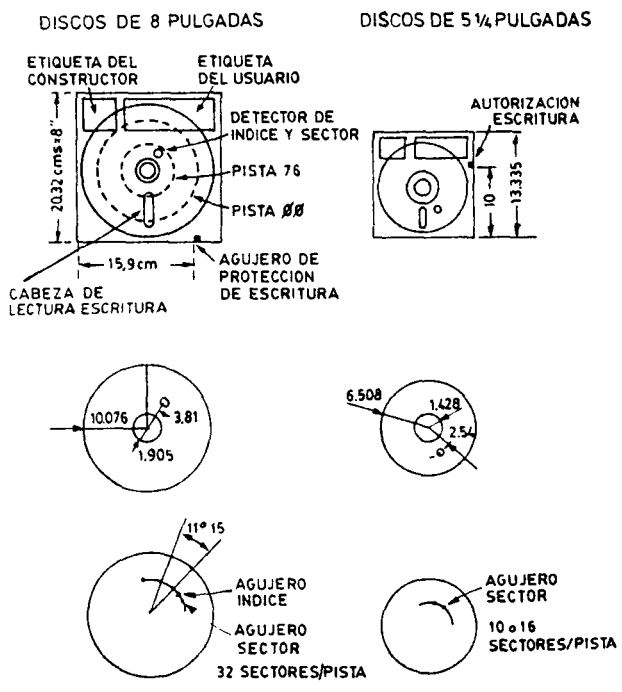


Figura 11.1 Dimensionado de los discos.

## IMPRESORAS

### 1. LAS CARACTERISTICAS DE LAS IMPRESORAS

Las impresoras son uno de los dispositivos usados para preparar un registro permanente del ordenador de salida; la otra es el plotter gráfico. Es necesario hacer una distinción entre el trabajo de la impresora y el plotter como existen ahora. Aunque tradicionalmente las impresoras tienen imprimidas letras y números mientras que los plotters tienen que trazar líneas y curvas para formar gráficos de líneas, dibujos o imágenes, usando X, \*, -, y otros tipos de símbolos para formar líneas y coordenadas.

### 2. IMPRESORAS VERSION TERMINAL Y TELEIMPRESORAS

Un terminal de impresión consiste de una impresora con otros dispositivos periféricos, normalmente para acceder al ordenador central. Por ejemplo, los teletipos son terminales de impresión con interfases de comunicación de líneas telefónicas.

### 3. CLASIFICACION DE LAS IMPRESORAS

Las impresoras usadas en mini y microordenadores se clasifican en tres amplias categorías:

### 3.1 LAS IMPRESORAS DE IMPACTO.

Las impresoras de impacto forman caracteres en papel en el tradicional camino para el choque del papel con la cabeza de la impresora y presionando una cinta de tinta entre la cabeza y el papel. Las impresoras de no impacto forman caracteres sin enganchar el mecanismo de impresión con la superficie mediante la sensibilidad del calor del papel o por la pulverización de la cinta desde un surtidor.

### 3.2 VERSION DE LA MATRIZ DE PUNTOS

Los caracteres formados están hechos por una máquina de escribir estándar. Todos los caracteres están fijados en el reverso de las barras de la máquina de escribir. Cuando se imprime todos los elementos aparecen conectados. La matriz de puntos está formada por puntos que forman un grupo de letras y números cuando se unen.

### 3.3 VERSION DE CARACTERES DE LINEA DE TIEMPO.

Los caracteres impresos de tiempo imprime cada carácter instantáneamente. Las impresoras de línea de tiempo imprime cada línea instantáneamente.

La figura 3.1 muestra la clasificación de impresoras sin estar mayores categorías.

## **4. CARACTERISTICAS DE LAS IMPRESORAS**

La mayoría de las impresoras usadas con mini y microordenadores usan el código ASCII.

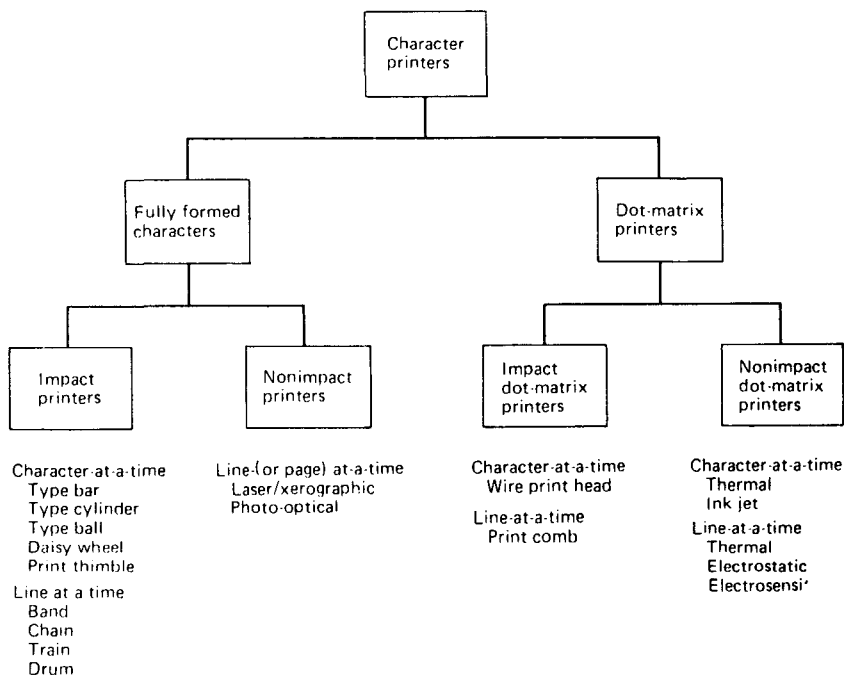
Las impresoras están especificadas según usen 48 , 64 , 96 , 128 caracteres.

Las impresoras de 48 y 64 incluyen unos símbolo , números y espacios especiales así como el alfabeto Inglés. Las impresoras de 96 caracteres incluyen minúsculas y ciertos caracteres especiales del alfabetos inglés y algunos símbolos especiales de suma. De los 96 caracteres el espacio y el borrado no se incluyen , por tanto quedan 94 caracteres. Las impresoras de 128 caracteres contienen 32 caracteres usados para la comunicación y control. Estos caracteres normalmente no se imprimen , y cuando una impresora lleva especificados los caracteres de 128 , normalmente se refiere a su función de expresión de comunicación o a su terminal de control. Una aplicación especial , son los 32 códigos adicionales que pueden imprimirse usando símbolos de otro lenguaje como el inglés , donde los caracteres del código ASCII en inglés no son aplicables.

## 5. LOS CARACTERES DE LAS IMPRESORAS DE TIEMPO DE IMPACTO

Las clásicas máquinas de escribir es un ejemplo de estas impresoras. Normalmente las máquinas de escribir

, sin embargo , no prestan atención ellas mismas al uso del ordenador porque ellas carecen de un código para facilitar la comunicación y son relativamente lentos.



**FIGURE 3-1** A classification of printers.

Sin embargo , algunas máquinas de escribir ordinarias han sido usadas con ordenador mediante el montage de solenoide sobre las llaves y la interconexión de solenoides con circuitos especiales directamente al ordenador.

La impresora clásica usada con mini y microordenadores fue la impresora de teletipo modelo 33 , originalmente diseñada para imprimir la comunicación

de las líneas de telefónica para las nuevas transmisiones(fig.3-2).

La habilidad y el bajo coste de estas impresoras así como su relativa facilidad de interconexión hacen de ellas que sean comunes para el uso de pequeños ordenadores. El modelo 33 imprime en un rango de 10 caracteres por segundo , lo cual es lenta , mientras que las impresoras de hoy tienen una velocidad por encima de 55 caracteres por segundo.

El mecanismo de impresión para el modelo 33 de la maquina de escribir es un cilindro vertical(fig.3-3). Los caracteres del código ASCII enviados a la impresora y trasladado a un circulo en movimiento así la columna contiene los caracteres diseñados en el papel y levanta o baja el cilindro para indicar que la columna contiene los caracteres diseñados en el papel. Un mecanismo de impresión lleva el cilindro hacia el papel , donde un sólo carácter es imprimido , creando el caracter en el papel.

La máquina de escribir de modelo 33 esta conectada con pequeños ordenadores por una corriente de 20 o 60 miliamperios usada para transmitir los bits en serie del código ASCII. Muchos ordenadores pequeños tienen una entrada y salida por puertos de 20 o 60 miliamperios y una serie de adaptadores para convertir el puerto RS232C en el modelo 33.

Similar al modelo 33 de teletipo esta la máquina de escribir eléctrica, la cual usa una cabeza típica sobre una esfera con caracteres agrupados alrededor de las superficies. Para golpear la cinta y entonces el papel, el código del último carácter transmitido produce que la esfera gire alrededor de un eje vertical y se inclina sobre el horizontal hasta que el carácter deseado este en la cinta(fig.3-4). Como el teletipo la cabeza requiere un movimiento mecánico de dos dimensiones. Los equipos adaptados se aprovechan para convertir la máquina de escribir eléctrica para ser usada en mini-microordenadores.

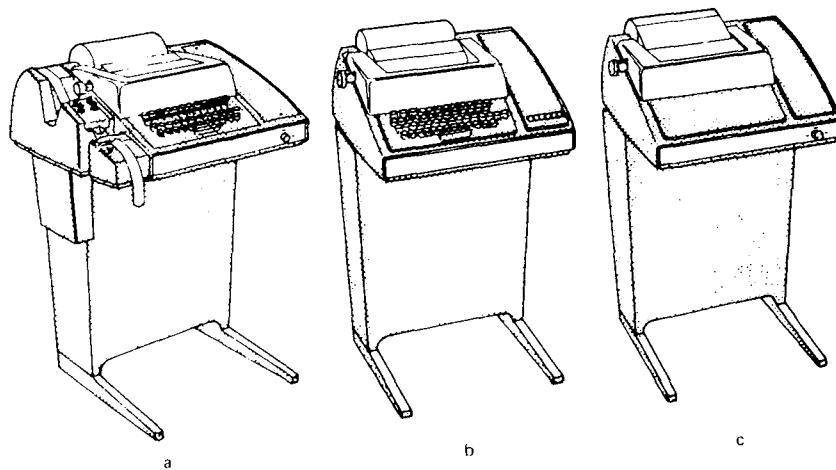
Ambos modelos el teletipo 33 y la máquina de escribir eléctrica esta diseñada para otros procedimientos y adaptados para ser usados como impresoras. Otros tipos de impresoras diseñadas para el uso del ordenador lo tienen montados sobre la cabeza de giro.

Dos versiones de este método son, las impresoras DAISY-WHEEL (figura 3.5).

La cabeza de la impresora DAISY-WHEEL esta montada sobre un disco de rotación con pétalos de flores similares a una margarita. Cada pétalo contiene los caracteres grabados en reverso.

La cabeza de la impresora THIMBLE es similar a la impresora DAISY-WHEEL ecepto que los petalos estan hacia arriba y formada como una canasta. Los caracteres estan





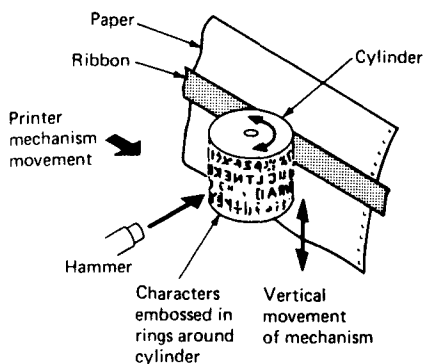
**FIGURE 3-2** Teletype printers were adapted early as computer printers. Many of the original model 33 printers are still used. (a) Automatic send-receive (ASR). (b) Keyboard send-receive (KSR). (c) Read only (RO). (*Teletype Corp.*)

agrupados en flexibles canastas alrededor de los periféricos.

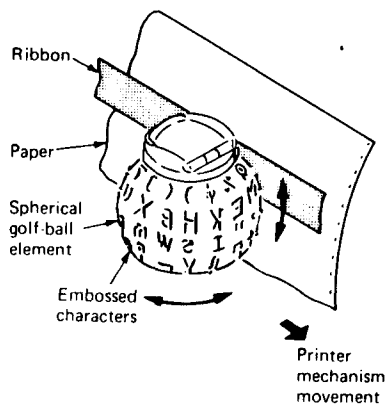
Un mango imprime sobre la cabeza y es similar a un engranaje en forma de margarita excepto que los pétalos son flexibles, son girados hacia arriba y esta configurado en forma de mango o como una cesta. Las letras están montadas en flexibles salidas alrededor de los periféricos de la cesta. Como la cesta gira y se mueve horizontalmente a través del papel, un cursor golpea la letra apropiada así como gira la posición.

Ellos han sido diseñados para el uso del ordenador, porque ambos tipos de impresoras de engranajes de

giro tiene una única característica para el uso de las impresoras. Más importante es su alta velocidad de impresión (alrededor de 50 caracteres por segundo) con esta forma de caracteres ejecuta la máquina de escribir una impresión de más calidad. Estas características hace que las impresoras de engranaje de giro sean usados para la aplicación de procesos.



**FIGURE 3-3** The model 33 Teletype uses a cylinder printer. Character set is embossed around the cylinder, which rotates and shifts up and down to position the character to be printed. The hammer strikes the cylinder to print a character.



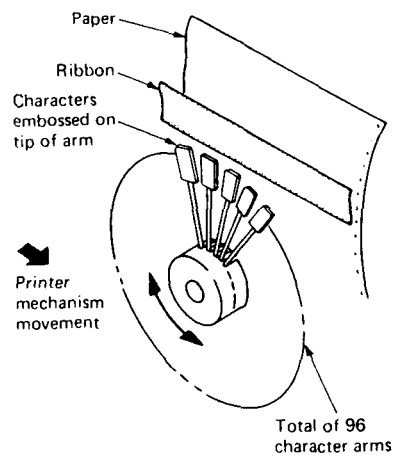
**FIGURE 3-4** The golf ball used on the IBM Selectric typewriter is an updated version of a cylinder printer. The sphere rotates to position a character, and the entire sphere strikes the paper.

Las impresoras de engranaje son de cabeza intercambiable, permitiendo un modelo de espacio proporcional, modelos de caracteres ópticos, modelos multilineales y modelos de aplicación especial

(fig.3-6).

Los tipos de ancho carrocería permite papel de anchura por encima de 15 y algunos modelos tiene dos engranajes simultaneos imprimiendolo sobre la misma línea. (fig.3-7). Los modelos de impresoras de engranaje en forma de margarita se ilustran en la fig.3-8.

Los caracteres imprimidos en el tiempo , como normalmente pensamos es la impresión de izquierda a



**FIGURE 3-5** Characters are embossed on the petals of the daisy wheel. The wheel spins to present each character for printing. A hammer strikes the character petal when it is in position.

derecha . Esta es una impresión unidireccional . Las

impresoras de engranaje de giro tienen la capacidad de impresión bidireccional. La segunda línea de impresión es almacenada en un buffer de memoria sin que el control de la impresora pueda imprimir en cada dirección, intentando tardar el menor tiempo posible.

Si la próxima línea a imprimir es una línea llena la cabeza de la impresora esta ya en la posición extremo derecha, la línea deber ser imprimida de derecha a izquierda regresando a donde empezamos a leer. Esto elimina que la impresora de tiempo sea de lo contrario menor que el retorno de la cabeza impresora a través del ancho de línea imprimida de la posición extrema izquierda. Los circuitos de control en las impresoras evalúan cada línea imprimida y el cual incluso selecciona la dirección imprimida en menor tiempo.

Algunos programas gráficos son ejecutados mediante un espacio fraccionario horizontal de la cabeza impresora, a lo largo del papel bajo un control de programa. La figura 3.9 ilustra un tipo de curva y los subíndices se hicieron posible mediante un espacio controlado por programa. Por ejemplo la separación normal de un carácter horizontal es de 10 caracteres por pulgada (25,4 mm) pero bajo un control de programa la separación de un carácter puede cambiarse en 1/120 y en un incremento de (0,2 mm).

Correspondientemente el papel tiene un espacio

vertical para imprimir seis líneas por pulgada normalmente. Bajo el control de programa, la separación entre líneas es controlable a 1/48 en un incremento de (0, 5mm). Con esta posibilidad la rueda de giro se mueve en pequeños incrementos horizontales similar al movimiento del papel. Se ejecutan pequeños desplazamientos en el eje X Y.

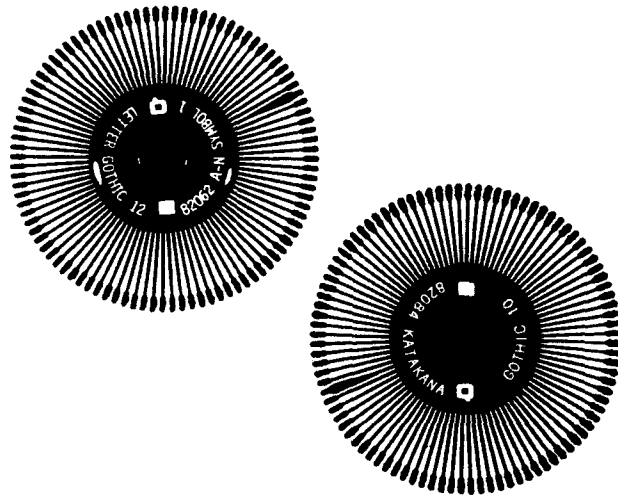
## 6. IMPRESORAS DE TIEMPO DE IMPACTO

Estas impresoras se llaman normalmente líneas impresoras. Las características o espacios que constituyen una línea están impresas simultáneamente a través de líneas enteras. El papel sube y se imprime la otra línea.

La velocidad de la impresora tiene un rango de cien a docientas líneas por minuto.

Porque de la alta velocidad de impresión y del alto coste de equipamiento con relación a los caracteres y al tiempo de impresión, es debido a que las impresoras se usan para altos volúmenes de salidas impresas y menor frecuencias con microprocesadores.

Un tipo de relieve está colocado a través de líneas para su impresión mediante el uso de cada tipo de relieve en el carro de la impresora, consistete en una cadena, tren o una banda moviendose horizontalmente a través del paper y de la línea



**FIGURE 3-6** Daisy wheels are interchangeable to permit printing with different fonts. (Qume Corp.)

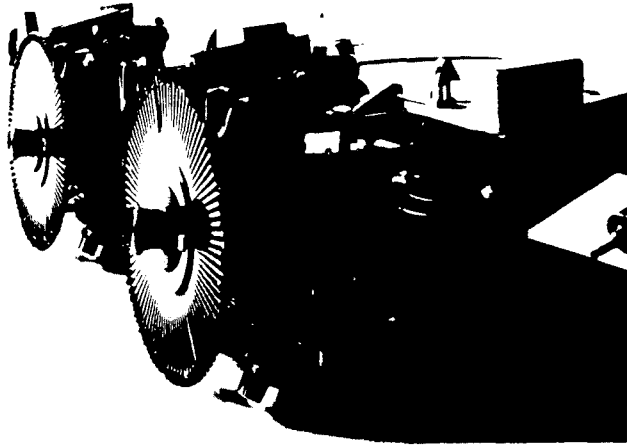
impresora o un tambor en rotación en frente del papel con los caracteres adosados en relieve en las 132 columnas del tambor. Como el tambor gira, las columnas de caracteres pasan verticalmente a través del papel y de la línea impresora. En ambos métodos el martillo se activa cuando se coloca el carácter correcto, imprimiendo el caracter en el papel con la tinta. La

figura 3-10 muestra la banda impresa para una impresora con líneas de impacto . Los caracteres impresos están en relieve sobre la banda . La banda gira entre dos rodillos , pasando en frente del papel . La cinta se coloca entre la banda de desplazamiento y el papel . Como los caracteres impresos en la banda se mueven mediante la impresión de 132 posiciones horizontales , las 132 corresponden a la impresión del martillo entre el papel y la banda en el tiempo apropiado , produciendo la impresión de los caracteres sobre la línea , cada carácter deseado se imprime en 132 posiciones (fig.3-11).

El circuito de tiempo en las impresoras está controlado cuando cada letra o número se coloca en la posición de impresión para controlar el tiempo del martillo . La cadena de impresión ejecuta el mismo principio (fig.3-12). Con esta técnica la cadena gira en frente de la cinta y del papel . Cada eslabón de la cadena está

diseñada para mantener sobre un pallet todos los caracteres en relieve . El martillo está localizado entre el papel y cada 132 golpes del martillo mueven el pallet cuando el carácter deseado pasa a la posición del tiempo de impresión.

Mucho más que poner un carácter en una cadena o banda es que son pocos los caracteres que se imprimen en línea



**FIGURE 3-7** Dual daisy wheels mounted on one carriage permit double-speed printing. (Qume Corp.)

por minuto.

Por esta razón , muchas líneas tienen 64 caracteres del código ASCII. Algunas impresoras incluso usan menos que esta , proporcionando solo 26 caracteres alfanuméricos , 10 números , y unos pocos símbolos . Muchas impresoras son de fácil movimiento para hacer posible cambiar caracteres y como consecuencia ajustar la impresión suficientemente para sentar las necesidades de las aplicaciones inmediatas.

La figura 3-13 ilustra un tabor de una impresora donde cada 64 o 96 caracteres usados están grabados en relieve en 132 columnas alrededor del tambor



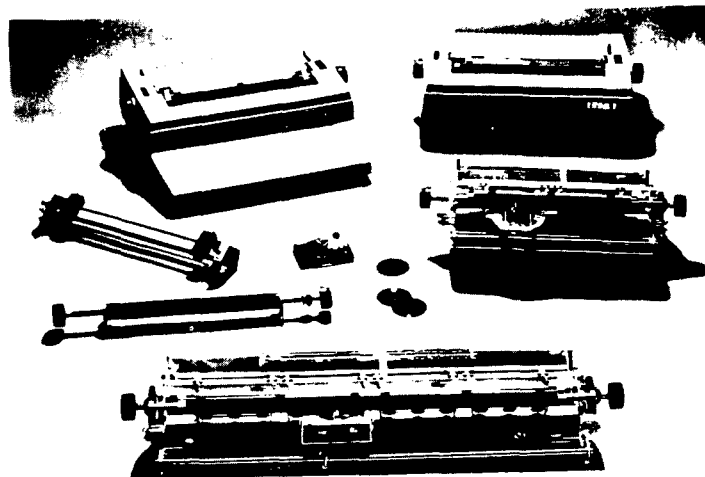
correspondiendo a la impresión. El tambor gira delante del papel y la cinta (fig 3-14). El martillo golpea el papel imprimiendo el carácter desde el tambor a través de la cinta y formando la impresión en la papel.

## 7. IMPRESORAS DE TIEMPO DE NO IMPACTO

Estas impresoras son dispositivos especiales usados para impresoras de salida de alta velocidad. Se usan en fotografía xerográficas, en tecnología laser, forman los caracteres sin golpear la superficie. Por que de sus especiales dispositivos y su alto coste, se usan frecuentemente como periféricos para mini-microcomputadores pero ellos mismos usan mini-microcomputadores como impresoras de control.

Las impresoras laser combina la impresión laser y xerográfica para la impresión a alta velocidad ; desde 1980 la velocidad es posible aumentarla de 18000 a 28000 líneas por minuto, correspondiendo a 300 a 500 líneas por segundo. El rayo laser controlado por el computador imprime datos cargados electricamente rotando alrededor de la correa del tambor. El torno se adhiere al cambio de sección y entonces se transfiere al papel para formar la página impresa.

Los mini-microcomputadores usan el phototypesetters. En estos sistemas son seleccionados por el computador



**FIGURE 3-8** Components and several models of the daisy-wheel impact printer. (Qume Corp.)

las imágenes de los caracteres especificando el tamaño y la proyección óptica en un papel fotosensitivo , donde las líneas de copia son registradas fotográficamente . La exposición del papel se desarrolla , y la copia de la fotoimpresión se usa para hacer presionar los platos para la impresión de los libros , revistas o periódicos.

La proyección de los caracteres en la parte delantera es ejecutado por un tubo de rayos catódicos

NEC has an interface to fit your every need

RS 232

Diablo compatible

Current Loop

Qume compatible

8080 Input Bus

Centronics Type

Special Interface (available by factory quote)

Programmable Vertical Tabbing

Bidirectional Printing

Proportional Spacing

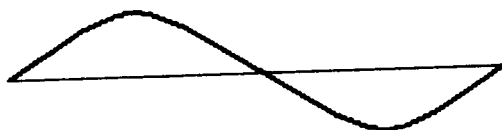
Programmable Horizontal T a b b i n g

Ribbon change - Red to Black

RANDOM CHANGES

Superscripting - Superscripting  $\frac{(X^2+Y^2)}{A_1+B_2}$

Graphics



All Spinwriter Proportional Spacing "THIMBLES" have 100 OR MORE CHARACTERS. The Emperor P.S. "THIMBLE" for example contains the following characters:

ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789\_~%&@:"'![]  
abcdefghijklmnopqrstuvwxyz >\*<./,ç\){\$?(&)=+-;#[+~^

FIGURE 3-9 A print sample demonstrating the flexibility of the thimble printer, similar to the daisy-wheel impact printer. (NEC Information Systems.)

con la generación de caracteres enmascarados para la proyección óptica de los caracteres seleccionados, todo controlado por minicomputador. La figura 3-15 y la figura 3-17 ilustra un minicomputador controlado por un tubo de rayos catódicos, es normalmente un dispositivo especial de impresoras de no impacto.

## IMPRESORAS DE MATRIZ DE PUNTOS

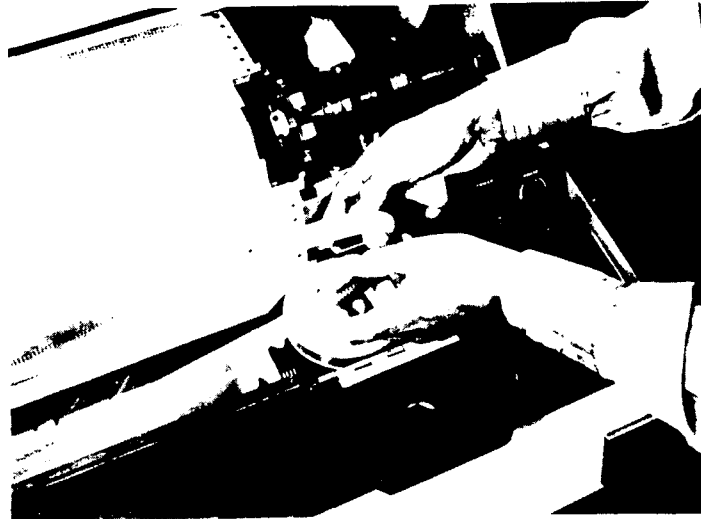
Los caracteres de una matriz de puntos están formados por la impresión de un grupo de puntos para formar una letra , números , y otros símbolos . Este método es ampliamente usado con mini-microcomputadores .

Los puntos están formados por los métodos de impresión de impacto y no impacto .

Los métodos usados para imprimir los puntos son impactos térmico , electrosensitivo , y tecnología de inyección.

La figura 3-18 muestra la letra A formada por un matriz de cinco puntos de anchura y 7 puntos de altura y en una matriz de 9\*7 donde los puntos son separados horizontalmente en nueve pasos. Otra configuración de una matriz son 7\*7 , 8\*8 , 7\*9 , 9\*9 , 7\*12. La figura 3.19 muestra unos 63 caracteres colocados de 7\*7 . La figura 3-20 muestra una norma sencilla de espaciamiento de caracteres. En vista que la figura 3-21 ilustra la expansión de caracteres horizontalmente , todas las demostraciones de la versatilidad de la matriz de caracteres y las impresoras.

Una matriz de puntos de 5\*7 se usa frecuentemente con letras son aceptables todas las mayúsculas .



**FIGURE 3-10** The print band on this band printer is easily changed to permit substitution of character sets. (*Dataproducts Corp.*)

Las impresoras de matriz de puntos pueden imprimir alguna combinación de puntos con todas las posibilidades de imprimir las posiciones en la matriz. Se señala uno de los 128 caracteres del código ASCII y es controlado por una memoria de lectura (ROM), la cual controla los tipos de puntos. Cambiando la ROM los caracteres de algunos lenguajes pueden ser utilizados por las impresoras.

En impresoras donde los puntos horizontales y

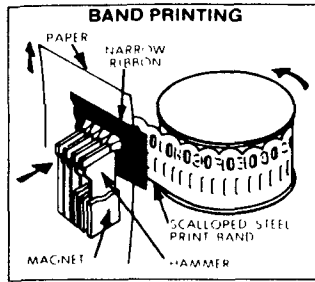
verticales se superponen . Los bloques de cuatro o más puntos pueden imprimirse para formar grandes bloques de caracteres y espacios.

La figura 3-22 ilustra algunos extraños caracteres producidos por una matriz de puntos. La figura 3-22 muestra un forma de impresión de una matriz de puntos simultáneamente con la forma del texto.

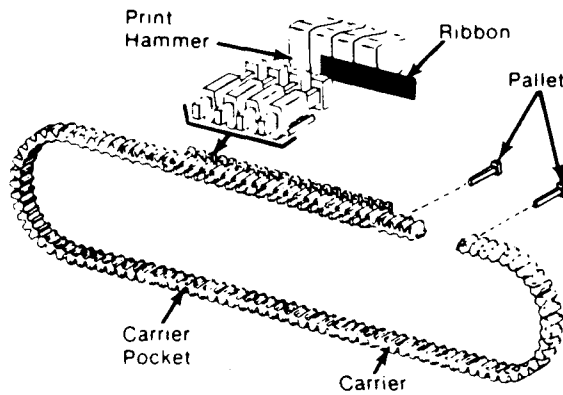
## 9. CARACTERES DE LAS IMPRESORAS CON MATRIZ DE PUNTOS

La cabeza de impresión para una impresora de impacto con matriz puntos esta normalmente compuesta de un conjunto de hilos que golpea los caracteres a través de la cinta(fig.3-24). Por esta razón estas impresoras son algunas veces llamadas "wire-matriz printers". La cabeza de impresión contiene sólo una simple columna de 7 cables , sin embargo puede ser de dos o más columnas de anchos hilos (fig.3-25). Se ilustra un matriz de puntos en la figura 3-26 . La cabeza de impresión se mueve horizontalmente a lo largo de un barra (fig. 3-27) normalmente imprime bidireccionalmente y multiples cabezas de impresión se montan en algunos carros (fig.3-28).

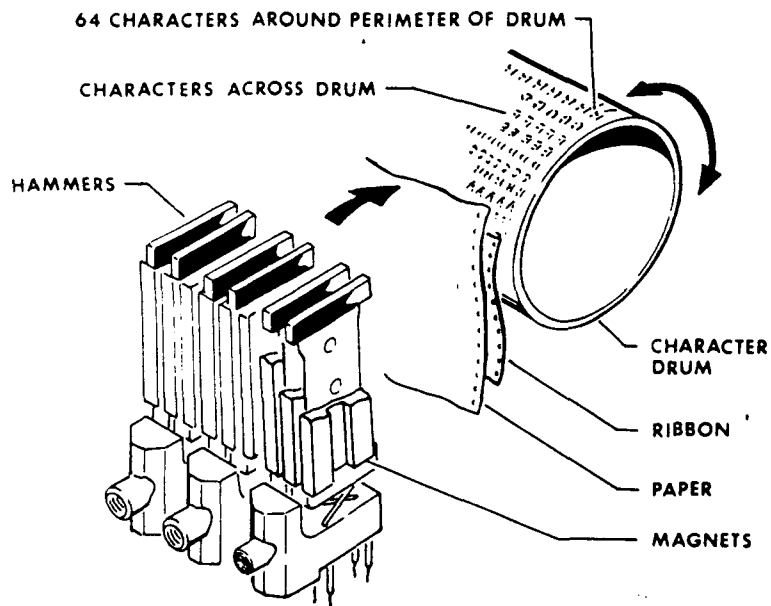
Para la ilustración de los dispositivos nosotros daremos por sentado que la cabeza de impresión contiene un simple columna de 7 hilos . Los hilos estan empujados



**FIGURE 3-11** Band printers (also called belt printers) contain characters embossed on the band. A set of hammers, one for each position, strikes the paper. (Tally Corporation.)



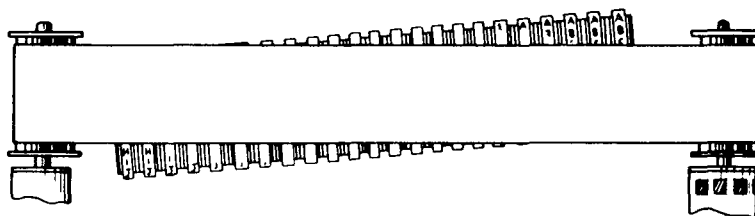
**Figure 3-12** On a chain printer, embossed characters on pallets are rotated on a chain. (Teletype Corp.)



**FIGURE 3-13** On a drum printer characters are embossed around the drum, one set of characters for each print position across the paper. A hammer strikes each time a character to be printed on a line is in position. (Data-products Corp.)

desde la cabeza de impresión (normalmente electromagnéticamente) en cualquier combinación la impresora controla la llamada para la creación de un carácter. Los hilos golpean el tambor y el giro del papel, imprimiendo en una columna vertical sencillos caracteres. Para un matriz de puntos de 5\*7 caracteres, la cabeza de impresión tiene un espacio de un etapa, imprime en la segunda columna de puntos y repite el proceso hasta la impresión de las 5 columnas.

Si la impresora esta diseñada para imprimir puntos, el mismo proceso se usa excepto que la novena etapa de impresión horizontal se usa para formar los caracteres mediante la



**FIGURE 3-14** Canting the ribbon across the face of the drum equalizes ribbon wear. (Control Data Corp.)

formación de los caracteres de una matriz de puntos de



3\*7.

Virtualmente todas las impresoras de impacto de la matriz de puntos usan una o dos columnas verticales de hilos que contienen un conjunto de caracteres de 5\*7 por ejemplo .

La complejidad del mecanismo de la cabeza de impresión con el circuito de control esta incrementada con multiples columna de hilos.

Hablado estrictamente , los caracteres de una matriz de puntos no es normalmente una impresión de caracteres pero imprime en una columna los caracteres en una matriz de puntos . Sin embargo la velocidad de impresión de una impresora de matriz de puntos son superiores a muchas impresoras.

En impresoras de alta velocidad la matriz de puntos puede imprimir en dos direcciones . La mayor desventaja es que los caracteres de una matriz de puntos no tienen apariencia de los de una máquina de escribir formada por caracteres y estas sin embargo no se usan para la aplicación del procesamiento de palabras .

Para el uso de otros ordenadores , sin embargo , son de mayor eficiencia y mecanicamente simple . El esfuerzo de la investigación ahora produce muchos caminos prácticos de una matriz de puntos donde los puntos son pequeños y la matriz es bastante larga para producir un matriz de puntos de caracteres

indistinguibles para la formación de caracteres por impresoras de impacto.

#### 10. LINEA Y TIEMPO DE IMPACTO DE LAS IMPRESORAS DE MATRIZ DE PUNTOS

La línea de un matriz de puntos usa un peine horizontal ensanchado a lo largo de la anchura del papel. El peine de la pieza simple consiste de 132 dedos (correspondiente a los dientes del peine); uno de los puntos salientes esta en cada martillo. Los martillos estan tirados mediante un electromagnetismo bajo el control del generador de caracteres y se desconecta al ir hacia delante, golpeando el tambor sobre el papel (fig.3-30). Cada martillo corresponde a la posición de un carácter. El peine oscila horizontalmente para imprimir cinco puntos constituyendo un fila horizontal de un matriz. El papel avanza verticalmente una posición y se imprime la segunda columna de puntos.

Las impresoras por lo tanto imprimen en una columna de puntos, todos los 132 caracteres, y entoces imprime cada serie de columnas de 132 caracteres hasta que se complete las siete etapas verticales, imprimiendo una matriz de puntos de 5\*7 por solapamiento hasta completar el ancho de 132 caracteres. Los 132 caracteres se imprimen sustancialmente en un línea de

tiempo , pero normalmente en una columna horizontal de puntos apareciendo a través de un página . La velocidad de la línea de impresión es de 125 líneas por minuto para una impresora típica y para todos los dispositivos de estas impresoras de líneas de tiempo.

## 11. MATRIZ DE PUNTOS PARA IMPRESORAS DE NO IMPACTO

La matriz de puntos para impresoras de no impacto pueden señalar directamente sin tocar el papel. Por lo tanto son inmóviles comparadas con las impresoras de impacto . Ellos no pueden ser hacer copias , sin embargo , tan poco se pueden forzar para que imprima el carácter a través de muchas copias de carbón.

Las impresoras de matriz de puntos usan cuatro tecnologías.

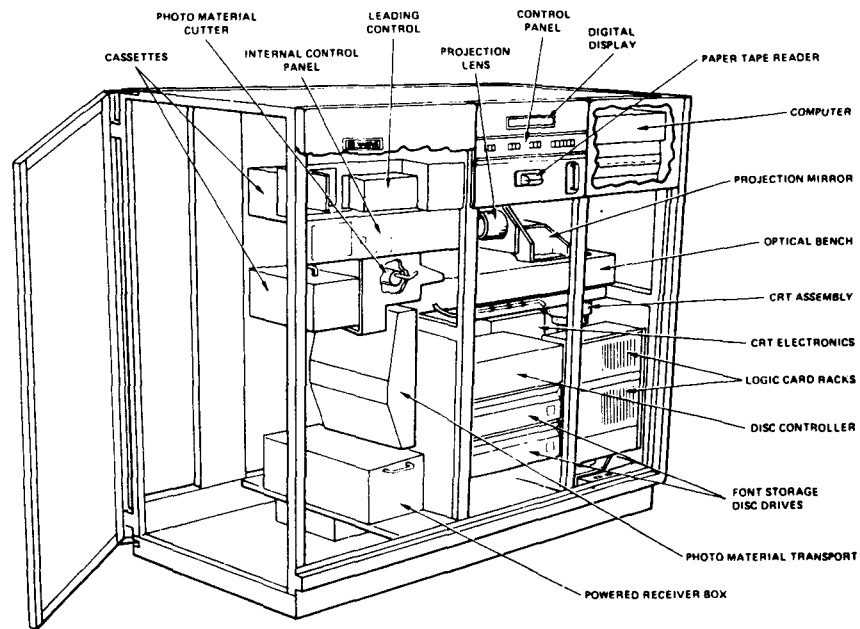
- 1.- Térmica
- 2.- Electrosensitiva
- 3.- Electrostático
- 4.- Inyección

### 11.1 IMPRESORAS TERMICAS

Electricamente las impresoras térmicas el modelo de carácter deseado esta en una matriz de puntos.

El calor activa un colorante sensible al calor que esta recubriendo el papel produciendo un negro o azul

que no aparezca en cada punto luminoso. Las cabezas de impresión térmica en todo caso se une horizontalmente a través del papel, imprimiendo en cada columna los caracteres en serie de una matriz de puntos, o



**FIGURE 3-15** This minicomputer-controlled photooptical line printer is used for typesetting. (Harris Corp.)

imprimiendo una línea horizontal de puntos en el tiempo.

La figura 3-31 muestra una cabeza de impresión serie para la impresión de siete puntos de un matriz. El calentamiento de los elementos resistivos están en la esquina superior derecha de la cabeza de impresión. La

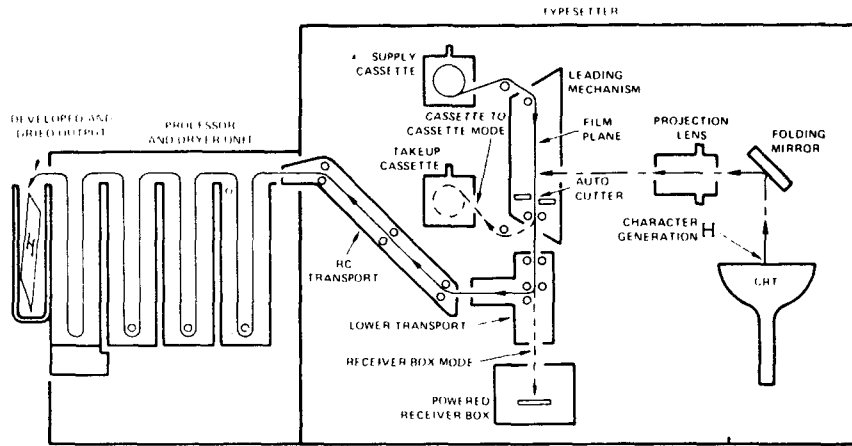


FIGURE 3-16 Schematic diagram of a photooptical line printer. (Harris Corp.)

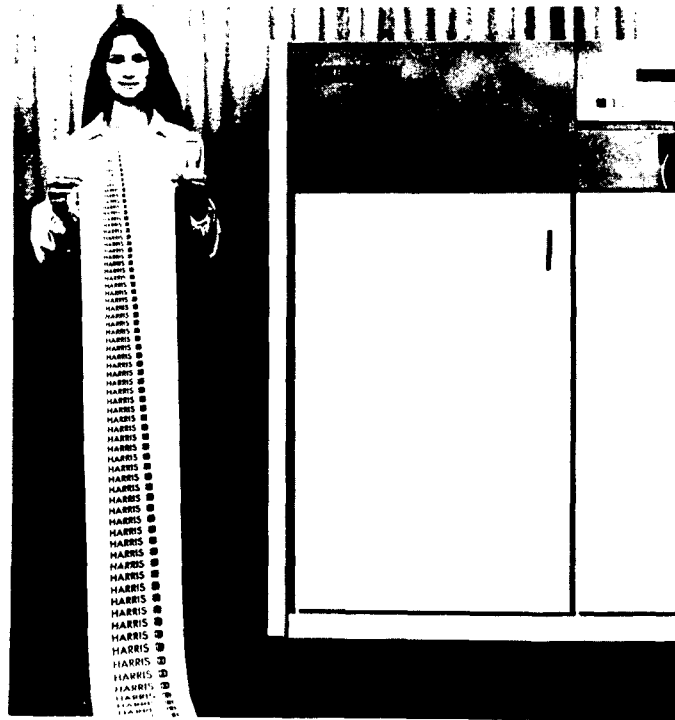
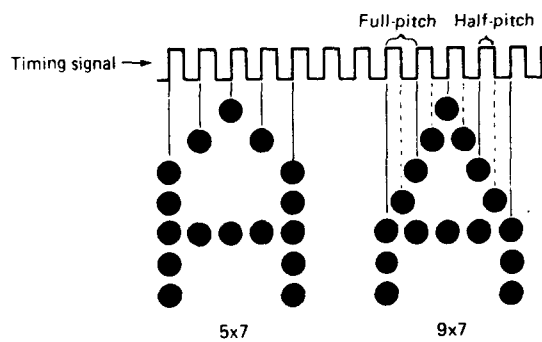


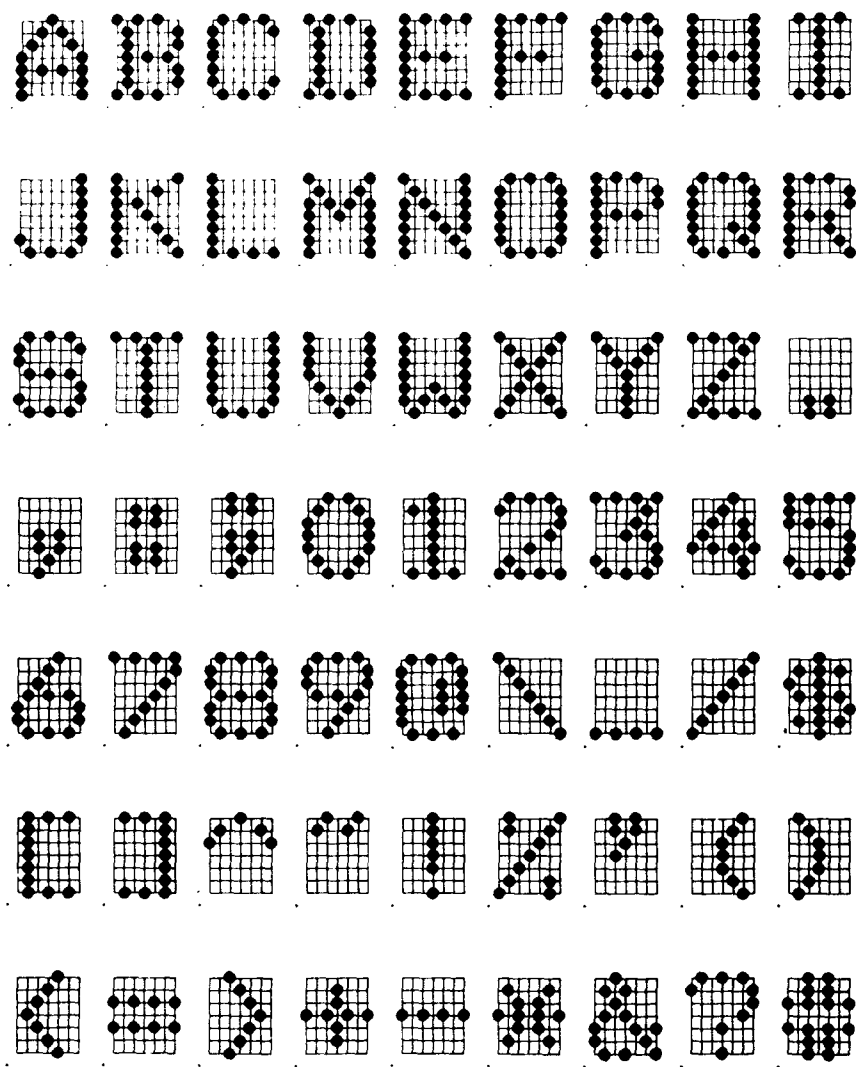
FIGURE 3-17 Sample output of photooptical line printer. (Harris Corp.)

alimentación esta aplicada a los circuitos para cada punto , resultando el calentamiento necesario para la posición produciendo el cambio de color en el papel.

Las impresoras térmicas de líneas y tiempo usan un atenuador fijo de impresión que contiene una fila de elementos resistivos de una pantalla de sustratos cerámico . Las impresoras térmicas son típicamente usadas para calculadoras , instrumentos de impresión (fig.3-32 y fig.3-33).



**FIGURE 3-18** Appearance of the letter A in several dot-matrix sizes. (C. Itoh Electronics, Inc.)



**FIGURE 3-19** A 63-character set of 7 X 7 dot matrix characters (space omitted). (Computer Peripherals, Inc., a subsidiary of Control Data Corporation.)

## 11.2 IMPRESORAS ELECTROSENSITIVAS

La corriente pasa a través de una punta de una aguja o una punta en una cabeza de impresión por un revestimiento especial de un papel sensitivo para un cambio de voltage . En muchas impresoras electrosensitivas el papel tiene un revestimiento de aluminio el cual se vaporiza cuando se aplica el voltage a la cabeza de impresión produciendo una marca negra . Distinto del papel de la impresoras térmicas , las cuales ultimamente reducen la intensidad de la señal mediante una luz o calor , el revestimiento de aluminio en el papel no esta afectado por el calor , la luz o el agua y es por lo tanto un disco permanentemente relativo (fig.3-34). Las impresoras electrosensitivas son frecuentemente usadas para discos o monitores de video.

## 11.3 IMPRESORAS ELECTROSTATICAS

Las tensiones son aplicadas a un conjunto de agujas usadas para formar los caracteres de una matriz de puntos . Las agujas energéticas forman un cambio de puntos con revestimiento dieléctrico sobre el papel como el papel desarrollado desde el rodillo para a través de las agujas. Cuando el papel esta pasando emite un sonido , carga las agujas atrayendo un sonido particular ,





formando los caracteres de un matriz(fig.3-35).

#### 11.4 IMPRESORAS DE INYECCION

La caída de tinta rocia directamente el papel para formar los caracteres de una matriz de puntos mediante impresoras de inyección.

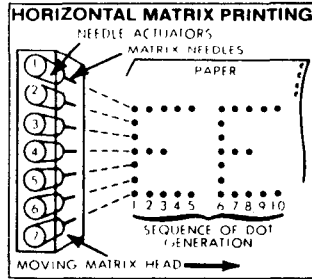
Se utilizan dos principales tecnologías:

En el sistema de demanda de caída , la tinta cae bajo la presión que ejerce sobre la tobera blanca del papel. Un dispositivo vertical en una matriz columna ejecuta la caída de la tinta , formando una columna de matriz de puntos . La tobera esta avanzando horizontalmente sobre la segunda matriz de columna , y se ejecuta la tinta seleccionada , hasta que se forme el caracter.

Las impresoras de inyección de corriente continua bombea la tinta continuamente a través de un tobera . La tinta pasa a través de un tubo donde la caída de la tinta esta cargada electricamente (fig.3-37). El recorrido de la tinta que cae al papel se apunta electricamente mediante una deflexión horizontal y vertical sobre la placa . El cambio en la deflexión de la placa esta controlado por un generador de caracteres electronicos. Este método de direccionamiento de la tinta es análogo al de un monitor de video , donde una corriente de electrones de la pantalla estan

DATE: 7/17/78		HEWLETT-PACKARD COMPANY		ORDER #241402176	
SOLD TO: ABD LIMITED		SHIP TO: ABD LIMITED 123 E. 32 ST. HEWLETT, MONTANA			
CUSTOMER ORDER NUMBER		CUSTOMER NO.	H.P. PURCH. NO.	H.P. QUOTE NO.	
20329003451		115	CS471	SC30791	
REQUIRE DATE	RATING	MFG. DISC.	S.O. DISC.	TERMS	
9-15-78	-	078	047	NET 30	
SHIP VIA INSTRUCTIONS: AIR BEST					
SPECIAL INSTRUCTIONS					
ITEM	PROD. NUMB.	DESCRIPTION	UNIT PRICE	QTY	TOTAL PRICE
01	2631A	PRINTER	3150	5	15750
02	#240	2640 I/F	105	3	315
03	#210	1000 I/F	650	2	1300
COMMENTS					
SHIP DATE	METHOD	CARRIER	FREIGHT	C.O.D.	BOX NUMB.

**FIGURE 3-23** A dot-matrix printer printed this form simultaneously with the data on the form. (Hewlett-Packard.)

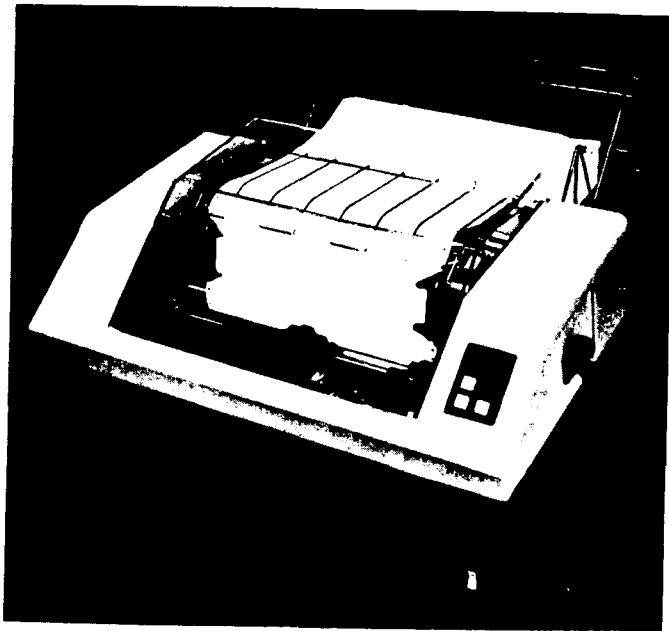


**FIGURE 3-24** Impact dot-matrix print head. Solenoids actuate matrix needles to print vertical column of dots. (Tally Corp.)

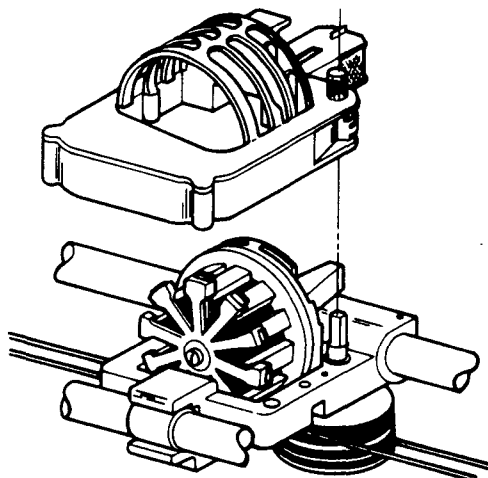


**FIGURE 3-25** Two columns of print wires are visible in this impact dot-matrix print head. (Dataproducts Corp.)

controlados por deflexión de las placas; aquí, sin embargo, la caída de la tinta está guiada por un cambio de electrones. Cuando no se imprime un carácter, las continuas corrientes de tinta es capturada y se mantiene en un depósito donde la tinta es reciclada.



**FIGURE 3-26** The print head of this dot-matrix printer is in the lower center of the paper. (Control Data Corp.)



**FIGURE 3-27** Dot-matrix print head rides across carriage width on two support bars. (Control Data Corp.)

Hablando estrictamente , los caracteres formados por continuas corrientes de tinta los caracteres de la matriz de puntos no están alineados en el sentido en que nosotros previamente lo usamos como término . Sin embargo , están formados por puntos compuestos de tinta extremadamente pequeños. La corriente continua se inyecta imprimiendo la calidad de los caracteres formados . Este sistema de inyección es actualmente un método híbrido de formación de caracteres.

## 12. TRATAMIENTO DEL PAPEL

Normalmente todas las impresoras necesitan mover el papel . Existen tres métodos: Alimentación de Fricción , Alimentación del eje del Rodillo y la alimentación del eje del tractor.

Alimentación de Fricción es el método usado por las máquinas de escribir ordinarias . El papel se mantiene por fricción en frente del rodillo de la máquina de escribir y se mueve cuando el rodillo avanza . La alimentación de fricción es usado en impresoras de baja velocidad.

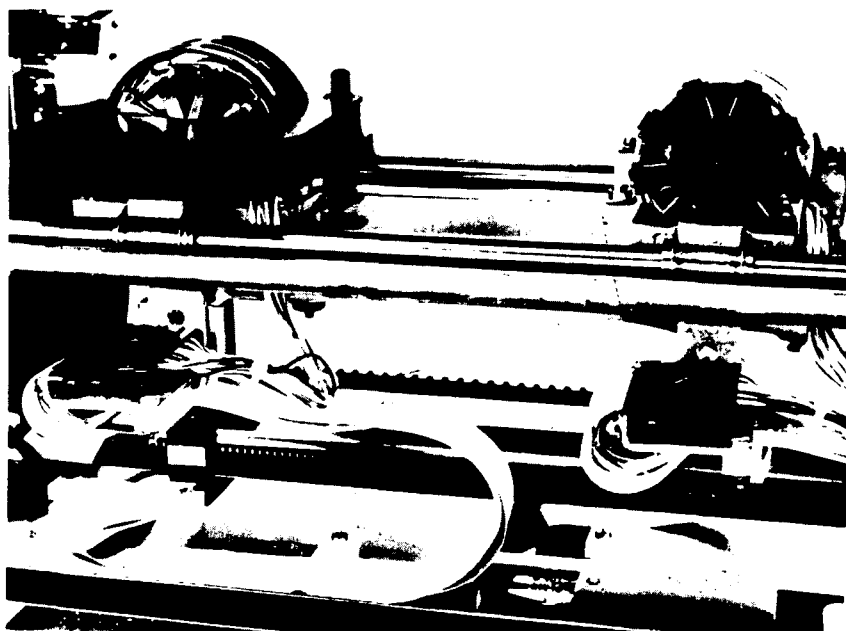
Alimentación del eje del Rodillo usa papel con agujeros perforados en el borde de forma continua . El eje asciende por el rodillo engranando los agujeros y moviendo el papel a través de la impresora . Las

impresoras de media velocidad usan este método para mantener el papel colocado en la impresora. Por que el eje esta montado en el rodillo y no esta ajustado horizontalmente, solo es posible a lo ancho.

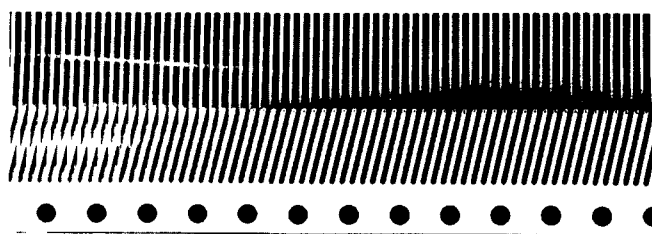
Alimentación del eje del Tractor usa dos motores en forma de tractor (fig.3-38).Cada tractor también contiene un eje, y el espacio horizontal del tractor es ejecutable permitiendo el uso de formas de algún ancho. Muchas de las formas del tractor estan frecuentemente bajo un programa de control por su alta velocidad.

Las impresoras de alta velocidad normalmente incorporan la alimentación del tractor.

Las dobles formas de los tractores se usan con impresoras de rodillo divididos. Este argumento permite simultaneamente el tratamiento de dos formas diferentes de anchos con espacios verticales separados. Ambas formas de tractores estan controladas por separado y tecleados en un espacio vertical. Por ejemplo las impresoras de rodillo dividido permiten escribir automaticamente el nombre propio en un certificado en el rodillo izquierdo mientras mantiene un libro de certificados editados sobre el rodillo derecho. Adicionalmente, los mensajes entre el computador y el operador se guardan en un libro sobre el rodillo derecho.



**FIGURE 3-28** Dual dot-matrix print heads mounted on one carriage double printer speed. (*Control Data Corp.*)

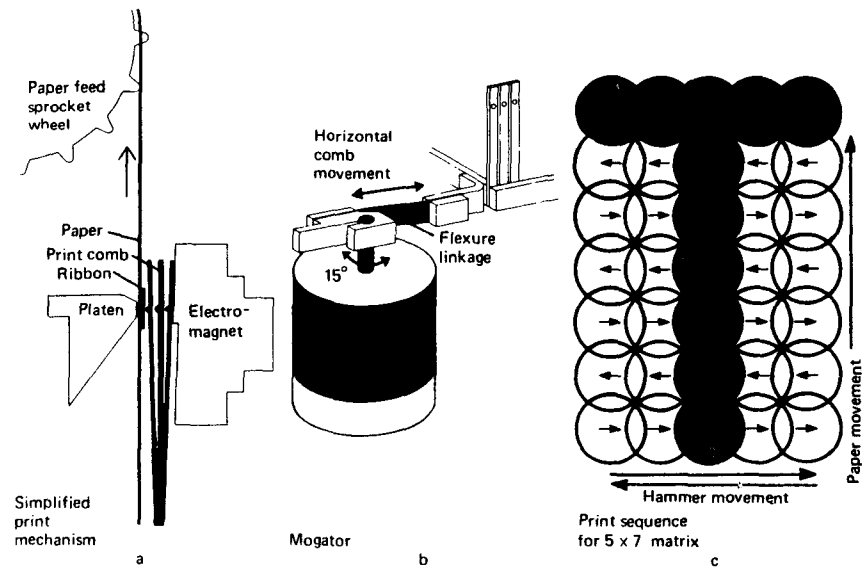


**FIGURE 3-29** Print comb of dot-matrix line printer. (*Tally Corp.*)

Por supuesto una sencilla lamina puede ser alimentada manualmente . Como alternativa tenemos la opción de insertar automáticamente una sencilla lámina siendo utilizado para muchas impresoras usando el procesamiento

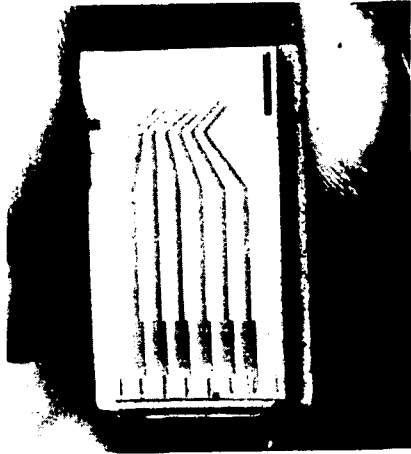
de palabras , particularmente en la separacion de cartas.

El mecanismo de alimentacion frotal se usa en muchas series como el alquiler de coches . Estos dispositivos de alimentacion frotal permite al operador guardar la figura en un plastico frontal . La secuencia de alimentacion vertical y espaciado se ejecuta automaticamente mediante una impresora y su mecanismo de control .



**FIGURE 3-30** (a) Print-comb fingers are pulled away from paper by an electromagnet at each horizontal character position. (b) Raised dots on each finger are moved across each horizontal print position by vibrating linkage to a stepping motor ("mogator"). (c) Comb prints dots in sequence. (Tally Corp.)





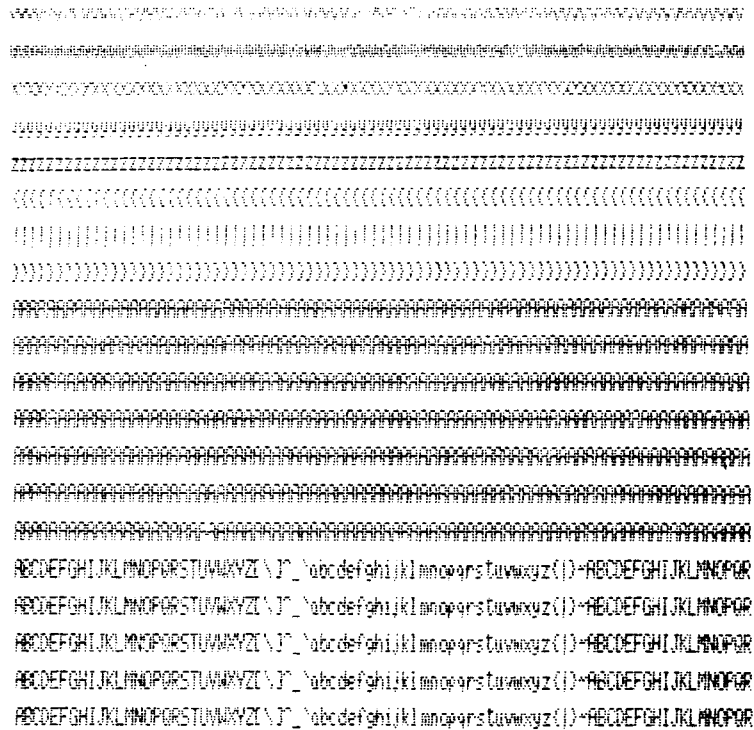
**FIGURE 3-31** Thermal dot-matrix print head for printing a character at a time. (Dataproducts Corp.)



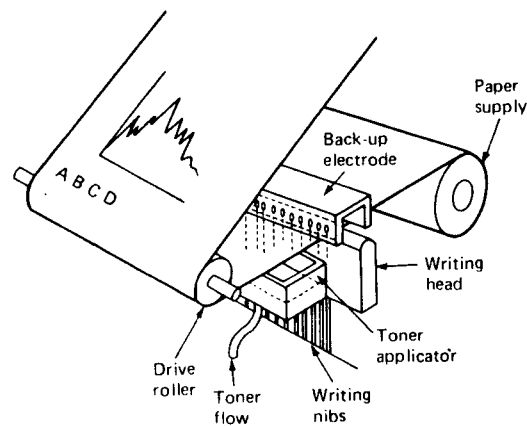
**FIGURE 3-32** Thermal printer used with a programmable calculator prints a line at a time of 20 characters with stationary heating elements.



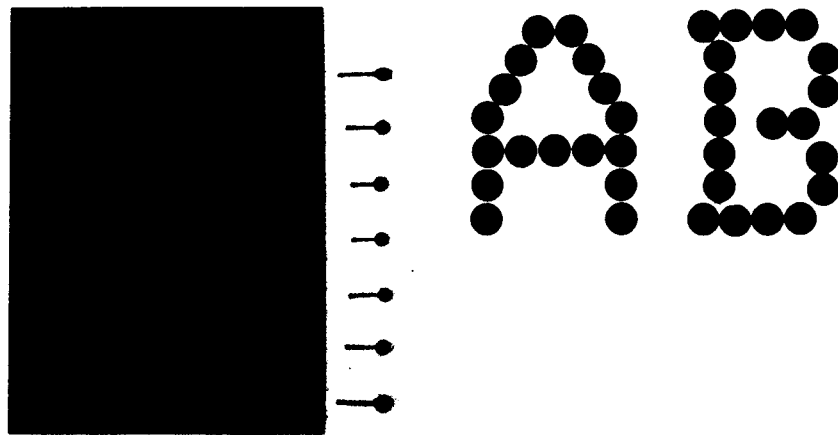
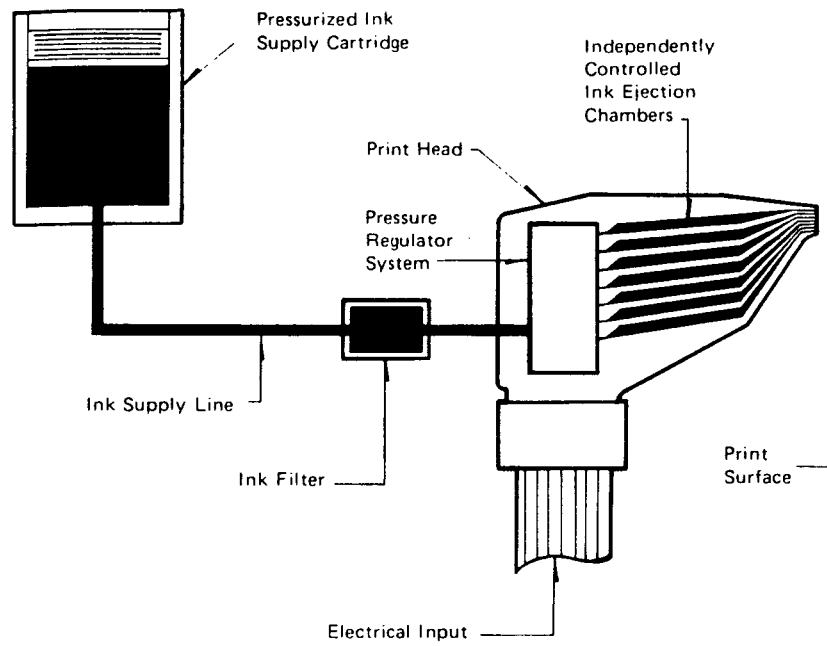
**FIGURE 3-33** Thermal printer used to record the display of a video monitor. (Perkin-Elmer Data Systems.)



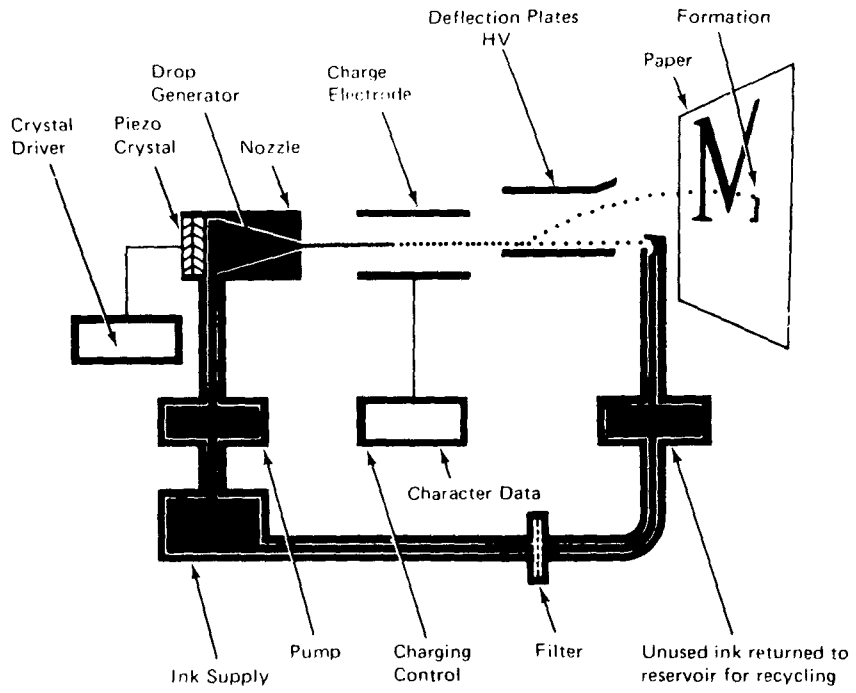
**FIGURE 3-34** Actual-size print sample produced by an electrosensitive printer often used to record display of video monitor. Paper is coated with aluminum.



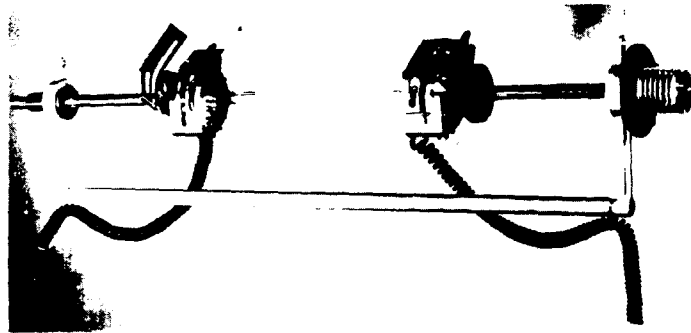
**FIGURE 3-35** Electrostatic printing. Paper specially coated with dielectric material is charged by an array of needles. Characters in the form of charged spots then attract black toner particles.



**FIGURE 3-36** Ink-jet print head sprays droplets of ink to form dot-matrix characters. (Silonics.)



**FIGURE 3-37** Continuous-stream ink-jet printer forms characters by deflecting a charged ink stream in the same way an electron beam is deflected in a cathode-ray tube.



**FIGURE 3-38** Pin-feed form tractors, shown here separated from a printer, permit printing forms of any width. (Control Data Corp.)



**FIGURE 3-39** Front-feed form insertion is convenient for automatic typing of multipart form sets. (Perkin-Elmer Data Systems.)

## TRANSFORMACIONES Y PROYECCIONES EN TRES DIMENSIONES

### 1. INTRODUCCION

Antes de entrar en el tema a desarrollar en parte se repasa brevemente algunos conceptos a tener en cuenta.

Las coordenadas homogneas en tres dimensiones se definen en un punto como :

$$X = x1/x4 \quad Y = x2/x4 \quad Z = x3/x4$$

$$(X, Y, Z) = (x1/x4, x2/x4, x3/x4, x4/x4) = (x1, x2, x3, x4)$$

Ejemplo

El punto (3,-1,2) dado en coordenadas cartesianas ,se puede poner en coordenadas homogeneas de la siguiente forma:

$$(3, -1, 2, 1) = (6, -2, 4, 2)$$

Como podemos observar en este ejemplo un mismo punto se puede representar por infinitas ternas de números reales. Dos ternas que representan el mismo punto tiene sus elementos correspondientes proporcionales.

Por tanto las coordenadas homogneas en tres dimensiones es (X,Y,Z,1).

La matriz transformación es aquella que posee los parametros necesarios para la traslación,estrechamiento

definida por la matriz  $V$  que sera el producto vectorial de las coordenadas homogeneas y la matriz transformación:

$$V = U \cdot T$$

NOTA

Valores de los angulos

$$\phi = 22.208^\circ$$

$$\theta = 20.705^\circ$$

, rotación , reflexión , proyección isométrica , proyección  
simétrica

La matriz transformación  $4 \times 4$  es la siguiente:

$$T = \begin{vmatrix} a & b & c & p \\ d & e & f & q \\ h & i & j & r \\ l & m & n & s \end{vmatrix}$$

## 2. CAMBIO DE ESCALA EN TRES DIMENSIONES

El cambio de escala se define por las coordenadas  
homogéneas y la matriz transformación.

Las coordenadas homogéneas serán:

$$U = (X, Y, Z, 1)$$

La matriz transformación T sera:

$$T = \begin{vmatrix} a & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & j & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

El cambio de escala de cualquier figura vendra definida por la matriz V que sera el producto vectorial de las coordenadas homogeneas y la matriz transformación:

$$V = U * T$$



### 3. ROTACION EN TRES DIMENSIONES

La rotación se define por las coordenadas homogéneas y la matriz transformación.

Las coordenadas homogéneas serán:

$$U = (X, Y, Z, 1)$$

La matriz transformación T será :

$$T = \begin{vmatrix} n_1^2 + (1 - n_1^2) \cos \theta & n_1 n_2 (1 - \cos \theta) + n_3 \sin \theta & n_1 n_3 (1 - \cos \theta) - n_2 \sin \theta & 0 \\ n_1 n_2 (1 - \cos \theta) - n_3 \sin \theta & n_2^2 + (1 - n_2^2) \cos \theta & n_2 n_3 (1 - \cos \theta) + n_1 \sin \theta & 0 \\ n_1 n_3 (1 - \cos \theta) + n_2 \sin \theta & n_2 n_3 (1 - \cos \theta) - n_1 \sin \theta & n_3^2 + (1 - n_3^2) \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

La rotación en tres dimensiones de cualquier figura vendrá definida por la matriz V que será el producto vectorial de las coordenadas homogéneas y la matriz transformación.

$$V = U \cdot T$$

NOTA

$n_1$  = rotación alrededor del eje X

$n_2$  = rotación alrededor del eje Y

$n_3$  = rotación alrededor del eje Z

#### 4. REFLEXION EN TRES DIMENSIONES

La reflexión en tres dimensiones se define por las coordenadas homogéneas y la matrix transformación.

Las coordenadas homogéneas serán :

$$U = (X, Y, Z, 1)$$

La matrix transformación T será:

$$T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Para el eje X-Y

$$T = \begin{vmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Para el eje Y-Z

$$T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Para el eje X-Z

La reflexión en tres dimensiones de cualquier figura vendrá definida por la matriz V que será el producto vectorial de las coordenadas homogéneas y la matriz transformación.

$$V = U * T$$

## 5. TRASLACION EN TRES DIMENSIONES

La traslación se define por las coordenadas homogéneas y la matriz transformación.

Las coordenadas homogéneas serán:

$$U = (X, Y, Z, 1)$$

La matriz transformación T será:

$$T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{vmatrix}$$

La traslación en tres dimensiones de cualquier figura vendrá definida por la matriz V que será el producto vectorial de las coordenadas homogéneas y la matriz transformación.

$$V = U * T$$

## 6. PROYECCION ISOMETRICA EN TRES DIMENSIONES

Las proyecciones isometricas se define por las coordenadas homogeneas y la matriz transformación .

Las coordenadas homogeneas seran:

$$U = (X, Y, Z, 1)$$

Las matriz tranformación T sera:

$$\begin{vmatrix} \cos \phi & \sin \phi \sin \theta & -\sin \phi \cos \theta & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ \sin \phi & -\cos \phi \sin \theta & \cos \phi \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Por tanto la matriz T queda:

$$\begin{vmatrix} 0.925820 & 0.133631 & -0.353553 & 0 \\ 0 & 0.935414 & 0.353553 & 0 \\ 0.377964 & -0.32751 & 0.866025 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

La proyección dimétrica de cualquier figura vendra

## 7. PROYECCION DIMETRICA EN TRES DIMENSIONES

La proyeccion dimétrica se define por las coordenadas homogeneas y la matriz transformación.

Las coordenadas homogeneas seran:

$$U = (X, Y, Z, 1)$$

La matriz transformación T sera:

$$\begin{vmatrix} \cos\phi & \sin\phi \sin\theta & -\sin\phi \cos\theta & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ \sin\phi & -\cos\phi \sin\theta & \cos\phi \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Por tanto la matriz T queda:

$$\begin{vmatrix} 0.707107 & 0.408248 & -0.577353 & 0 \\ 0 & 0.816597 & 0.577343 & 0 \\ 0.707107 & -0.408248 & 0.577353 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

La proyección dimétrica de cualquier figura vendrá

definida por la matriz  $V$  que sera el producto vectorial de las coordenadas homogeneas y la matriz transformación.

$$V = U * T$$

NOTA

Valores de los angulos

$$\varnothing = 45^\circ$$

$$\theta = 35.26439^\circ$$



PARTE II

## 1. INTERFACE DE CONEXION ENTRE EL SISTEMA DE DESARROLLO Y EL PLOTTER.

### 1.1 DESCRIPCION GENERAL

La idea general del proyecto se basa en trazar gráficas con el PLOTTER X-Y PM 8043 conectandolo al sistema de Desarrollo por medio de este interface.

Este interface consta de una serie de integrados que ahora describiremos.

Quizas el más interesante por su funcionamiento es el DAC 80 .

El DAC 80 se caracteriza por poseer 12 entradas siendo la más significativa BIT1 y la menos significativa la BIT12 . Los niveles de tensión de las entradas es de  $-1.0v$  a  $+7v$ .

La alimentación del integrado es de  $\pm 18v$ .

La salida del convertidor puede ir desde 00 a FF . Graduando convenientemente el plotter se consiguió que FF coincidiera con el máximo valor en la escala del plotter , tanto en el eje X como en el eje Y de valor de tensión 5v y 0v con el menor valor en la escala.

Por tanto necesitamos un nivel de tensión de salida de 5v.

Existe una tabla de conexiones para este nivel que es

la siguiente:

OUTPUT RANGE	CONEXION DE LA PATILLA 15	CONEXION DE LA PATILLA 17	CONEXION DE LA PATILLA 19	CONEXION DE LA PATILLA 16
0 a 5V	18	21	20	21

## 1.2 DESCRIPCION DE LOS INTEGRADOS

- 1.- Un contador 7493.
- 2.- un decodificador/demultiplexor 741s138.
- 3.- Dos inversores 7404
- 4.- Tres registros de desplazamiento de 8 bits 741s273.
- 5.- Dos DAC 80.

## 1.3 EXPLICACION DEL HARDWARE

El sistema de Desarrollo posee unas conexiones con Perifericos mediante 6 conectores traseros.

1. Salida TTY.
2. Terminal
3. Salida para tarjeta perforada . Punch.
4. Lectora de tarjeta.
5. Programador Universal de Prooms.
6. Salida para impresoras de línea .

Estan siendo utilizadas dos salidas ,concretamente la salida para impresoras de línea y la del terminal .

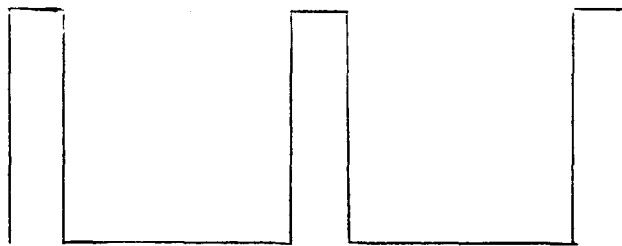
La salida TTY podria haberse utilizado pero era poco práctico por su complicado hardware . Por lo que se utiliza la salida para tarjeta perforada PUNCH.

El conector PUNCH se caracteriza por tener 25 patillas .

El controlador I/O paralelo 8741 activa al integrado 741s154 generado las señal PUNCH COMMAND.

Esta señal corresponde en el conector a la patilla 11. Posee un periodo de  $T=80$  seg . Su frecuencia es  $1/T = 1/80 = 12.5$  KHz. El ancho del pulso es  $T_w = 1.2 * 10$  seg = 1200 nseg.

Seguidamente se muestra la señal aproximadamente.



La patilla 12 corresponde al PUNCH READY , se activa cuando el PUNCH COMMAND esta preparado para su ejecución

La patilla 13 corresponde al SISTEM READY ,nos indica la alimentación del PUNCH.

Las patillas 11 , 23 ,25 corresponde a la tierra . Las restantes patillas no se utilizan.

Seguidamente describire el funcionamiento del montaje . La señal PUNCH funciona como el reloj que se aplica al contador 7493 a traves de la entrada CP0 activa al nivel bajo . La entrada CP1 activa a nivel bajo debe estar conectada a la salida Q0. En este caso nos interesa sólo dos salidas Q0 y Q1 las restantes no se utilizan . Para que se active la cuenta es necesario que las entradas RESET esten conectadas como indica la tabla:

ENTRADA RESET		SALIDA
MR1	MR2	Q0 Q1 Q2 Q3
L	H	cuenta
H	L	cuenta
L	L	cuenta

Por tanto el contador 7493 esta diseñado para que cuente por cuatro . Las combinaciones que resultan estan indicadas en la tabla.

	$Q_0$	$Q_1$
0	0	0
1	0	1
2	1	0
3	1	1

Las salidas  $Q_0$  y  $Q_1$  se aplican a las entradas  $A_0$  y  $A_1$  del decodificador / demultiplexor 741s138 mientras que la otra entrada no se utiliza.

Las salidas utilizadas por el decodificador / demultiplexor son  $Q_1, Q_2$  y  $Q_3$  debido a que nos interesa las siguientes combinaciones :

$\bar{Q}_1$	$\bar{Q}_2$	$\bar{Q}_3$
0	1	1
1	0	1
1	1	0

Estas salidas son activas a nivel bajo . Posteriormente estas salidas se han invertido mediante el inversor 7404 obteniendose la siguientes combinaciones:

$Q_1$	$Q_2$	$Q_3$
1	0	0
0	1	0
0	0	1

Hemos invertido estas señales para aplicarlas a la

entrada de reloj del registro de desplazamiento 74ls273. Son activas a nivel alto .

Cuando se selecciona un registro la información que esta a la entrada pasa a través de "1 obteniendola a la salida . Para el primer registro tendremos la información del lápiz cuya salida se obtiene a través de Q0 . La información que tendremos sera un 1 o un 0 que nos indicará lápiz arriba o lápiz abajo. El segundo registro corresponderá al eje X y el tercer registro corresponderá al eje Y cuyos funcionamiento serán análogos a los anteriores.

Las salidas del registro X que serán Q0/Q7 se conectarán a un DAC 80 y lo mismo ocurre para las salidas del registro Y.

Los registros de desplazamientos poseen una entrada MASTER RESET que estaran conectadas entre si y a su vez estará conectada a la entrada MASTER RESET del contador ,pero a través de un inversor por que es activo a nivel a nivel bajo . Para poder activar el MASTER RESET tenemos un interruptor conectado entre Vcc y tierra a través de una resistencia de 1 K ohmios.

El funcionamiento del DAC 80 es el de convertir una señal digital a analógica , debido a que las entradas del plotter son analógicas.

El bus de datos es paralelo de ocho bits de salidas y activas a nivel bajo y por tanto es necesario al

utilización de inversores . Una vez invertidas las salidas se conectan a las entradas de los registros de desplazamiento.

Como pretendo obtener la máxima resolución del plotter me encuentro con un inconveniente . Debido a que su zona de funcionamiento es muy limitada , la máxima resolución del Plotter sera hasta 9 bits . Si es mayor no podrá activarse.

Como el bus de datos es de 8 bits tendre que obtener el otro bits restante del registro del lápiz mediante una repetición de un dato.

Por tanto del registro del lápiz obtendremos las dos entradas que nos hacen falta para obtener los 9 bits. Estas dos entradas serán las menos significativas del DAC 80.

Mediante el primer DAC obtendremos la salida del eje X y mediante el segundo DAC la salida del eje Y.

Las salidas del lápiz ,del eje X y del eje Y irán conectadas al plotter a través de un conector RS232.

La información que proviene del bus de datos serán convertidas por el DAC 80 en distitos niveles de tensión que posteriormente serán dibujados por el plotter correspondiendo a una figura.

Posee unas variables bien diferenciadas que son: U , L , X .

La variable U indica el lápiz arriba , la variable L



indica el lápiz abajo y la variable X indica la finalización del proceso.

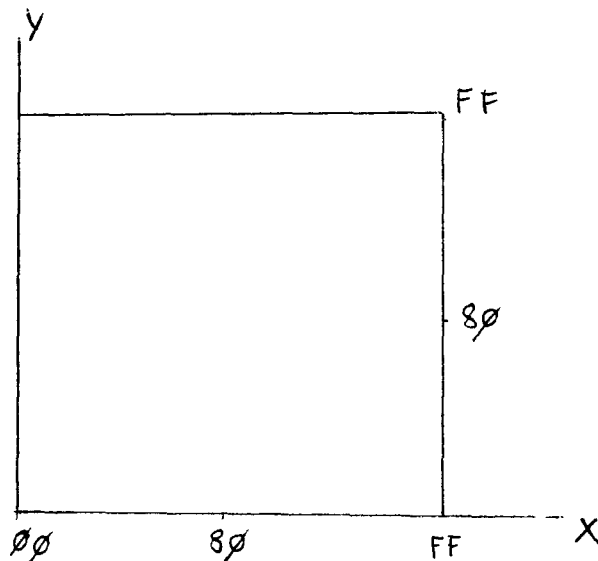
Cada una de estas variables van acompañadas de las coordenadas en el eje X e Y. Por ejemplo U0000.

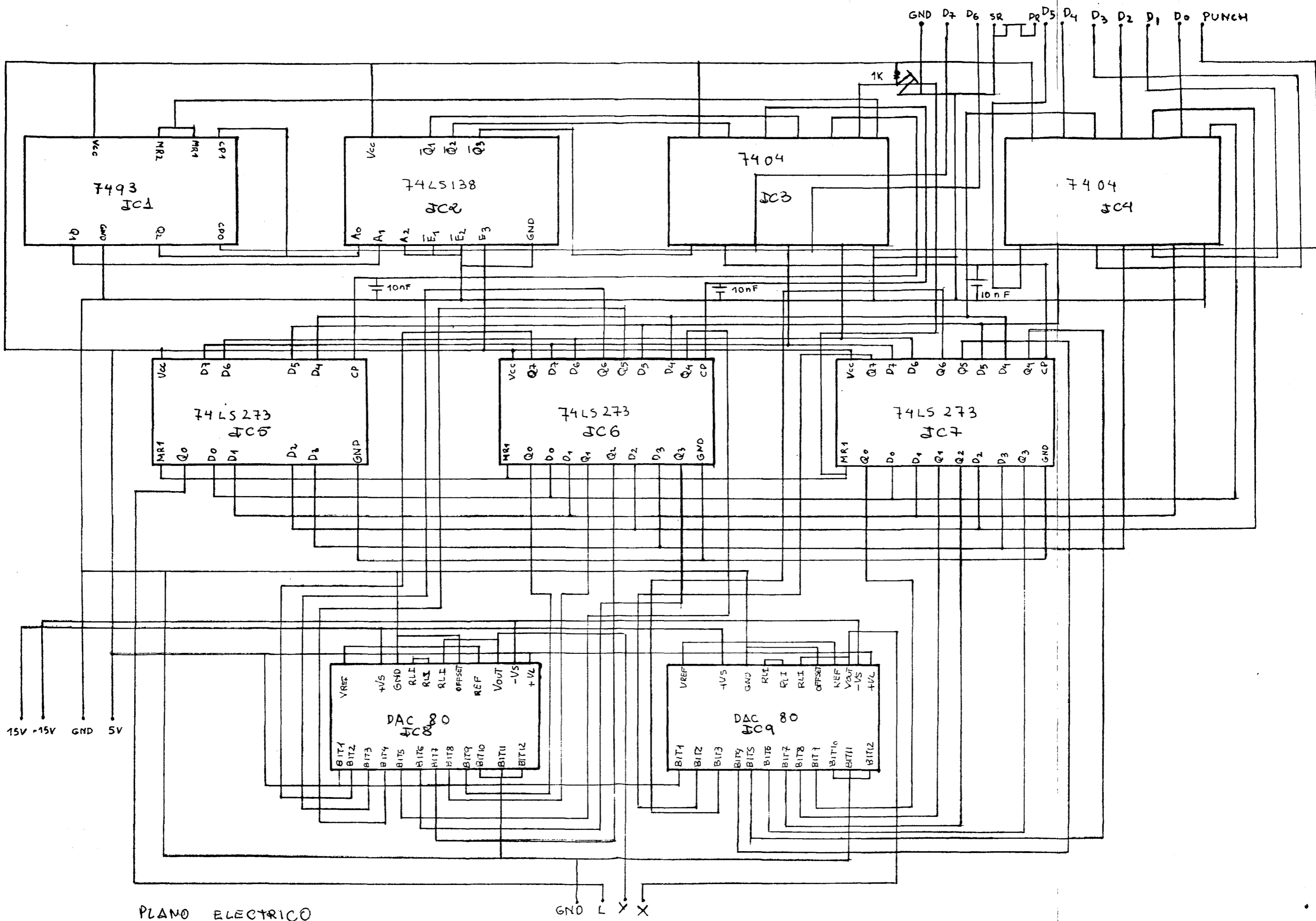
Vamos a explicar su funcionamiento mediante un sencillo ejemplo que es la calibración del plotter:

U0000L0000L8000LFF00LFF80LFFFFL80FFL00FFL0080L0000X

A la hora de trazar una recta entre dos puntos debemos de tener en cuenta un punto inicial de X que llamamos X1 y un punto final de X que lo llamamos X2.

Mediante la ecuación de la recta obtenemos el desplazamiento de X que es igual a  $X = X2 - X1$  y de la misma forma obtenemos el desplazamiento de Y que es igual a  $Y = Y2 - Y1$ .





PLANO ELECTRICO

PRESUPUESTO

INTEGRADOS	UNIDADES	PVP (Pts)
7493	1	300
74LS138	1	300
7404	2	300
74LS273	3	300
DAC 80	2	8.000
RESISTENCIA		
1K	1	25
CONDENSADORES		
10nF	3	50
CAJA	1	4.000

En total, el presupuesto asciende a:

22.273 Pts.

### ALTERNATE TRIM CIRCUIT

As shown in Figures 1 and 2, offset trimming is accomplished by using an external potentiometer connected through a 3.9MΩ resistor to pin 21 of the AD DAC71 and AD DAC72. In some applications, it may be desirable to use an equivalent "T" network of lower-value resistors. A suggested circuit is shown in Figure 6. These resistors and the trimming potentiometer should be located close to the DAC to minimize noise pickup.

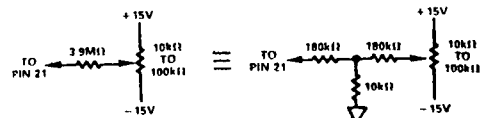


Figure 6. Equivalent Trim Circuit

### DRIVING RESISTIVE LOADS

The AD DAC71 and AD DAC72 current output models can drive resistive loads directly without the use of an external op amp as a current-to-voltage converter.

#### Unipolar Models

The unipolar current output version of the AD DAC71 and AD DAC72 delivers a nominal ±2mA full scale output. Since the output compliance voltage range is specified as ±2.5V, the parallel combination of the DAC resistance and load resistance must be held below a maximum value of 750Ω, limiting load resistance to a maximum of 857Ω. The output voltage for lower values of  $R_L$  can be computed from the formula:

$$V_{OUT} = -2mA \frac{(6k\Omega \times R_L)}{(6k\Omega + R_L)}$$

The output voltage thus produced will always be negative, and must be limited to a maximum of -1.5V for proper operation.

### ORDERING GUIDE

Model	Output	Input Code	Temp Range	Seal	Package Identification
AD DAC71-COB-I	Current	Comp. Offset Binary	0 to +70°C	Polymer	HY24B
AD DAC71-CSB-I	Current	Comp. Straight Binary	0 to +70°C	Polymer	HY24B
AD DAC71H-COB-I	Current	Comp. Offset Binary	0 to +70°C	Hermetic	HY24B
AD DAC71H-CSB-I	Current	Comp. Straight Binary	0 to +70°C	Hermetic	HY24B
AD DAC72C-COB-I	Current	Comp. Offset Binary	0 to +70°C	Hermetic	HY24B
AD DAC72C-CSB-I	Current	Comp. Straight Binary	0 to +70°C	Hermetic	HY24B
AD DAC72-COB-I	Current	Comp. Offset Binary	-25°C to +85°C	Hermetic	HY24B
AD DAC72-CSB-I	Current	Comp. Straight Binary	-25°C to +85°C	Hermetic	HY24B
AD DAC71-COB-V	Voltage	Comp. Offset Binary	0 to +70°C	Polymer	HY24B
AD DAC71-CSB-V	Voltage	Comp. Straight Binary	0 to +70°C	Polymer	HY24B
AD DAC71H-COB-V	Voltage	Comp. Offset Binary	0 to +70°C	Hermetic	HY24B
AD DAC71H-CSB-V	Voltage	Comp. Straight Binary	0 to +70°C	Hermetic	HY24B
AD DAC72C-COB-V	Voltage	Comp. Offset Binary	0 to +70°C	Hermetic	HY24B
AD DAC72C-CSB-V	Voltage	Comp. Straight Binary	0 to +70°C	Hermetic	HY24B
AD DAC72-COB-V	Voltage	Comp. Offset Binary	-25°C to +85°C	Hermetic	HY24B
AD DAC72-CSB-V	Voltage	Comp. Straight Binary	-25°C to +85°C	Hermetic	HY24B

<sup>1</sup>See Section 19 for package outline information.

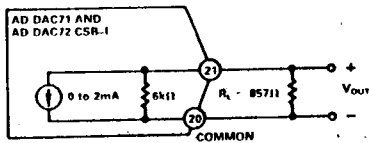


Figure 7. Connections for Unipolar Output AD DAC71 and AD DAC72 Driving Resistive Load

#### Bipolar Models

The bipolar current output versions of the AD DAC71 and AD DAC72 produce a nominal ±1mA output range. The output resistance of the bipolar versions is approximately 3kΩ. The external load resistance for a given output voltage range can be computed from the relationship:

$$V_{OUT} = \pm 1mA \frac{(3k\Omega \times R_L)}{(3k\Omega + R_L)}$$

The maximum permissible voltage swing is -1.5 to +10V, which limits the maximum value of  $R_L$  to 3kΩ.

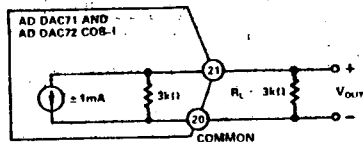


Figure 8. Connections for Bipolar Output AD DAC71 and AD DAC72 Driving Resistive Load



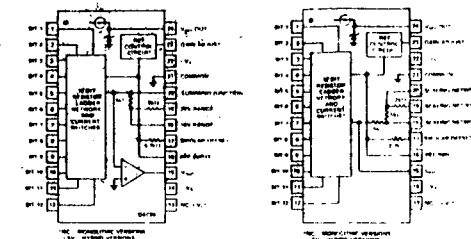
# Complete Low Cost 12-Bit Monolithic D/A Converter

## AD DAC80, AD DAC85, AD DAC87

### FEATURES

- Single Chip Construction
- On-Board Output Amplifier
- Low Power Dissipation: 300mW
- Monotonicity Guaranteed over Temperature
- Guaranteed for Operation with ±12V Supplies
- Improved Replacement for Standard DAC80, DAC800, HI-5880
- High Stability, High Current Output
- Buried Zener Reference
- Laser Trimmed to High Accuracy: ±1/2LSB max Nonlinearity
- Low Cost Plastic Packaging

### AD DAC80 SERIES FUNCTIONAL BLOCK DIAGRAM



### PRODUCT DESCRIPTION

The AD DAC80 Series is a family of low cost 12-bit digital-to-analog converters with both a high stability voltage reference and output amplifier combined on a single monolithic chip. The AD DAC80 Series is recommended for all low cost 12-bit D/A converter applications where reliability and cost are of paramount importance.

Advanced circuit design and precision processing techniques result in significant performance advantages over conventional DAC80 devices. Innovative circuit design reduces the total power consumption to 300mW which not only improves reliability but also improves long term stability.

The AD DAC80 incorporates a fully differential, non-saturating precision current switching cell structure which provides greatly increased immunity to supply voltage variation. This same structure also reduces nonlinearities due to thermal transients as the various bits are switched; nearly all critical components operate at constant power dissipation. High stability, SiCr thin film resistors are trimmed with a fine resolution laser, resulting in lower differential nonlinearity errors. A low noise, high stability, subsurface zener diode is used to produce a reference voltage with excellent long term stability, high external current capability and temperature drift characteristics which challenge the best discrete zener references.

The AD DAC80 Series is available in three performance grades and two package types. The AD DAC80 is specified for use over the 0 to +70°C temperature range and is available in both plastic and ceramic DIP packages. The AD DAC85 and AD DAC87 are available in hermetically sealed ceramic packages and are specified for the -25°C to +85°C and -55°C to +125°C temperature ranges.

### PRODUCT HIGHLIGHTS

- The AD DAC80 series of D/A converters directly replaces all other devices of this type with significant increases in performance.
- Single chip construction and low power consumption provides the optimum choice for applications where low cost and high reliability are major considerations.
- The high speed output amplifier has been designed to settle within 1/2LSB for a 10V full scale transition in 2.0μs, when properly compensated.
- The precision buried zener reference can supply up to 2.5mA for use elsewhere in the application.
- The low TC binary ladder guarantees that all units are monotonic over the specified temperature range.
- System performance upgrading is possible without redesign.

### PRODUCT OFFERING

Analog Devices has developed a number of technologies to support products within the data acquisition market. In serving the market new products are implemented with the technology best suited to the application. The DAC80 series of products was first implemented in hybrid form and now it is available in a single monolithic chip. We will provide both the hybrid and monolithic versions of the family so that in existing designs changes to documentation or product qualification will not have to be done. Specifications and ordering information for both versions is delineated in this data sheet.



# SPECIFICATIONS (T<sub>A</sub> = +25°C, rated power supplies unless otherwise noted.)

Model	AD DAC8M1		AD DAC8MIL		AD DAC87		Units
	Min	Max	Min	Max	Min	Max	
<b>TECHNOLOGY</b>	Hybrid		Hybrid		Hybrid		
<b>DIGITAL INPUTS</b>							
Binary - CBI		12		12		12	Bus Digits
RCD-CCD							
Logic Levels (TTL Compatible)							
V <sub>OH</sub> (Logic "1")	2.0	5.5	2.0	5.5	2.0	5.5	V
V <sub>OL</sub> (Logic "0")	0	0.8	0	0.8	0	0.8	V
I <sub>OH</sub> (V <sub>OH</sub> = 5.5V)	1		1		1		μA
I <sub>OL</sub> (V <sub>OL</sub> = 0.8V)	100		100		100		μA
<b>TRANSFER CHARACTERISTICS</b>							
<b>ACCURACY</b>							
Linearity Error (at +25°C)		±1/2		±1/2	±1/4	±1/2	LSB <sup>1</sup>
CBI							LSB
CCD							LSB
T <sub>1</sub> for T <sub>min</sub> to T <sub>max</sub>		±1/2		±1/4			
Differential Linearity Error (at +25°C)					±1/2		LSB
CBI							LSB
CCD							LSB
T <sub>1</sub> for T <sub>min</sub> to T <sub>max</sub>		±1		±1			LSB
Gain Error <sup>2</sup>	±0.1		±0.1		±0.1		%FSR <sup>3</sup>
Offset Error <sup>2</sup>	-0.05		-0.05		-0.1		%FSR <sup>3</sup>
Temperature Range for Guaranteed Monotonicity	25	+85	55	+125	-55	+125	°C
<b>DRIFT (T<sub>min</sub> to T<sub>max</sub>)</b>							
Total Bipolar Drift, max. includes gain, offset, and linearity drift(s)					±15	±30	ppm of FSR/°C
Total Error <sup>4</sup> - T <sub>min</sub> to T <sub>max</sub> <sup>5</sup>					-0.13	-0.30	% of FSR
Unipolar					-0.12	-0.24	% of FSR
Bipolar							
Gain					-10	±25	ppm of FSR/°C
Including Internal Reference		±10		±20		±10	ppm of FSR/°C
Excluding Internal Reference					-1	±3	ppm of FSR/°C
Unipolar Offset	-1		±2		±4		ppm of FSR/°C
Bipolar Offset		±5		±10			
<b>CONVERSION SPEED</b>							
Voltage Model (V <sup>1</sup> )							
Settling Time to ±0.01% of FSR for FSR Change (2k/100μF load)					5	5	μs
with 10kΩ Feedback	5		5		3	3	μs
with 5kΩ Feedback	3		3		1.5	1.5	μs
For LSB Change	1.5		1.5		20	20	ns
Slew Rate	20		20				μs
Current Model (I <sup>1</sup> )							
Settling Time to ±0.01% of FSR for FSR Change (10mA/100Ω Load for 1kΩ Load)	100		300		100	1	μs
for 1kΩ Load	1		1		1	1	μs
<b>ANALOG OUTPUT</b>							
Voltage Models							
Ranges - CBI	-2.5, ±5, -10, +5, +10		-2.5, ±5, -10, +5, +10		±2.5, ±5, -10, +5, +10		V
- CCD							V
Output Current	0.05		0.05		0.05		mA
Output Impedance (dc)	Indefinite to Common		Indefinite to Common		Indefinite to Common		Ω
Short Circuit Duration							
Current Models							
Ranges - Unipolar	2.0		-2.0		-2.0		mA
- Bipolar	-1.0		±1.0		±1.0		mA
Output Impedance - Bipolar	3.2		3.2		2.5 3.2 4.1		kΩ
- Unipolar	6.6		6.6		5.0 6.6 8.2		kΩ
Compliance	±6.17 ±6.1	±6.41	±6.17 ±6.3	±6.41	±6.17 ±6.3 ±6.41		V
Internal Reference Voltage (V <sub>R</sub> )	1.5		1.5		1.5		V
Max External Current <sup>6</sup>	±2.5		±2.5		±2.5		mA
Tempco of Drift	±10		±10		±5		ppm of V <sub>R</sub> /°C
<b>POWER SUPPLY SENSITIVITY</b>							
±15V ±10%, 5V supply when applicable	±0.002		±0.002		±0.002	±0.003	% of FSR/V <sub>S</sub>
<b>POWER SUPPLY REQUIREMENTS</b>							
Rated Voltages	±15, 5		±15, 5		±15, 5		V
Range							
Analog Supplies	±14.5	±15.5	±14.5	±15.5	±13.5	±16.5	V
Logic Supplies	±4.5	±15.5	±4.5	±15.5	±4.5	±15.5	V
Supply Drain							mA
+15V	15	20	15	20	10	20	mA
-15V	25	30	25	30	20	30	mA
+5V	15	20	15	20	10	20	mA
<b>TEMPERATURE RANGE</b>							
Specification	25	+85	55	+125	55	+125	°C
Operating	55	+125	55	+125	55	+125	°C
Storage	55	+125	55	+120	65	+150	°C

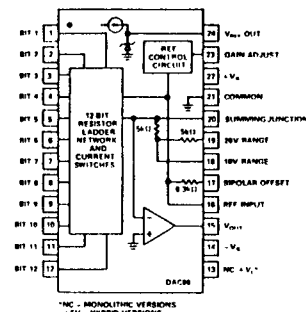
NOTES  
 1. Least Significant Bit.  
 2. Adjustable to zero with external trim potentiometer.  
 3. FSR means "Full Scale Range" and is 20V for the ±10V range and 10V for the ±5V range.  
 4. Gain and offset errors adjusted to zero at +25°C.

5. C<sub>BI</sub> = 0, see Figure 1a.  
 6. Maximum with no degradation of specification, must be a constant load including 4mA load.  
 7. Specifications subject to change without notice.

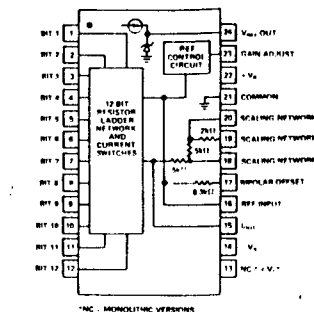
## ABSOLUTE MAXIMUM RATINGS

+V<sub>S</sub> to Power Ground . . . . . 0V to +18V  
 -V<sub>S</sub> to Power Ground . . . . . 0V to -18V  
 Digital Inputs (Pins 1 to 12) to Power Ground . . . . . -1.0V to +7V

Ref In to Reference Ground . . . . . ±12V  
 Bipolar Offset to Reference Ground . . . . . ±12V  
 10V Span R to Reference Ground . . . . . ±12V  
 20V Span R to Reference Ground . . . . . ±24V  
 Ref Out . . . . . Indefinite short to power ground or +V<sub>S</sub>



Voltage Model Functional Diagram and Pin Configuration



Current Model Functional Diagram and Pin Configuration

## ORDERING GUIDE

Model	Input Code	Output Mode	Technology	Temperature Range	Linearity Error	Package Option <sup>1</sup>
AD DAC80N-CBI-V	Binary	Voltage	Monolithic	0 to +70°C	±1.21 SB	N24A
AD DAC80D-CBI-V	Binary	Voltage	Monolithic	0 to +70°C	±1.21 SB	D24A
AD DAC80D-CBI-I	Binary	Current	Monolithic	0 to +70°C	±1.21 SB	D24A
AD DAC85D-CBI-V	Binary	Voltage	Monolithic	25°C to +85°C	±1.21 SB	D24A
AD DAC87D-CBI-V	Binary	Voltage	Monolithic	55°C to +125°C	±1.21 SB	D24A
AD DAC80-CBI-I	Binary	Voltage	Hybrid	0 to +70°C	±1.21 SB	HY24A
AD DAC80-CBI-V	Binary	Current	Hybrid	0 to +70°C	±1.21 SB	HY24A
AD DAC80-CCD-I	Binary Coded Decimal	Voltage	Hybrid	0 to +70°C	±1.41 SB	HY24A
AD DAC80-CCD-V	Binary Coded Decimal	Current	Hybrid	0 to +70°C	±1.41 SB	HY24A
AD DAC80Z-CBI-V	Binary	Voltage	Hybrid	0 to +70°C	±1.21 SB	HY24A
AD DAC80Z-CBI-I	Binary	Current	Hybrid	0 to +70°C	±1.21 SB	HY24A
AD DAC80Z-CCD-V	Binary Coded Decimal	Voltage	Hybrid	0 to +70°C	±1.41 SB	HY24A
AD DAC80Z-CCD-I	Binary Coded Decimal	Current	Hybrid	0 to +70°C	±1.41 SB	HY24A
AD DAC85C-CBI-V	Binary	Voltage	Hybrid	0 to +70°C	±1.21 SB	HY24A
AD DAC85C-CBI-I	Binary	Current	Hybrid	0 to +70°C	±1.21 SB	HY24A
AD DAC85C-CBI-V	Binary	Voltage	Hybrid	25°C to +85°C	±1.21 SB	HY24A
AD DAC85C-CBI-I	Binary	Current	Hybrid	25°C to +85°C	±1.21 SB	HY24A
AD DAC85LD-CBI-V	Binary	Voltage	Hybrid	25°C to +85°C	±1.21 SB	HY24A
AD DAC85LD-CBI-I	Binary	Current	Hybrid	25°C to +85°C	±1.21 SB	HY24A
AD DAC85MIL-CBI-V	Binary	Voltage	Hybrid	55°C to +125°C	±1.21 SB	HY24A
AD DAC85MIL-CBI-I	Binary	Current	Hybrid	55°C to +125°C	±1.21 SB	HY24A
AD DAC85C-CCD-V	Binary Coded Decimal	Voltage	Hybrid	0 to +70°C	±1.41 SB	HY24A
AD DAC85C-CCD-I	Binary Coded Decimal	Current	Hybrid	0 to +70°C	±1.41 SB	HY24A
AD DAC85-CCD-V	Binary Coded Decimal	Voltage	Hybrid	25°C to +85°C	±1.41 SB	HY24A
AD DAC85-CCD-I	Binary Coded Decimal	Current	Hybrid	25°C to +85°C	±1.41 SB	HY24A
AD DAC87-CBI-V	Binary	Voltage	Hybrid	55°C to +125°C	±1.21 SB	HY24A

<sup>1</sup> See Section 19 for package outline information.

## DIGITAL INPUT CODES

The AD DAC80 Series accepts complementary digital input code in binary (CBI) format. The CBI model may be connected by the user for anyone of three complementary codes: CSB, COB or CTC.

Digital Input		Analog Output		
MSB	LSB	CSB Compl. Straight Binary	COB Compl. Offset Binary	CTC* Compl. Two's Compl.
0 0 0 0 0 0 0 0 0 0		+ Full Scale	+ Full Scale	- 1LSB
0 1 1 1 1 1 1 1 1 1		+ 1/2 Full Scale	Zero	- Full Scale
1 0 0 0 0 0 0 0 0 0		Mid-Scale	- 1LSB	+ Full Scale
1 1 1 1 1 1 1 1 1 1		Zero	- Full Scale	Zero

\*Invert the MSB of the COB code with an external inverter to obtain CTC code.

Table 1. Digital Input Codes

## ACCURACY

Accuracy error of a D/A converter is the difference between the analog output that is expected when a given digital code is applied and the output that is actually measured with that code applied to the converter. Accuracy error can be caused by gain error, zero error, linearity error, or any combination of the three. Of these three specifications, the linearity error specification is the most important since it cannot be corrected. Linearity error is specified over its entire temperature range. This means that the analog output will not vary by more than its maximum specification, from an ideal straight line drawn between the end points (inputs all "1"s and all "0"s) over the specified temperature range.

Differential linearity error of a D/A converter is the deviation from an ideal 1LSB voltage change from one adjacent output state to the next. A differential linearity error specification of  $\pm 1/2$ LSB means that the output voltage step sizes can range from 1/2LSB to 1 1/2LSB when the input changes from one adjacent input state to the next.

## DRIFT

**Gain Drift** is a measure of the change in the full scale range output over temperature expressed in parts per million of full scale range per °C (ppm of FSR/°C). Gain drift is established by: 1) testing the end point differences for each AD DAC80 model at the lowest operating temperature; +25°C and the highest operating temperature; 2) calculating the gain error with respect to the +25°C value and; 3) dividing by the temperature change.

**Offset Drift** is a measure of the actual change in output with all "1"s on the input over the specified temperature range. The maximum change in offset is referenced to the offset at +25°C and is divided by the temperature range. This drift is expressed in parts per million of full scale range per °C (ppm of FSR/°C).

## SETTLING TIME

Settling time for each model is the total time (including slew time) required for the output to settle within an error band around its final value after a change in input.

**Voltage Output Models.** Three settling times are specified to  $\pm 0.01\%$  of full scale range (FSR); two for maximum full scale range changes of 20V, 10V and one for a 1LSB change. The

1LSB change is measured at the major carry (0 1 1 1 . . . 1 1 to 1 0 0 0 . . . 0 0), the point at which the worst case settling time occurs. The settling time characteristic depends on the compensation capacitor selected, the optimum value is 25pF as shown in Figure 1a.

**Current Output Models.** Two settling times are specified to  $\pm 0.01\%$  of FSR. Each is given for current models connected with two different resistive loads: 10 to 100 ohms and 1000 to 1875 ohms. Internal resistors are provided for connecting nominal load resistances of approximately 1000 to 1800 ohms for output voltage ranges of  $\pm 1$ V and 0 to  $-2$ V.

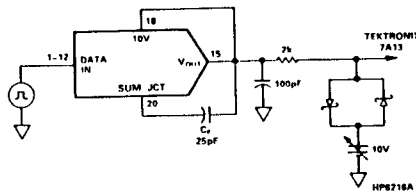


Figure 1a. Voltage Model Settling Time Circuit

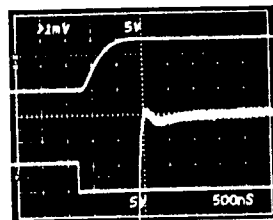


Figure 1b. Voltage Model Settling Time  $C_F = 25$ pF

## POWER SUPPLY SENSITIVITY

Power supply sensitivity is a measure of the effect of a power supply change on the D/A converter output. It is defined as a per cent of FSR per per cent of change in either the positive or negative supplies about the nominal power supply voltages.

## REFERENCE SUPPLY

All models are supplied with an internal 6.3 volt reference voltage supply. This voltage (pin 24) is accurate to  $\pm 1\%$  and must be connected to the Reference Input (pin 16) for specified operation. This reference may also be used externally with external current drain limited to 2.5mA. An external buffer amplifier is recommended if this reference is to be used to drive other system components. Otherwise, variations in the load driven by the reference will result in gain variations. All gain adjustments should be made under constant load conditions.

## ANALYZING DEVICE ACCURACY OVER THE TEMPERATURE RANGE

For the purposes of temperature drift analysis, the major device components are shown in Figure 2. The reference element and buffer amplifier drifts are combined to give the total reference temperature coefficient. The input reference current to the DAC,  $I_{REF}$ , is developed from the internal reference and will show the same drift rate as the reference voltage. The DAC output current,  $I_{DAC}$ , which is a function of the digital input codes, is designed to track  $I_{REF}$ ; if there is a slight mismatch in these currents over temperature, it will contribute to the gain T.C. The bipolar offset resistor,  $R_{BP}$ , and gain setting resistor,  $R_{GAIN}$ , also have temperature coefficients which contribute to system drift errors. The input offset voltage drift of the output amplifier, OA, also contributes a small error.

There are three types of drift errors over temperature: offset, gain, and linearity. Offset drift causes a vertical translation of the entire transfer curve; gain drift is a change in the slope of the curve; and linearity drift represents a change in the shape of the curve. The combination of these three drifts results in the complete specification for total error over temperature.

Total error is defined as the deviation from a true straight line transfer characteristic from exactly zero at a digital input which calls for zero output to a point which is defined as full scale. A specification for total error over temperature assumes that both the zero and full scale points have been trimmed for zero error at +25°C. Total error is normally expressed a percentage of the full scale range. In the bipolar situation, this means the total range from  $-V_{FS}$  to  $+V_{FS}$ .

Several new design concepts not previously used in DAC80-type devices contribute to a reduction in all the error factors over temperature. The incorporation of low temperature coefficient silicon-chromium thin-film resistors deposited on a single chip, a patented, fully differential, emitter weighted, precision current steering cell structure, and a T.C. trimmed buried zener diode reference element results in superior wide temperature range performance. The gain setting resistors and bipolar offset resistor are also fabricated on the chip with the same SiCr material as the ladder network, resulting in low gain and offset drift.

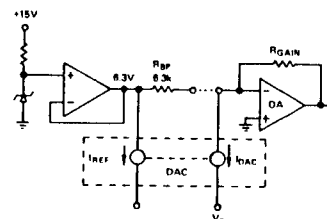


Figure 2. Bipolar Configuration

## Performance Over Temperature

### MONOTONICITY AND LINEARITY

The initial linearity error of  $\pm 1/2$ LSB max and the differential linearity error of  $\pm 3/4$ LSB max guarantee monotonic performance over the specified range. It can, therefore, be assumed that linearity errors are insignificant in computation of total temperature errors.

### UNIPOLAR ERRORS

Temperature error analysis in the unipolar mode is straightforward: there is an offset drift and a gain drift. The offset drift (which comes from leakage currents and drift in the output amplifier (OA)) causes a linear shift in the transfer curve as shown in Figure 3. The gain drift causes a change in the slope of the curve and results from reference drift, DAC drift, and drift in  $R_{GAIN}$  relative to the DAC resistors.

### BIPOLAR RANGE ERRORS

The analysis is slightly more complex in the bipolar mode. In this mode  $R_{BP}$  is connected to the summing node of the output amplifier (see Figure 2) to generate a current which, exactly balances the current of the MSB so that the output voltage is zero with only the MSB on.

Note that if the DAC and application resistors track perfectly, the bipolar offset drift will be zero even if the reference drifts. A change in the reference voltage, which causes a shift in the bipolar offset, will also cause an equivalent change in  $I_{REF}$  and thus  $I_{DAC}$ , so that  $I_{DAC}$  will always be exactly balanced by  $I_{BP}$  with the MSB turned on. This effect is shown in Figure 3. The net effect of the reference drift then is simply to cause a rotation in the transfer around bipolar zero. However, consideration of second order effects (which are often overlooked) reveals the errors in the bipolar mode. The unipolar offset drifts discussed before will have the same effect on the bipolar offset. A mismatch of  $R_{BP}$  to the DAC resistors is usually the largest component of bipolar drift, but in the AD DAC80 this error is held to 10ppm/°C max. Gain drift in the DAC also contributes to bipolar offset drift, as well as full scale drift, but again is held to 10ppm/°C max.

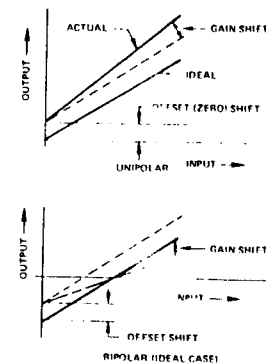


Figure 3. Unipolar and Bipolar Drifts

# Using the AD DAC80 Series

## POWER SUPPLY CONNECTIONS

For optimum performance power supply decoupling capacitors should be added as shown in the connection diagrams. These capacitors (1 $\mu$ F electrolytic recommended) should be located close to the AD DAC80. Electrolytic capacitors, if used, should be paralleled with 0.01 $\mu$ F ceramic capacitors for optimum high frequency performance.

## EXTERNAL OFFSET AND GAIN ADJUSTMENT

Offset and gain may be trimmed by installing external OFFSET and GAIN potentiometers. These potentiometers should be connected as shown in the block diagrams and adjusted as described below. TCR of the potentiometers should be 100ppm/°C or less. The 3.9M $\Omega$  and 10M $\Omega$  resistors (20% carbon or better) should be located close to the AD DAC80 to prevent noise pickup. If it

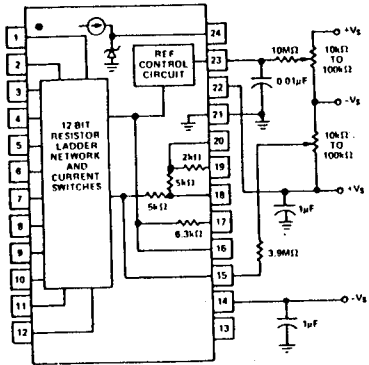


Figure 4. External Adjustment and Voltage Supply Connection Diagram, Current Model

is not convenient to use these high-value resistors, a functionally equivalent "T" network, as shown in Figure 6 may be substituted in each case. The gain adjust (pin 23) is a high impedance point and a 0.01 $\mu$ F ceramic capacitor should be connected from this pin to common to prevent noise pickup.

**Offset Adjustment.** For unipolar (CSB) configurations, apply the digital input code that should produce zero potential output and adjust the OFFSET potentiometer for zero output. For bipolar (COB, CTC) configurations, apply the digital input code that should produce the maximum negative output voltage. Example: If the FULL SCALE RANGE is connected for 20 volts, the maximum negative output voltage is -10V. See Table II for corresponding codes.

**Gain Adjustment.** For either unipolar or bipolar configurations, apply the digital input that should give the maximum positive voltage output. Adjust the GAIN potentiometer for this positive full scale voltage. See Table II for positive full scale voltages.

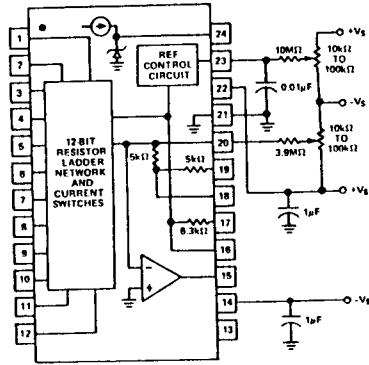


Figure 5. External Adjustment and Voltage Supply Connection Diagram, Voltage Model

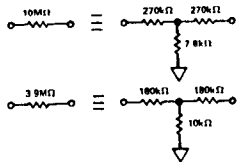


Figure 6. Equivalent Resistances

Digital Input		Analog Output			
		Voltage*		Current	
MSB	LSB	0 to +10V	$\pm 10V$	0 to -2mA	$\pm 1mA$
CBI Model	0 0 0 0 0 0 0 0 0 0 0 0	+ 9.9976V	+ 9.9951V	- 1.9995mA	- 0.9995mA
	0 1 1 1 1 1 1 1 1 1 1 1	+ 5.0000V	0.0000V	- 1.0000mA	0.0000mA
	1 0 0 0 0 0 0 0 0 0 0 0	+ 4.9976V	4.88mV	0.488mA	+ 1.000mA
	1 1 1 1 1 1 1 1 1 1 1 1	0.0000V	- 10.0000V	0.0000mA	0.488 $\mu$ A
ILSB		2.44mV	- 0.0049V	- 0.9995mA	+ 0.0005mA

\* To obtain values for other binary ranges 0 to +5V range: divide 0 to +10 values by 2;  
- 5V range: divide +10V range values by 2; + 2.5V range: divide +10V range values by 4.

Table II. Digital Input/Analog Output

## VOLTAGE OUTPUT MODELS

Internal scaling resistors provided in the AD DAC80 may be connected to produce bipolar output voltage ranges of  $\pm 10$ ,  $\pm 5$  or  $\pm 2.5V$  or unipolar output voltage ranges of 0 to +5 or 0 to +10V (see Figure 7).

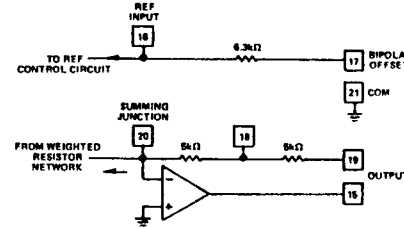


Figure 7. Output Amplifier Voltage Range Scaling Circuit

Gain and offset drift are minimized in the AD DAC80 because of the thermal tracking of the scaling resistors with other device components. Connections for various output voltage ranges are shown in Table III. Settling time is specified for a full scale range change: 4 microseconds for a 10k $\Omega$  feedback resistor; 3 microseconds for a 5k $\Omega$  feedback resistor when using the compensation capacitor shown in Figure 1.

The equivalent resistive scaling network and output circuit of the current model are shown in Figures 8 and 9. External  $R_{LS}$  resistors are required to produce exactly 0 to -2V or  $\pm 1V$  output. TCR of these resistors should be  $\pm 100ppm/^\circ C$  or less to maintain the AD DAC80 output specifications. If exact output ranges are not required, the external resistors are not needed.

Output Range	Digital Input Codes	Connect Pin 15 to	Connect Pin 17 to	Connect Pin 19 to	Connect Pin 16 to
$\pm 10V$	COB or CTC	19	20	15	24
$\pm 5V$	COB or CTC	18	20	N.C.	24
$\pm 2.5V$	COB or CTC	18	20	20	24
0 to +10V	CSB	18	21	N.C.	24
0 to +5V	CSB	18	21	20	24
0 to +10V	CCD	19	N.C.	15	24

Table III. Output Voltage Range Connections-Voltage Model AD DAC80

Digital Input Codes	Output Range	Internal Resistance $R_{LI}$	1% Metal Film External Resistance $R_{LS}$	$R_{LI}$ Connections			Reference Connect Pin 16 to	Bipolar Offset	
				Connect Pin 15 to	Connect Pin 18 to	Connect Pin 20 to		Connect Pin 17 to	$R_{LS}$
CSB	0 to -2V	0.968k $\Omega$	210 $\Omega$	20	19 & $R_{LS}$	15	24	Com (21)	Between Pin 18 & Com (21)
COB or CTC	$\pm 1V$	1.2k $\Omega$	249 $\Omega$	18	19	$R_{LS}$	24	15	Between Pin 20 & Com (21)
CCD	0 to $\pm 2V$	3k $\Omega$	N/A	N.C.	21	N.C.	24	N.C.	N/A

Table IV. Current Model Resistive Load Connections

## Applying the AD DAC80

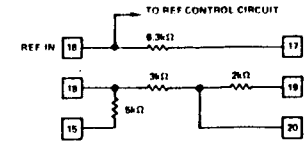


Figure 8. Internal Scaling Resistors

Internal resistors are provided to scale an external op amp or to configure a resistive load to offer two output voltage ranges of  $\pm 1V$  or 0 to -2V. These resistors ( $R_{LI}$ ; TCR = 20ppm/°C) are an integral part of the AD DAC80 and maintain gain and bipolar offset drift specifications. If the internal resistors are bypassed, external  $R_{LI}$  (or  $R_{LF}$ ) resistors should have a TCR of  $\pm 25ppm/^\circ C$  or less to minimize drift. This will typically add  $\pm 50ppm/^\circ C$  to the TCR of  $R_{LI}$  (or  $R_{LF}$ ) to the total drift.

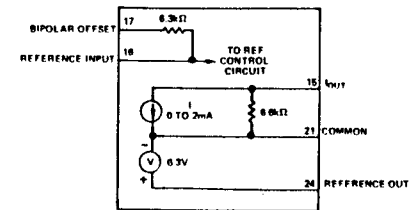


Figure 9. AD DAC80 Current Model Equivalent Output Circuit



### DRIVING A RESISTIVE LOAD UNIPOLAR

A load resistance,  $R_L = R_{L1} + R_{L2}$ , connected as shown in Figure 10 will generate a voltage range,  $V_{OUT}$ , determined by:

$$V_{OUT} = -2\text{mA} \left( \frac{6.6\text{k} \times R_L}{6.6\text{k} + R_L} \right)$$

Where  $R_{L \text{ max}} = 1.54\text{k}\Omega$   
and  $V_{OUT \text{ max}} = -2.5\text{V}$

To achieve specified drift, connect the internal scaling resistor ( $R_{L1}$ ) as shown in Table IV to an external metal film trim resistor ( $R_{L2}$ ) to provide full scale output voltage range of 0 to  $-2\text{V}$ . With  $R_{L2} = 0$ ,  $V_{OUT} = -1.69\text{V}$ .

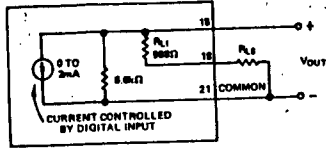


Figure 10. Equivalent Circuit AD DAC80-CBI-I Connected for Unipolar Voltage Output with Resistive Load

### DRIVING A RESISTOR LOAD BIPOLAR

The equivalent output circuit for a bipolar output voltage range is shown in Figure 11,  $R_L = R_{L1} + R_{L2}$ .  $V_{OUT}$  is determined by:

$$V_{OUT} = \pm 1\text{mA} \left( \frac{R_L \times 3.22\text{k}}{R_L + 3.22\text{k}} \right)$$

Where  $R_{L \text{ max}} = 11.18\text{k}\Omega$   
and  $V_{OUT \text{ max}} = \pm 2.5\text{V}$

To achieve specified drift, connect the internal scaling resistors ( $R_{L1}$ ) as shown in Table IV for the COB or CTC codes and add an external metal film resistor ( $R_{L2}$ ) in series to obtain a full scale output range of  $\pm 1\text{V}$ . In this configuration, with  $R_{L2}$  equal to zero, the full scale range will be  $\pm 0.874\text{V}$ .

### DRIVING AN EXTERNAL OP AMP

The current model AD DAC80 will drive the summing junction of an op amp used as a current to voltage converter to produce an output voltage. As seen in Figure 12,

$$V_{OUT} = I_{OUT} \times R_F$$

where  $I_{OUT}$  is the AD DAC80 output current and  $R_F$  is the feedback resistor. Using the internal feedback resistors of the current model AD DAC80 provides output voltage ranges the

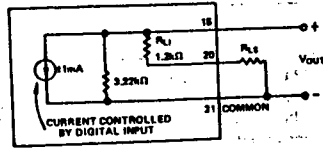


Figure 11. AD DAC80-CBI-I Connected for Bipolar Output Voltage with Resistive Load

same as the voltage model AD DAC80. To obtain the desired output voltage range when connecting an external op amp, refer to Table V and Figure 12.

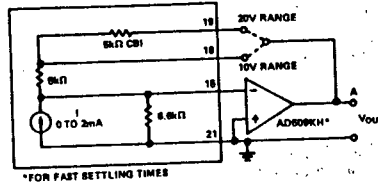


Figure 12. External Op Amp—Using Internal Feedback Resistors

### OUTPUT LARGER THAN 20V RANGE

For output voltage ranges larger than  $\pm 10$  volts, a high voltage op amp may be employed with an external feedback resistor. Use  $I_{OUT}$  values of  $\pm 1\text{mA}$  for bipolar voltage ranges and  $-2\text{mA}$  for unipolar voltage ranges (see Figure 13). Use protection diodes when a high voltage op amp is used.

The feedback resistor,  $R_F$ , should have a temperature coefficient as low as possible. Using an external feedback resistor, overall drift of the circuit increases due to the lack of temperature tracking between  $R_F$  and the internal scaling resistor network. This will typically add 50ppm/°C +  $R_F$  drift to total drift.

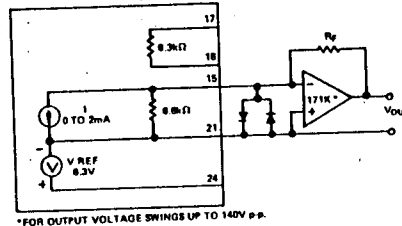


Figure 13. External Op Amp—Using External Feedback Resistors

Output Range	Digital Input Codes	Connect A to	Connect Pin 17 to	Connect Pin 19 to	Connect Pin 16 to
$\pm 10\text{V}$	COB or CTC	19	15	A	24
$\pm 5\text{V}$	COB or CTC	18	15	N.C.	24
$\pm 2.5\text{V}$	COB or CTC	18	15	15	24
0 to +10V	CSB	18	21	N.C.	24
0 to +5V	CSB	18	21	15	24

Table V. External Op Amp Voltage Mode Connections

**ANALOG DEVICES**

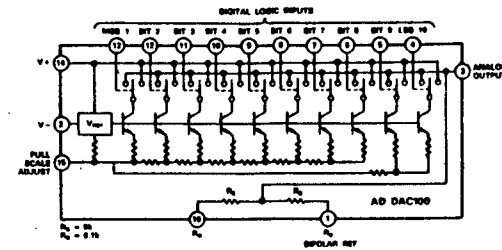
**10-Bit Monolithic D/A Converter**

**AD DAC100**

### FEATURES

- Complete Current Output Converter
- High Stability Buried Zener Reference
- Single Chip Monolithic Construction
- Wide Supply Range  $\pm 6\text{V}$  to  $\pm 18\text{V}$
- Trimmed Output Application Resistors
- Fast Settling — 225ns (8 Bits), 375ns (10 Bits)
- Guaranteed Monotonicity Over Full Operating Temperature Range
- TTL and DTL Compatible Logic Inputs
- Hermetically-Sealed 16-Pin Ceramic DIP (All Grades)

### AD DAC100 FUNCTIONAL BLOCK DIAGRAM



### PRODUCT DESCRIPTION

The AD DAC100 is a 10-bit digital-to-analog converter with a high stability voltage reference fabricated on a monolithic chip. Using 10 precision high-speed current-steering switches, a control amplifier, voltage reference, and laser-trimmed thin-film SiCr resistor network, the device produces a fast, accurate analog output current. Laser trimmed output application resistors are also included to facilitate accurate, stable current-to-voltage conversion.

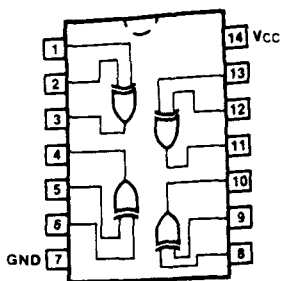
The AD DAC100 also incorporates a low noise, high stability subsurface zener diode to produce a reference voltage with excellent long term stability and temperature cycle characteristics which challenge the best discrete zener references. A temperature compensation circuit is laser-trimmed to allow correction of the temperature coefficient of each device. This results in a maximum full-scale temperature coefficient of 15ppm/°C for the L version, 30ppm/°C max for the K and T versions, 60ppm/°C max for the J and S versions.

All grades are packaged in a 16-pin hermetically-sealed ceramic dual-in-line package. The J, K, L versions are specified for operation over the  $-25^\circ\text{C}$  to  $+85^\circ\text{C}$  temperature range, the AD DAC100S and T for operation over the extended temperature range from  $-55^\circ\text{C}$  to  $+125^\circ\text{C}$ .

### PRODUCT HIGHLIGHTS

- The AD DAC100 is a true second-source equivalent to the industry standard DAC100.
- The high impedance current output can be used with an external op amp and the internal applications resistors to produce a low impedance output voltage.
- The AD DAC100 is available with a 10 volt range; for 5 volt version, consult factory.
- The AD DAC100 is available in chip form for use in hybrid microcircuits.

CONNECTION DIAGRAM  
PINOUT A



TRUTH TABLE

INPUTS		OUTPUT
A	B	Z
L	L	H
L	H	L
H	L	L
H	H	H

H = HIGH Voltage Level  
L = LOW Voltage Level

CONDITIONS

Vcc = Max  
Other Input LOW  
Figs. 3-2, 3-5  
Other Input HIGH  
Figs. 3-2, 3-5

## 54LS/74LS273

### 8-BIT REGISTER (With Clear)

**DESCRIPTION** — The '273 is a high speed 8-bit register, consisting of eight D-type flip-flops with a common Clock and an asynchronous active LOW Master Reset. This device is supplied in a 20-pin package featuring 0.3 inch row spacing.

- EDGE-TRIGGERED
- 8-BIT HIGH SPEED REGISTER
- PARALLEL IN AND OUT
- COMMON CLOCK AND MASTER RESET

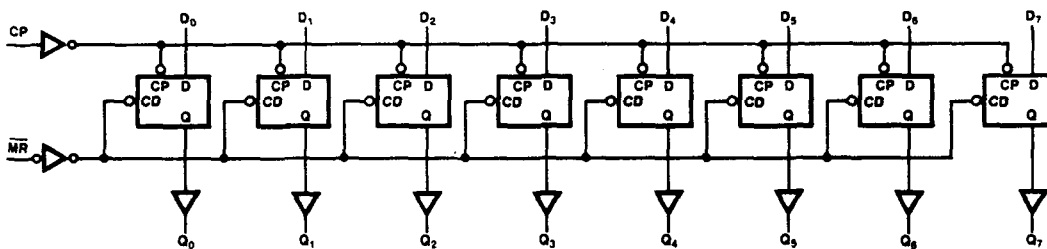
ORDERING CODE: See Section 9

PKGS	PIN OUT	COMMERCIAL GRADE	MILITARY GRADE	PKG TYPE
		Vcc = +5.0 V ±5%, TA = 0°C to +70°C	Vcc = +5.0 V ±10%, TA = -55°C to +125°C	
Plastic DIP (P)	A	74LS273PC		9Z
Ceramic DIP (D)	A	74LS273DC	54LS273DM	4E
Flatpak (F)	A	74LS273FC	54LS273FM	4F

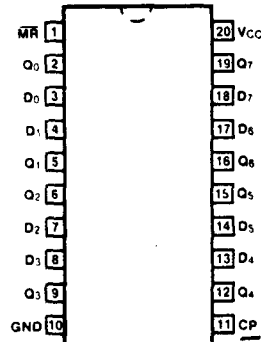
INPUT LOADING/FAN-OUT: See Section 3 for U.L. definitions

PIN NAMES	DESCRIPTION	54/74LS (U.L.) HIGH/LOW
CP	Clock Pulse Input (Active Rising Edge)	0.5/0.25
D <sub>0</sub> — D <sub>7</sub>	Data Inputs	0.5/0.25
MR	Asynchronous Master Reset Input (Active LOW)	0.5/0.25
Q <sub>0</sub> — Q <sub>7</sub>	Flip-flop Outputs	10/5.0 (2.5)

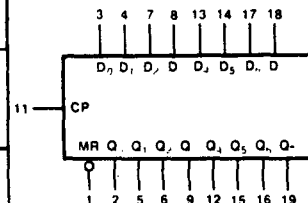
LOGIC DIAGRAM



CONNECTION DIAGRAM  
PINOUT A



LOGIC SYMBOL



Vcc = Pin 20  
GND = Pin 10

# 54S/74S138

## 54LS/74LS138

### 1-OF-8 DECODER/DEMULTIPLEXER

**DESCRIPTION** — The '138 is a high speed 1-of-8 decoder/demultiplexer. This device is ideally suited for high speed bipolar memory chip select address decoding. The multiple input enables allow parallel expansion to a 1-of-24 decoder using just three '138 devices or to a 1-of-32 decoder using four '138 devices and one inverter. The '138 is fabricated with the Schottky barrier diode process for high speed.

- SCHOTTKY PROCESS FOR HIGH SPEED
- DEMULTIPLEXING CAPABILITY
- MULTIPLE INPUT ENABLE FOR EASY EXPANSION
- ACTIVE LOW MUTUALLY EXCLUSIVE OUTPUTS

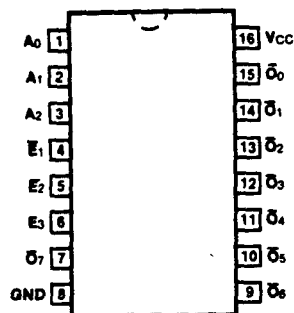
**ORDERING CODE:** See Section 9

PKGS	PIN OUT	COMMERCIAL GRADE	MILITARY GRADE	PKG TYPE
		VCC = +5.0 V ±5%, TA = 0°C to +70°C	VCC = +5.0 V ±10%, TA = -55°C to +125°C	
Plastic DIP (P)	A	74S138PC, 74LS138PC		9B
Ceramic DIP (D)	A	74S138DC, 74LS138DC	54S138DM, 54LS138DM	6B
Flatpak (F)	A	74S138FC, 74LS138FC	54S138FM, 54LS138FM	4L

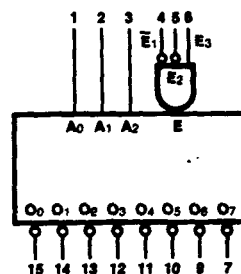
**INPUT LOADING/FAN-OUT:** See Section 3 for U.L. definitions

PIN NAMES	DESCRIPTION	54/74S (U.L.) HIGH/LOW	54/74LS (U.L.) HIGH/LOW
A <sub>0</sub> - A <sub>2</sub>	Address Inputs	1.25/1.25	0.5/0.25
$\bar{E}_1, \bar{E}_2$	Enable Inputs (Active LOW)	1.25/1.25	0.5/0.25
E <sub>3</sub>	Enable Input (Active HIGH)	1.25/1.25	0.5/0.25
$\bar{O}_0 - \bar{O}_7$	Outputs (Active LOW)	25/12.5	10/5.0 (2.5)

#### CONNECTION DIAGRAM PINOUT A



#### LOGIC SYMBOL



VCC = Pin 16  
GND = Pin 8

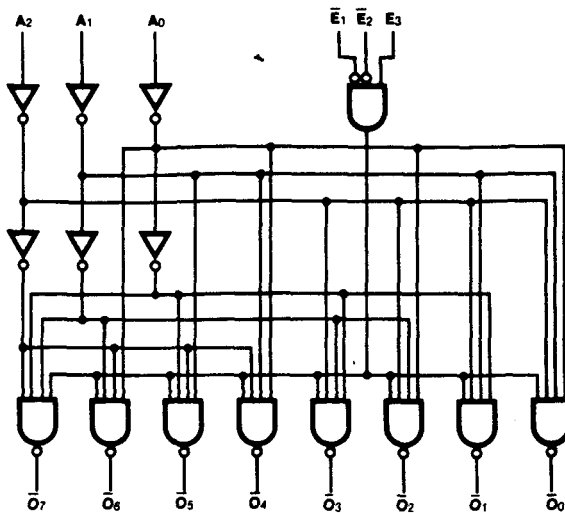
**FUNCTIONAL DESCRIPTION** — The '138 is a high speed 1-of-8 decoder/demultiplexer fabricated with the low power Schottky barrier diode process. The decoder accepts three binary weighted inputs ( $A_0$ ,  $A_1$ ,  $A_2$ ) and when enabled provides eight mutually exclusive active LOW outputs ( $\bar{O}_0$  —  $\bar{O}_7$ ). The '138 features three Enable inputs, two active LOW ( $\bar{E}_1$ ,  $\bar{E}_2$ ) and one active HIGH ( $E_3$ ). All outputs will be HIGH unless  $\bar{E}_1$  and  $\bar{E}_2$  are LOW and  $E_3$  is HIGH. This multiple enable function allows easy parallel expansion of the device to a 1-of-32 (5 lines to 32 lines) decoder with just four '138 devices and one inverter. (See Figure a.) The '138 can be used as an 8-output demultiplexer by using one of the active LOW Enable inputs as the data input and the other Enable inputs as strobes. The Enable inputs which are not used must be permanently tied to their appropriate active HIGH or active LOW state.

TRUTH TABLE

INPUTS						OUTPUTS							
$\bar{E}_1$	$\bar{E}_2$	$E_3$	$A_0$	$A_1$	$A_2$	$\bar{O}_0$	$\bar{O}_1$	$\bar{O}_2$	$\bar{O}_3$	$\bar{O}_4$	$\bar{O}_5$	$\bar{O}_6$	$\bar{O}_7$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	L	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

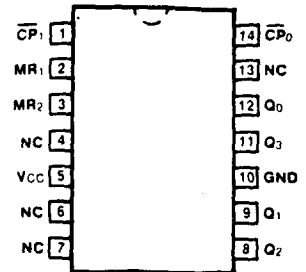
H = HIGH Voltage Level  
L = LOW Voltage Level  
X = Immaterial

LOGIC DIAGRAM



**54/7493A**  
**54LS/74LS93**  
 DIVIDE-BY-SIXTEEN COUNTER

**CONNECTION DIAGRAM**  
**PINOUT A**

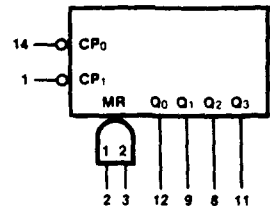


**DESCRIPTION** — The '93 is a 4-stage ripple counter containing a high speed flip-flop acting as a divide-by-two and three flip-flops connected as a divide-by-eight. HIGH signals on the Master Reset (MR) inputs override the clocks and force all outputs to the LOW state.

**ORDERING CODE:** See Section 9

PKGS	PIN OUT	COMMERCIAL GRADE	MILITARY GRADE	PKG TYPE
		V <sub>CC</sub> = +5.0 V ±5%, T <sub>A</sub> = 0° C to +70° C	V <sub>CC</sub> = +5.0 V ±10%, T <sub>A</sub> = -55° C to +125° C	
Plastic DIP (P)	A	7493APC, 74LS93PC		9A
Ceramic DIP (D)	A	7493ADC, 74LS93DC	5493ADM, 54LS93DM	6A
Flatpak (F)	A	7493AFC, 74LS93FC	5493AFM, 54LS93FM	3I

**LOGIC SYMBOL**



V<sub>CC</sub> = Pin 5  
 GND = Pin 10  
 NC = Pins 4, 6, 7, 13

**INPUT LOADING/FAN-OUT:** See Section 3 for U.L. definitions

PIN NAMES	DESCRIPTION	54/74 (U.L.) HIGH/LOW	54/74LS (U.L.) HIGH/LOW*
CP <sub>0</sub>	+2 Section Clock Input (Active Falling Edge)	2.0/2.0	1.0/1.5
CP <sub>1</sub>	+5 Section Clock Input (Active Falling Edge)	2.0/2.0	1.0/1.0
MR <sub>1</sub> , MR <sub>2</sub>	Asynchronous Master Reset Inputs (Active HIGH)	1.0/1.0	0.5/0.25
Q <sub>0</sub>	+2 Section Output*	20/10	10/5.0 (2.5)
Q <sub>1</sub> — Q <sub>3</sub>	+8 Section Outputs	20/10	10/5.0 (2.5)

\*The Q<sub>0</sub> output is guaranteed to drive the full rated fan-out plus the CP<sub>1</sub> input.

**FUNCTIONAL DESCRIPTION** — The '93 is a 4-bit ripple type binary counter. It consists of four master/slave flip-flops which are internally connected to provide a divide-by-two section and a divide-by-eight section. Each section has a separate clock input which initiates state changes of the counter on the HIGH-to-LOW clock transition. State changes of the Q outputs do not occur simultaneously because of internal ripple delays. Therefore, decoded output signals are subject to decoding spikes and should not be used for clocks or strobes. The Q<sub>0</sub> output of each device is designed and specified to drive the rated fan-out plus the  $\overline{CP}_1$  input of the device. A gated AND asynchronous Master Reset (MR<sub>1</sub>, MR<sub>2</sub>) is provided which overrides the clocks and resets (clears) all the flip-flops. Since the output from the divide-by-two section is not internally connected to the succeeding stages, the devices may be operated in various counting modes.

- A. 4-Bit Ripple Counter — The output Q<sub>0</sub> must be externally connected to input  $\overline{CP}_1$ . The input count pulses are applied to input  $\overline{CP}_0$ . Simultaneous divisions of 2, 4, 8, and 16 are performed at the Q<sub>0</sub>, Q<sub>1</sub>, Q<sub>2</sub>, and Q<sub>3</sub> outputs as shown in the Truth Table.
- B. 3-Bit Ripple Counter — The input count pulses are applied to input  $\overline{CP}_1$ . Simultaneous frequency divisions of 2, 4, and 8 are available at the Q<sub>1</sub>, Q<sub>2</sub>, and Q<sub>3</sub> outputs. Independent use of the first flip-flop is available if the reset function coincides with reset of the 3-bit ripple-through counter.

**MODE SELECTION**

RESET INPUTS		OUTPUTS			
MR <sub>1</sub>	MR <sub>2</sub>	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>
H	H	L	L	L	L
L	H	Count			
H	L	Count			
L	L	Count			

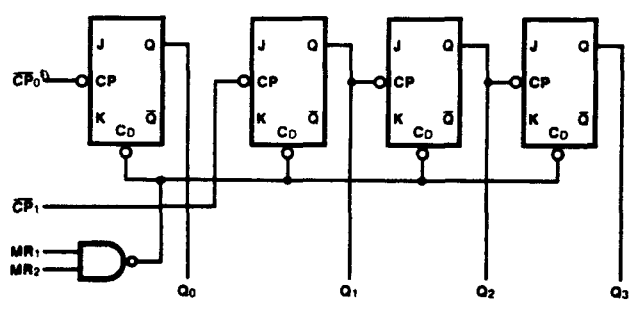
H = HIGH Voltage Level  
L = LOW Voltage Level

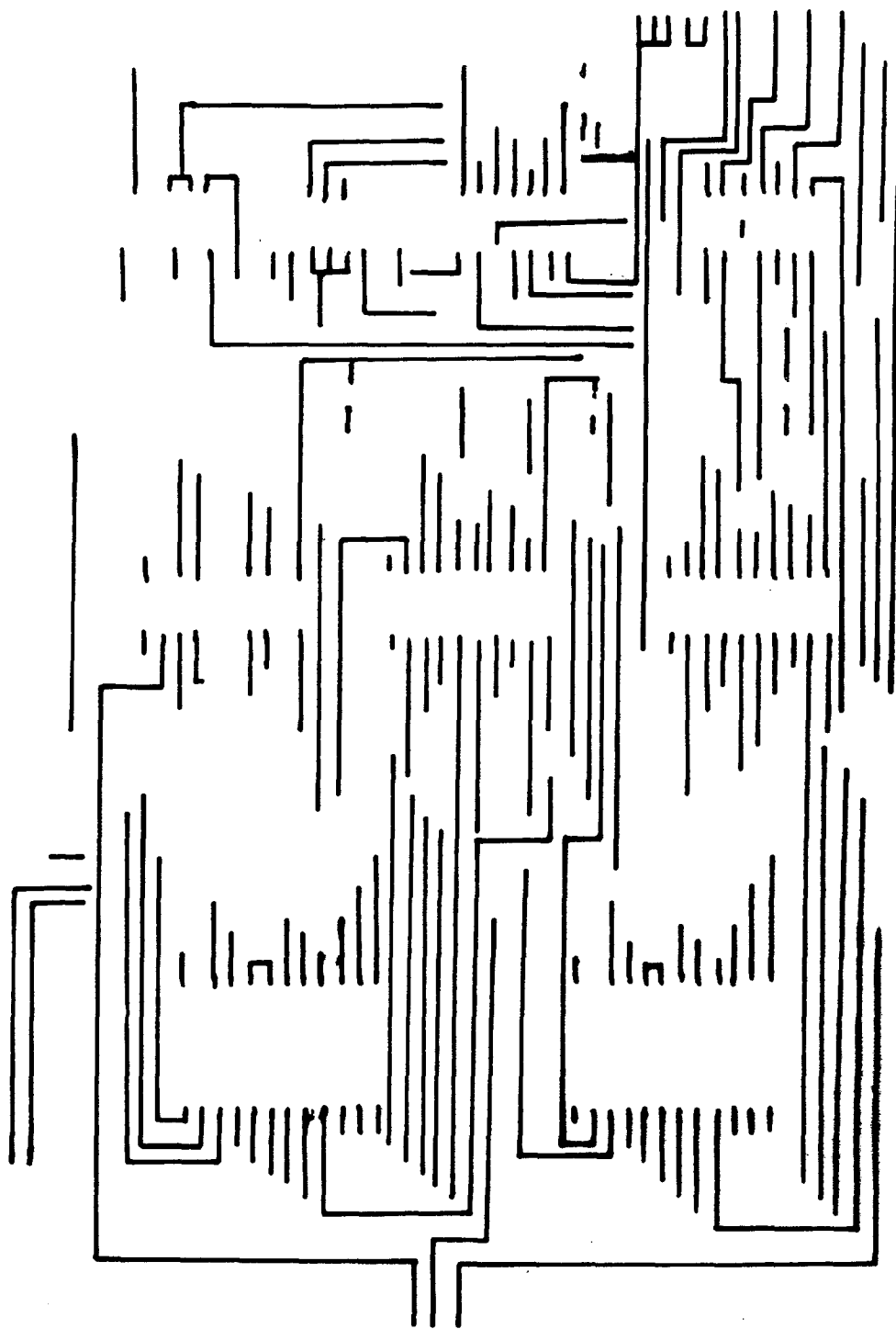
**TRUTH TABLE**

COUNT	OUTPUTS			
	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>
0	L	L	L	L
1	H	L	L	L
2	L	H	L	L
3	H	H	L	L
4	L	L	H	L
5	H	L	H	L
6	L	H	H	L
7	H	H	H	L
8	L	L	L	H
9	H	L	L	H
10	L	H	L	H
11	H	H	L	H
12	L	L	H	H
13	H	L	H	H
14	L	H	H	H
15	H	H	H	H

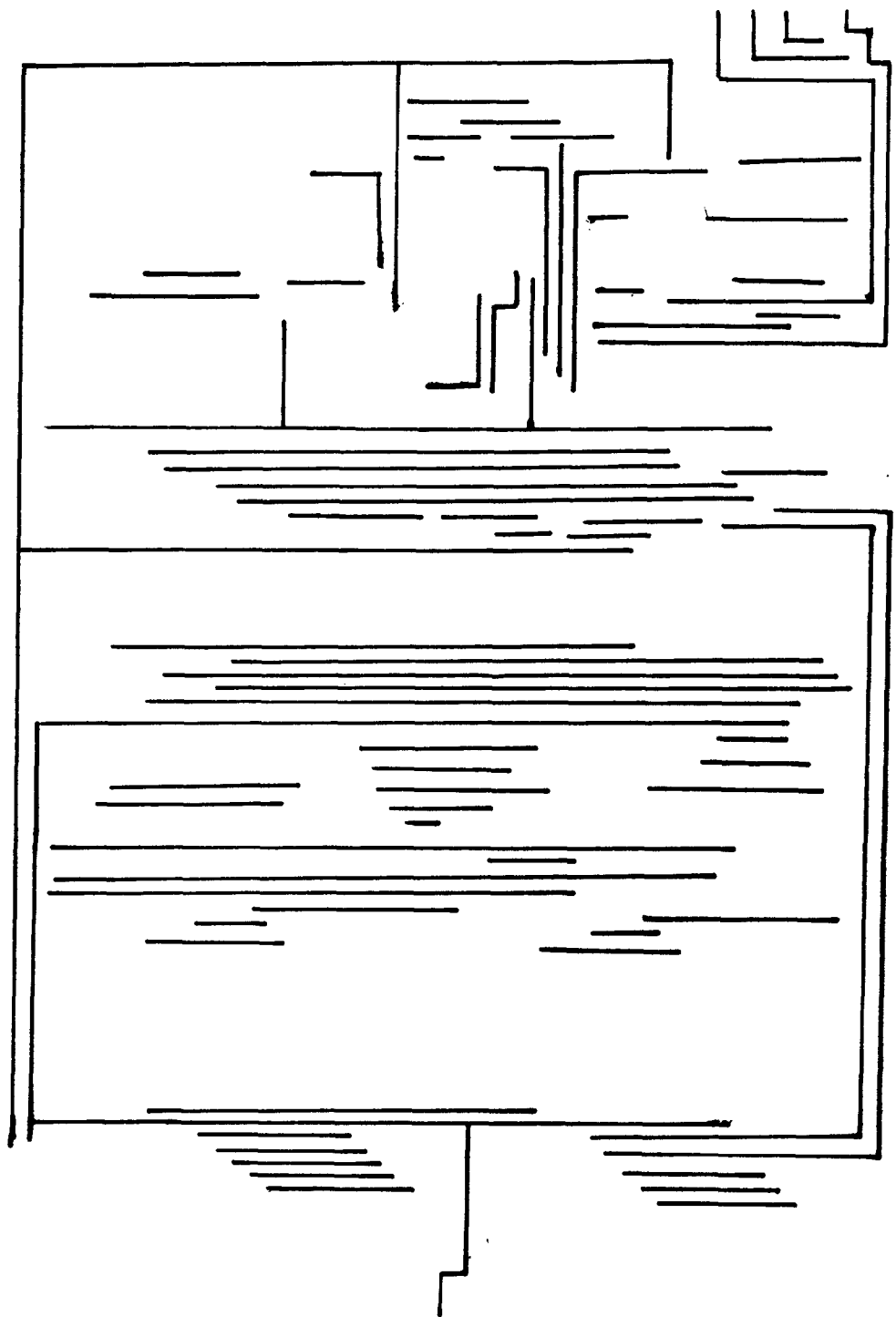
NOTE: Output Q<sub>0</sub> connected to  $\overline{CP}_1$ .

**LOGIC DIAGRAM**



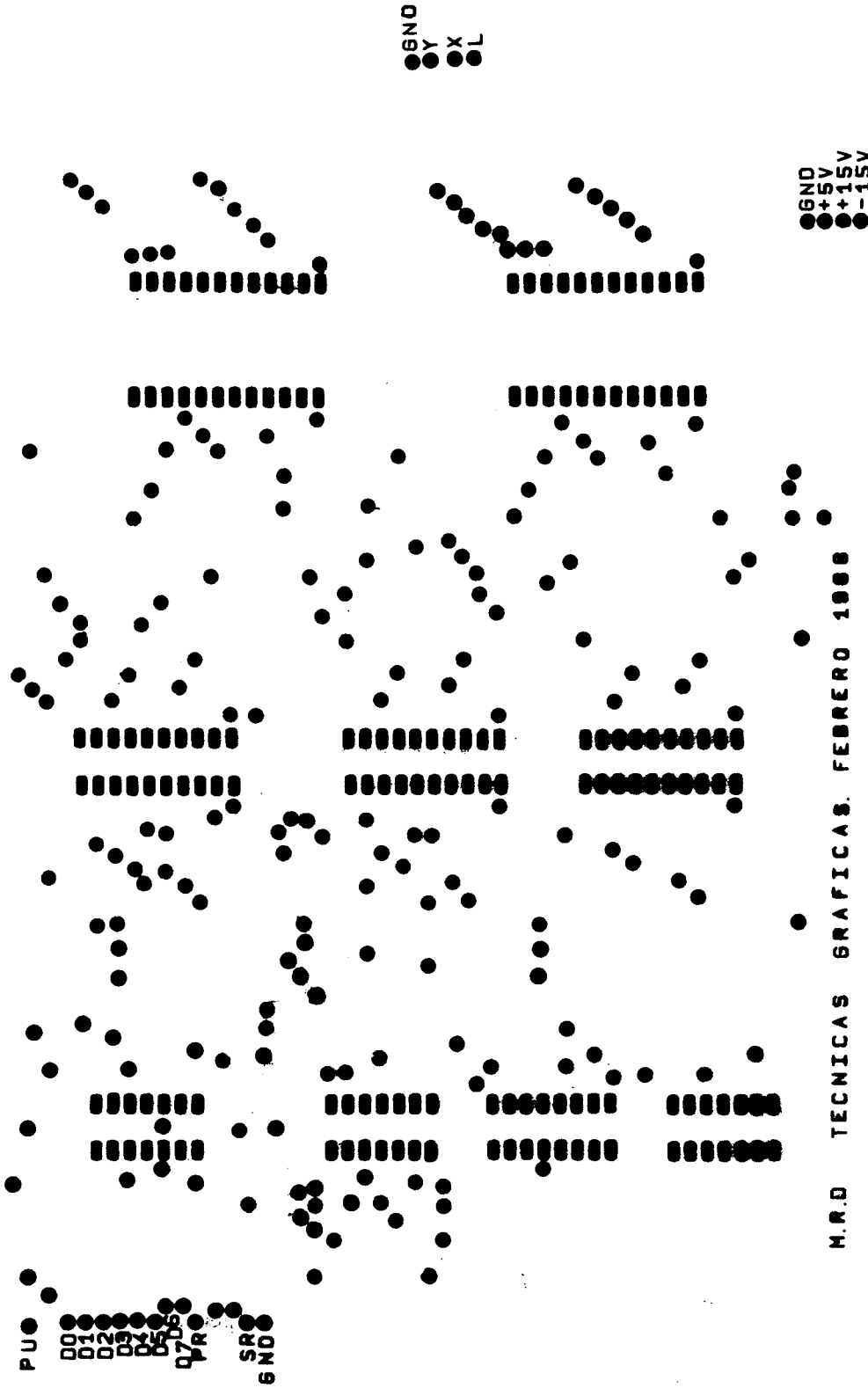


PLANO DE CIRCUITO IMPRESO (CARA SUPERIOR)



PLANO DE CIRCUITO IMPRESO (CARA SUPERIOR)



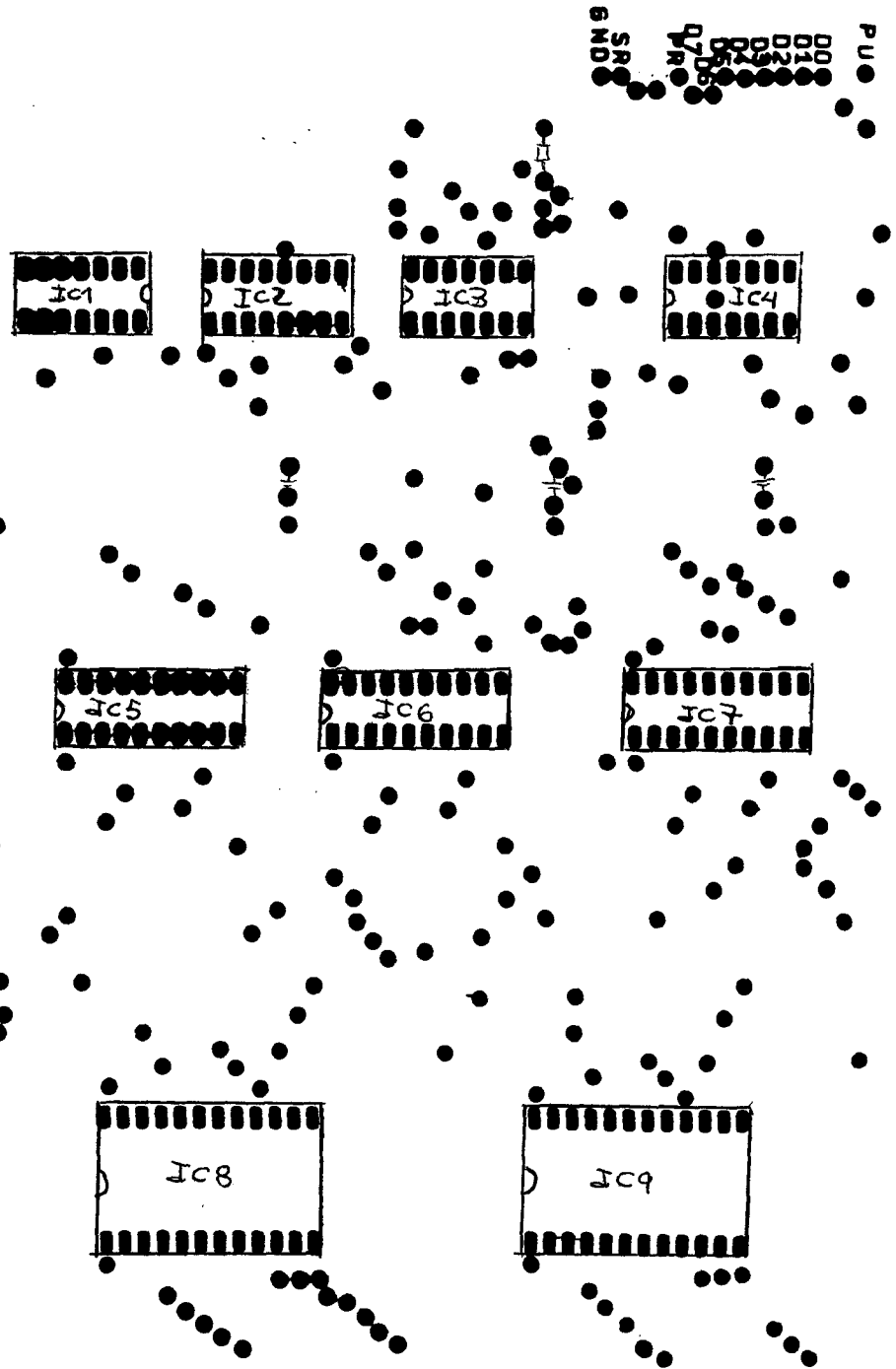


M.R.D. TÉCNICAS GRÁFICAS. FEBRERO 1988

● GND  
● +5V  
● +15V  
● -15V

● GND  
● Y  
● X  
● L

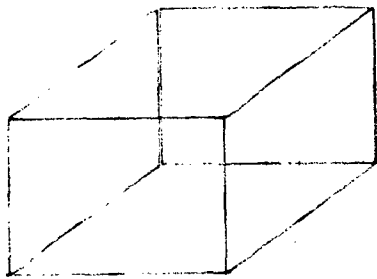
PLANO DE MONTAGE



DIBUJOS

Y

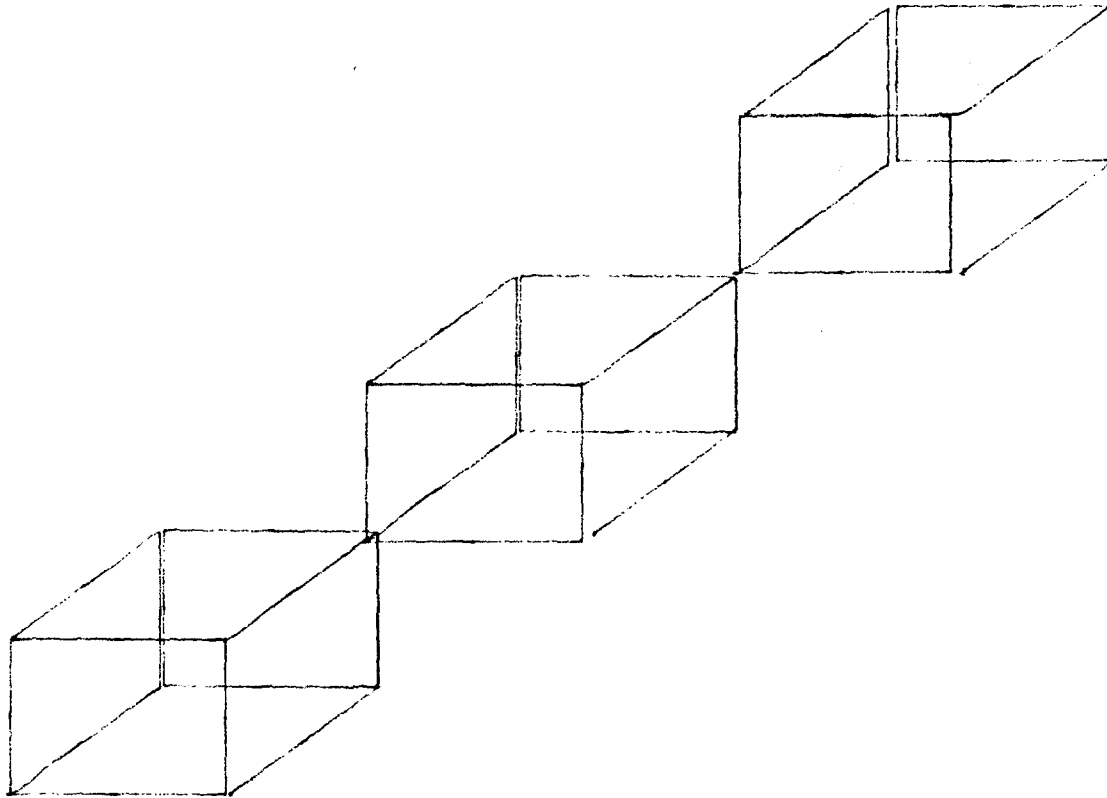
EXDGOVO (FIGURA BASICA)



X

y

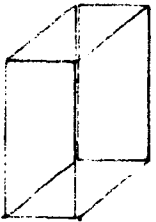
EXAGONO (TRASLACION)



x

Y

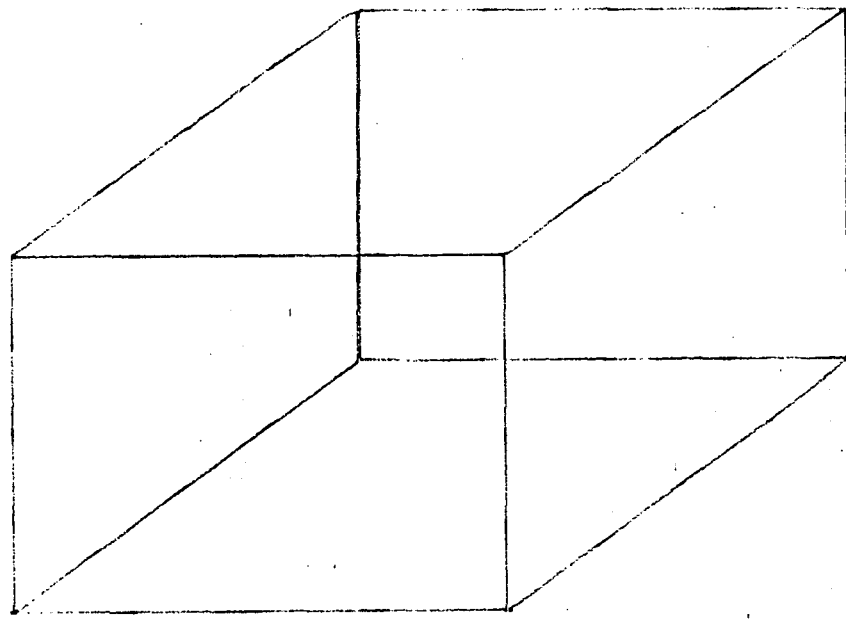
EXAGONO (ESTRECHAMIENTO)



X

y

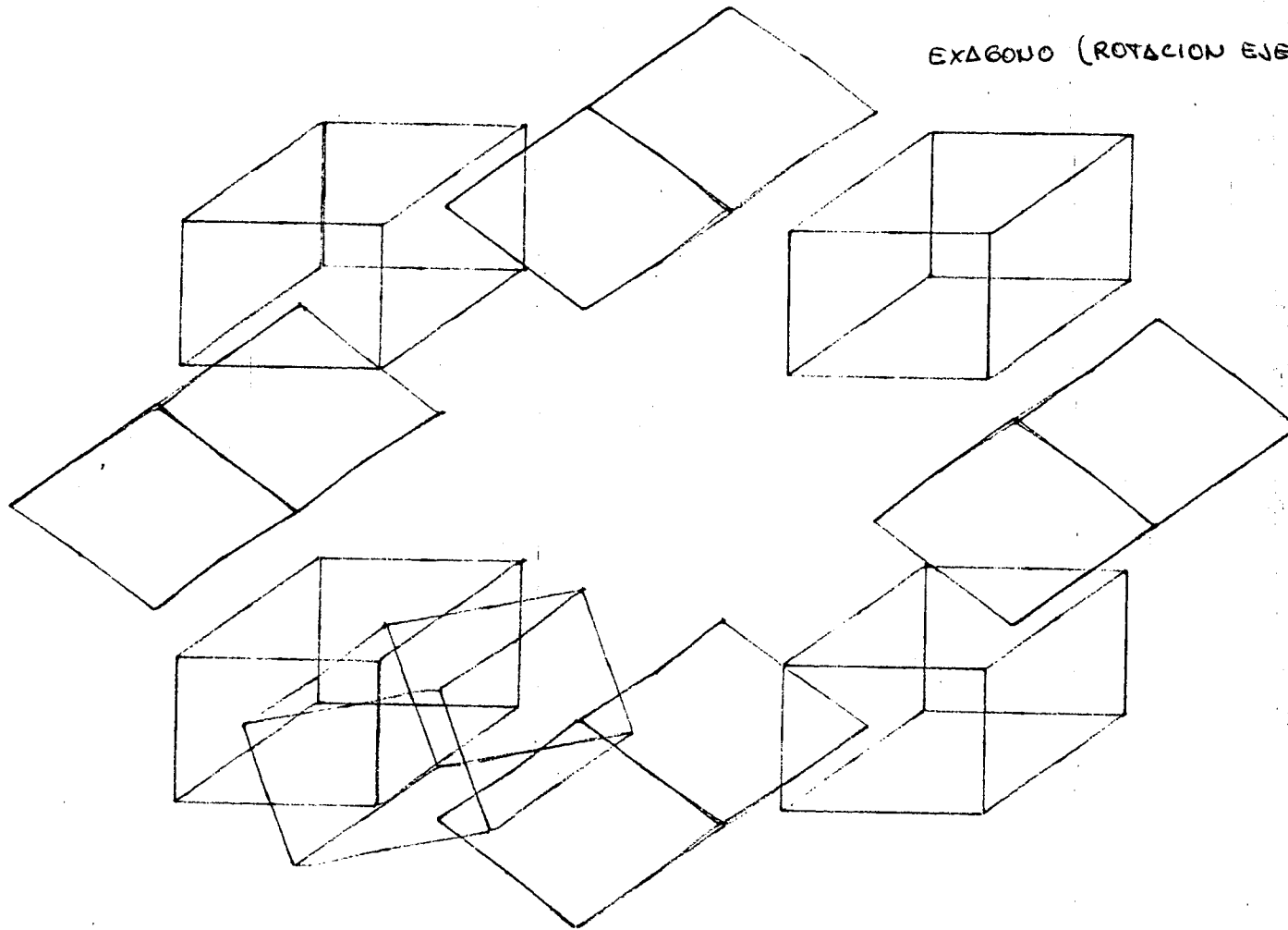
ΕΚΔΩΝΟ (ΒΡΕΣΑΝΧΑΔΗΜΑΤΟ)



x

Y

EXAGONO (ROTACION EJEX)

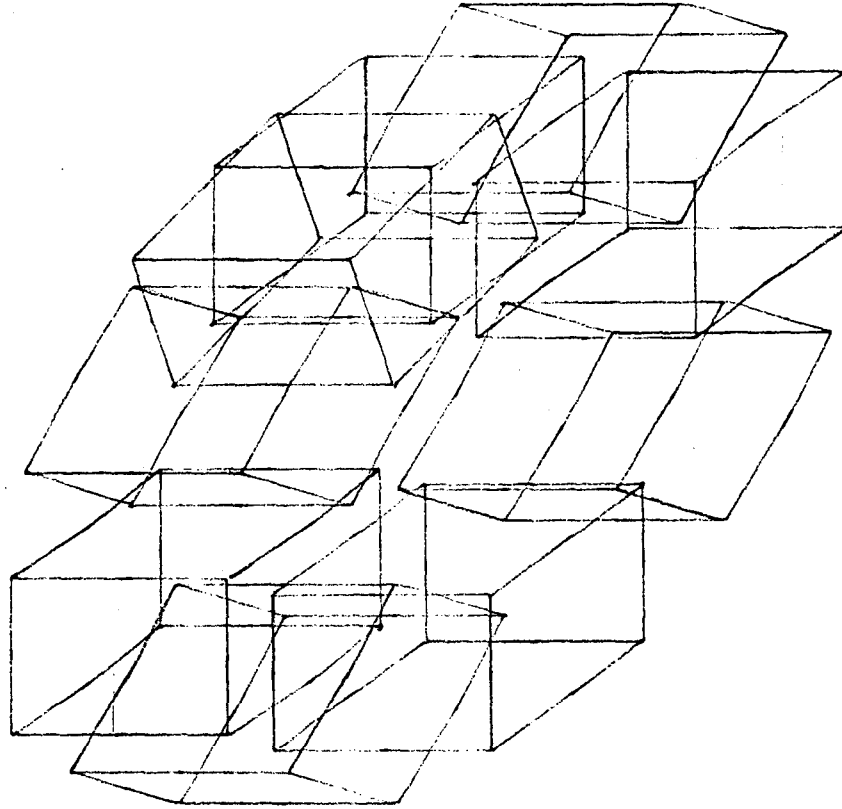


X



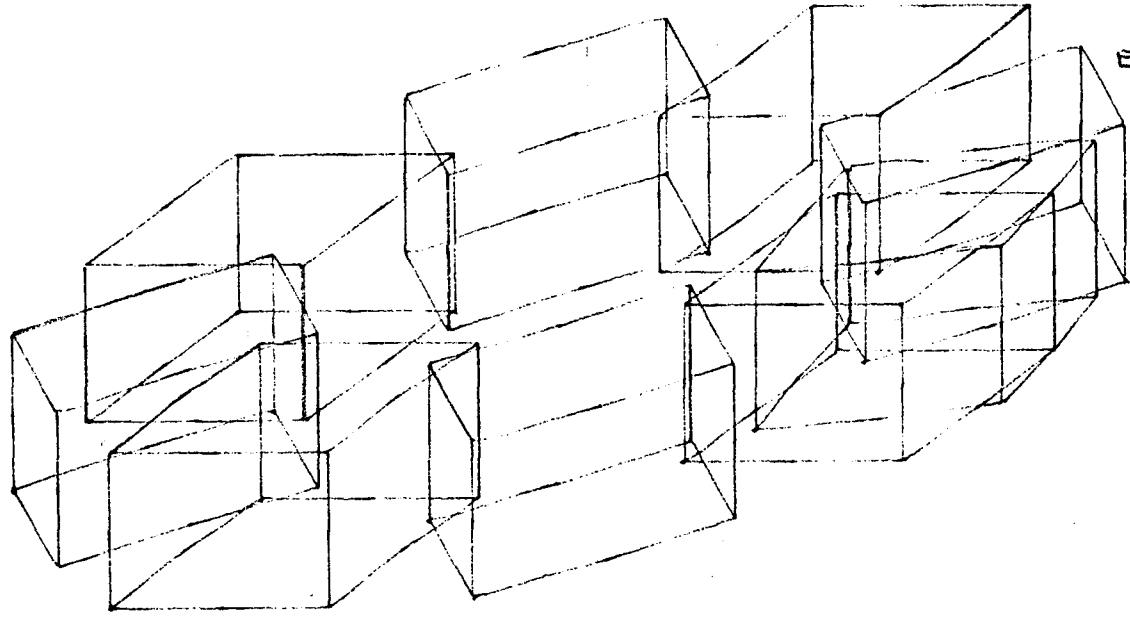
Y

EXAGONO (ROTACION EJE Y)



X

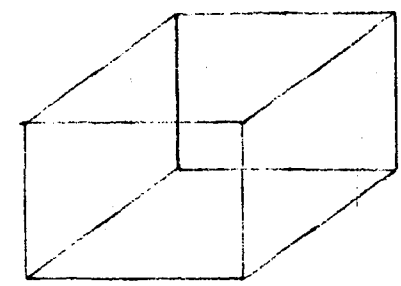
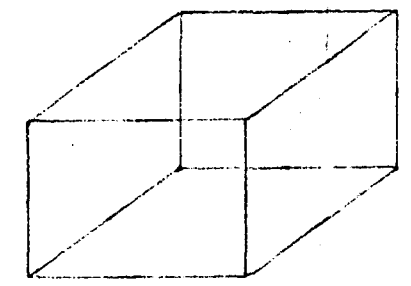
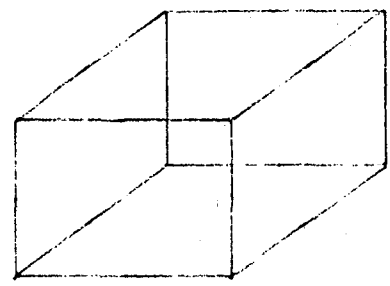
Y



EXAGONO (ROTACION EJE Z)

X

y

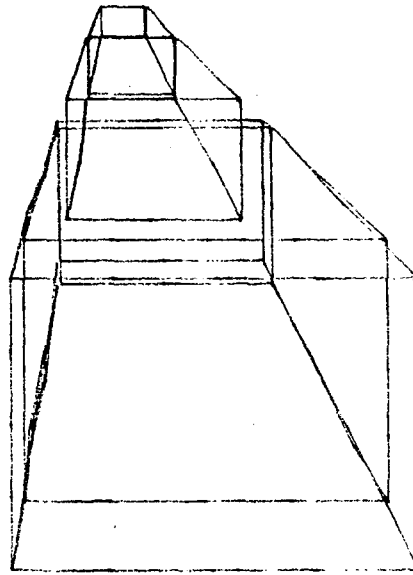


EXAGONO (REFLEXION)

x

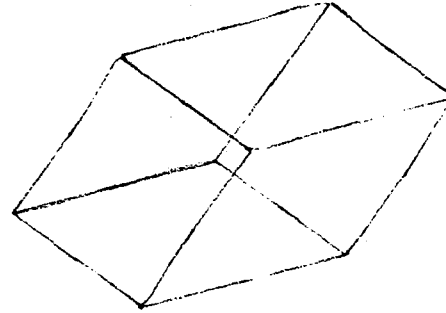
y

EXAGONO (CONICA)



x

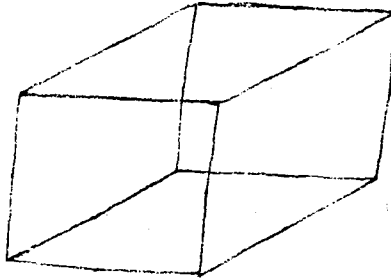
y



EXAGONO (DINETRICO)

x

y

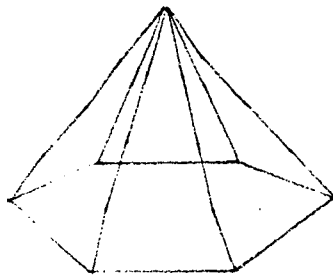


EXAGONO (ISOMETRICO)

x

Y

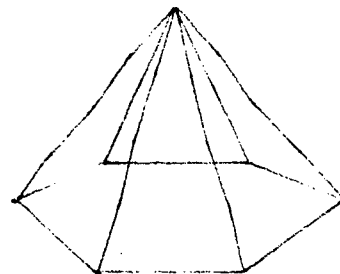
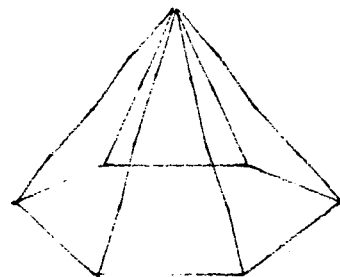
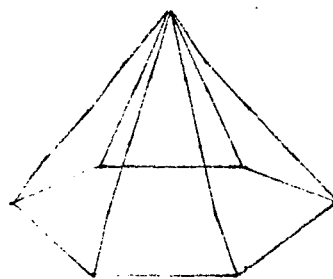
PIRAMIDE DE (FIGURA BASICA)  
BASE PENTAGONAL



X

Y

PIRAMIDE DE  
BASE PENTAGONAL (TRASLACION)

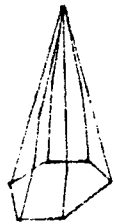


X



Y

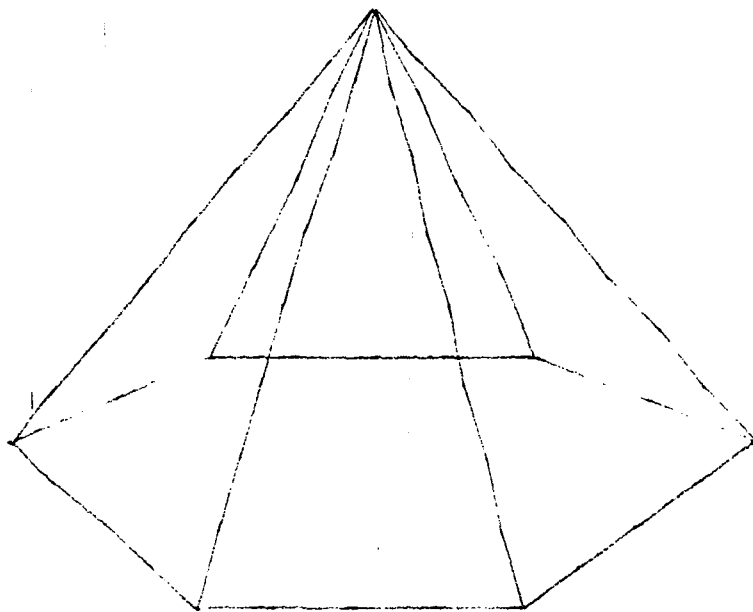
PIRAMIDE DE (ESTRECHAMIENTO)  
BASE PENTAGONAL



X

Y

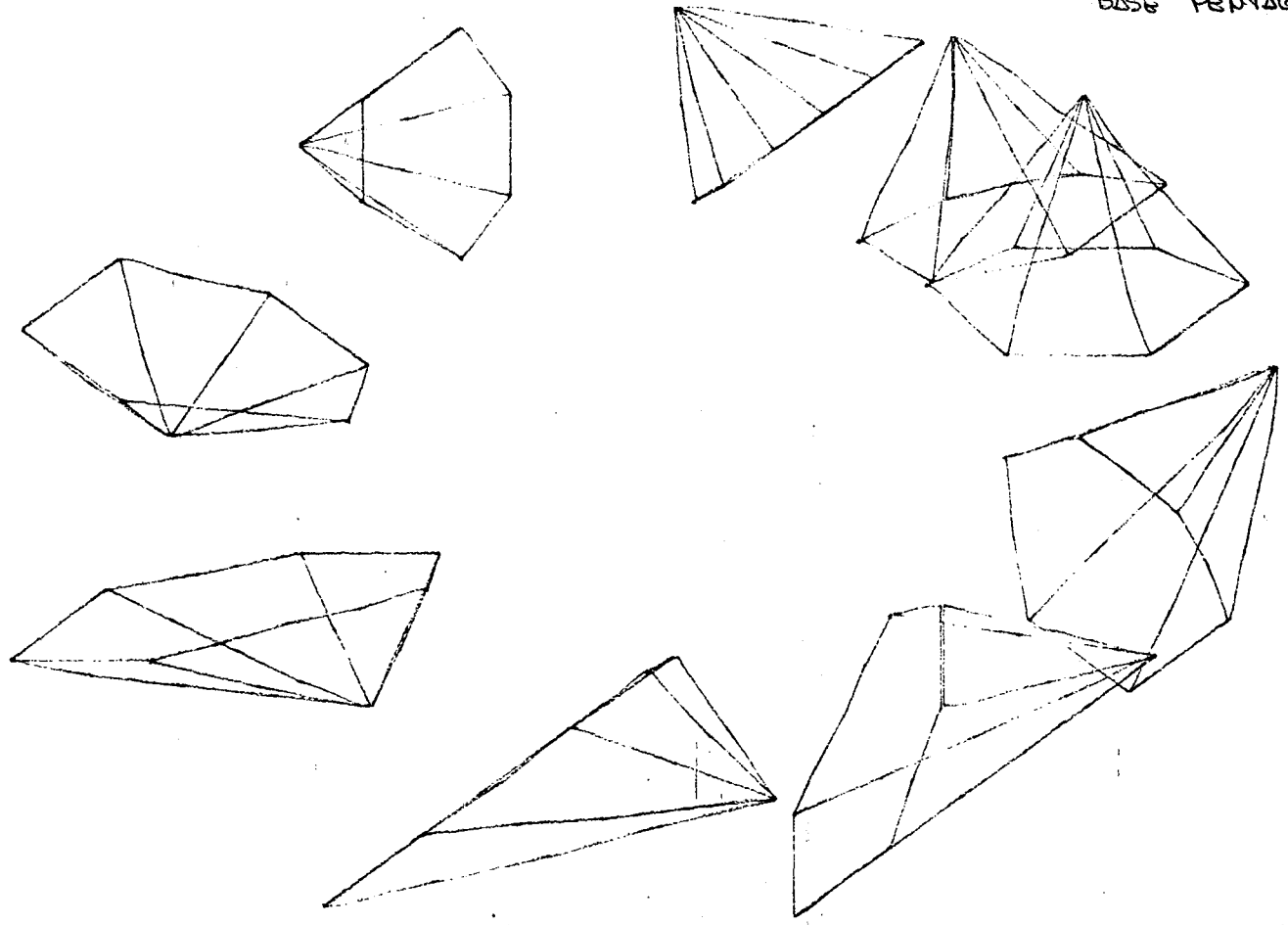
PIRAMIDE DE (ENSANCHAMIENTO)  
BASE PENTAGONAL



X

Y

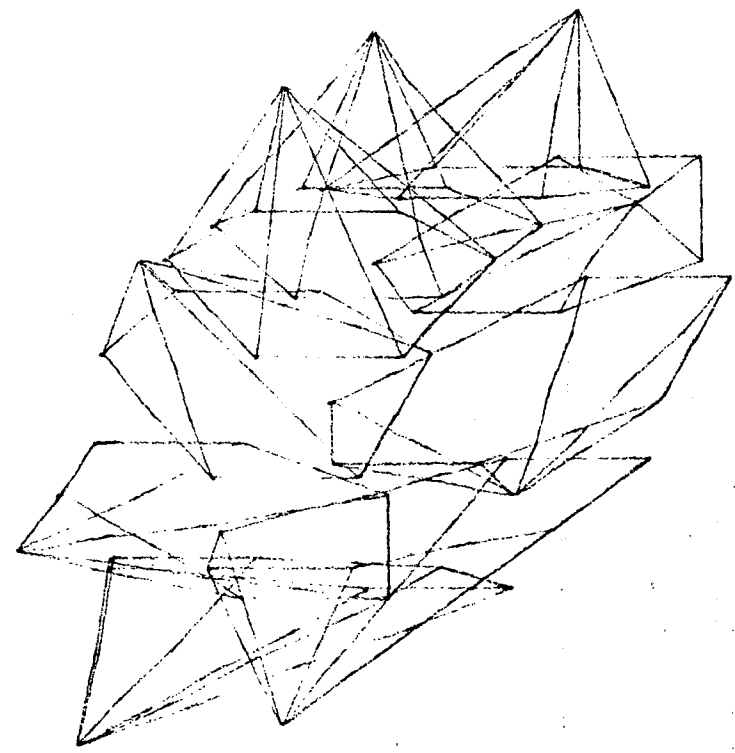
PIRAMIDE DE (ROTACION EJE X)  
BASE PENTAGONAL



X

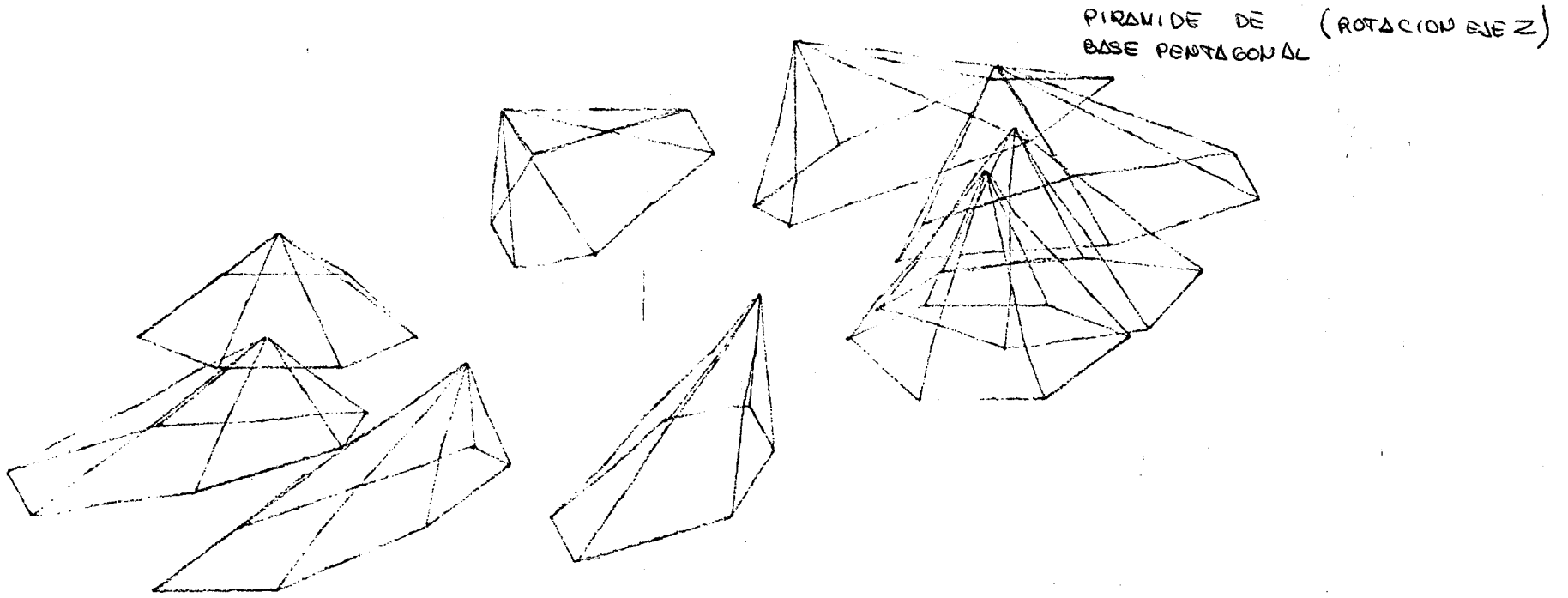
Y

PIRAMIDE DE (ROTACION EJE Y)  
BASE PENTAGONAL



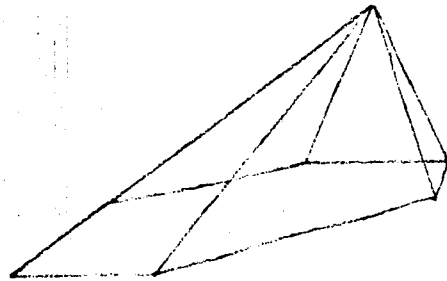
X

Y

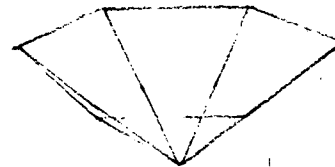


X

Y



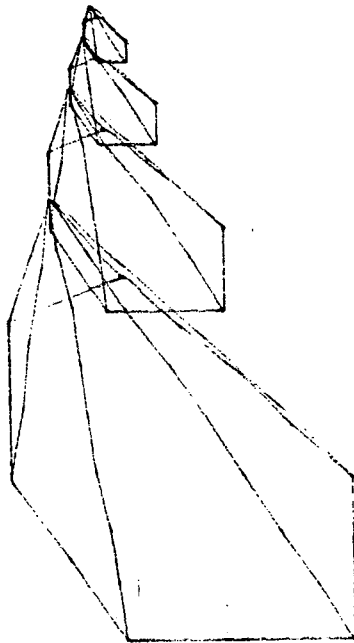
PIRAMIDE DE (REFLEXION)  
BASE PENTAGONAL



X

Y

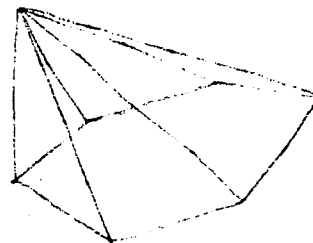
PIRAMIDE DE (CONICA)  
BASE PENTAGONAL



X

Y

PIRAMIDE DE (DIMETRICO)  
BASE PENTAGONAL



X



Y

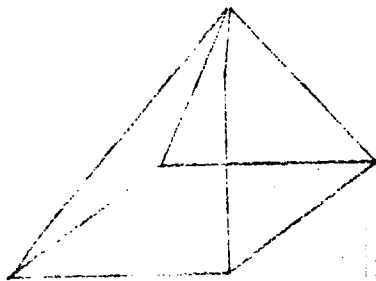
PÍRAMIDE DE (ISOMÉTRICO)  
BASE PENTAGONAL



X

Y

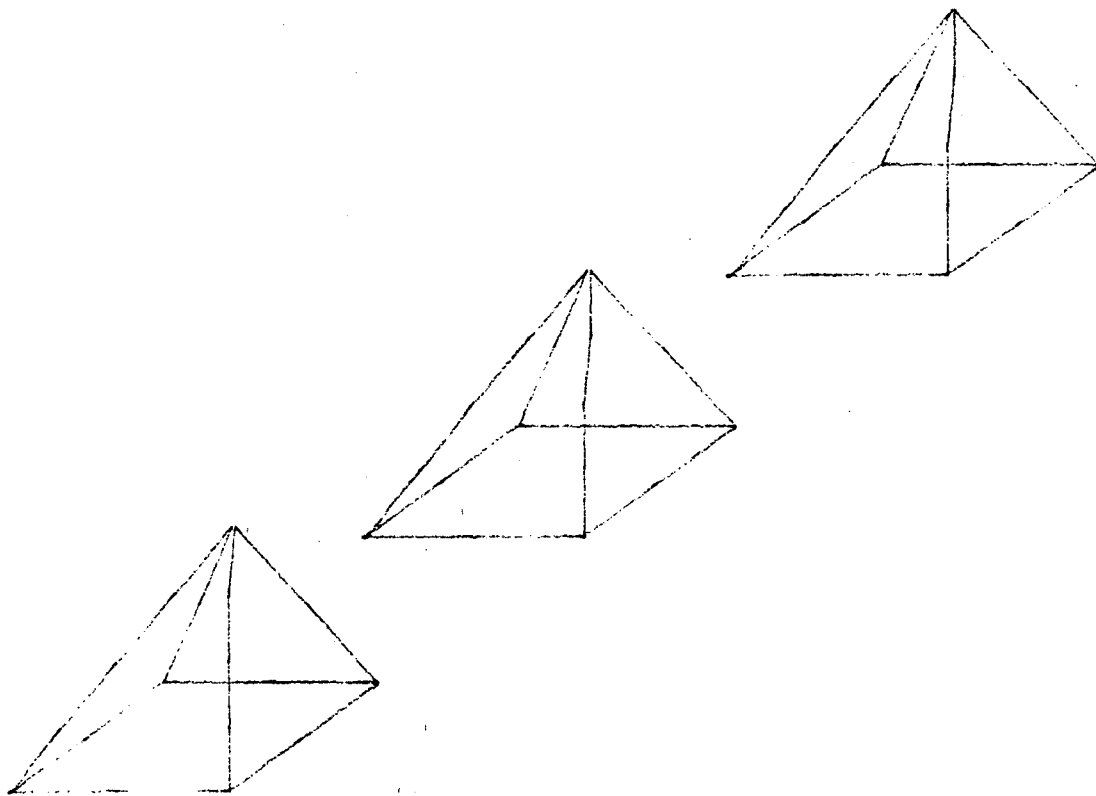
PIRAMIDE (FIGURA BASICA)



X

Y

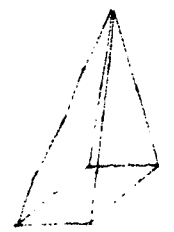
PIRAMIDE (TRASCACION)



X

y

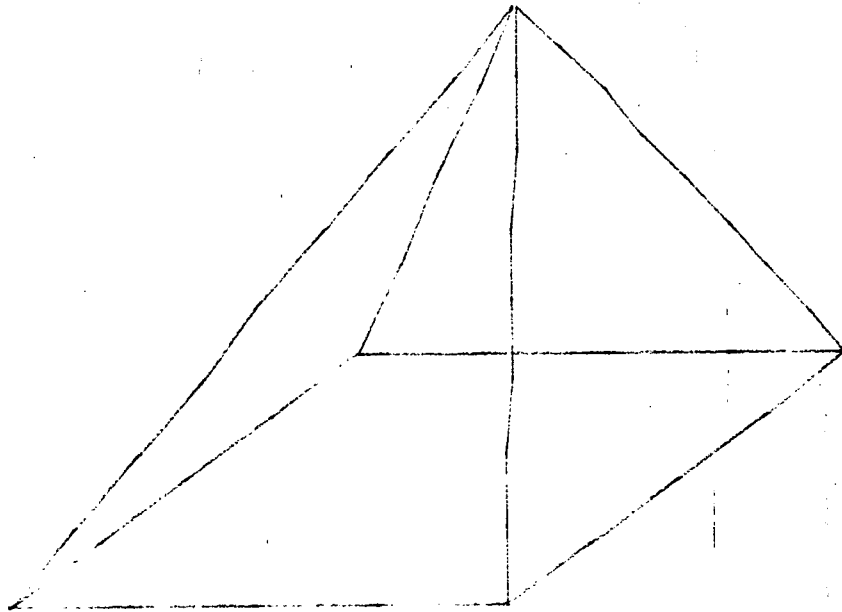
PIRAMIDE (ESTRECHAMIENTO)



x

Y

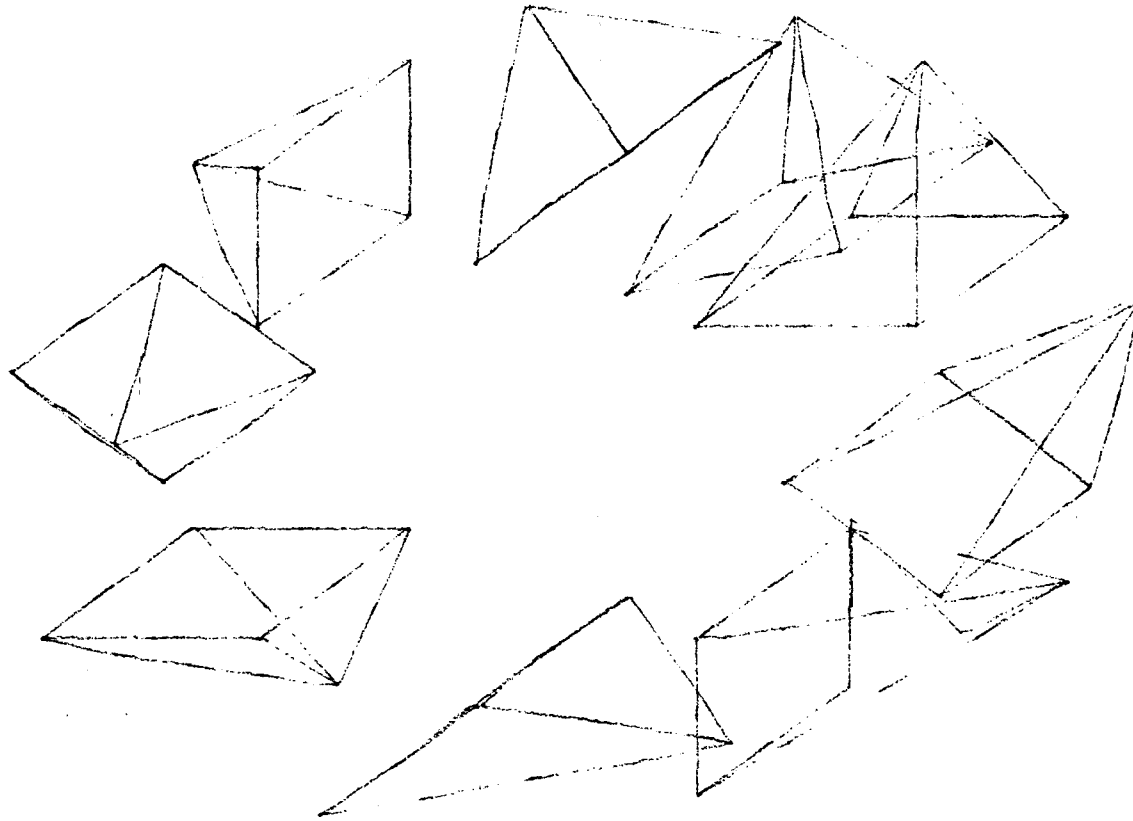
PIRAMIDE (ENSANCHAMIENTO)



X

Y

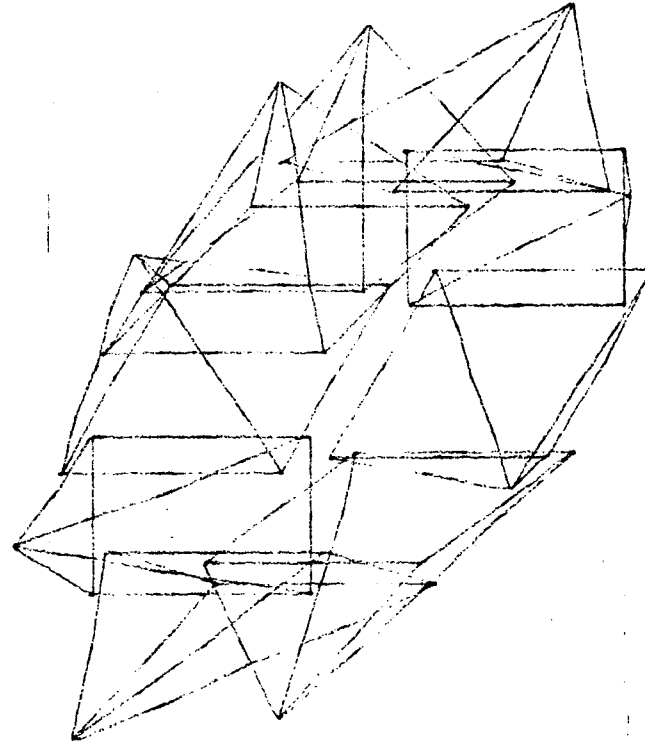
PIRAMIDE (ROTACION EJE X)



X

Y

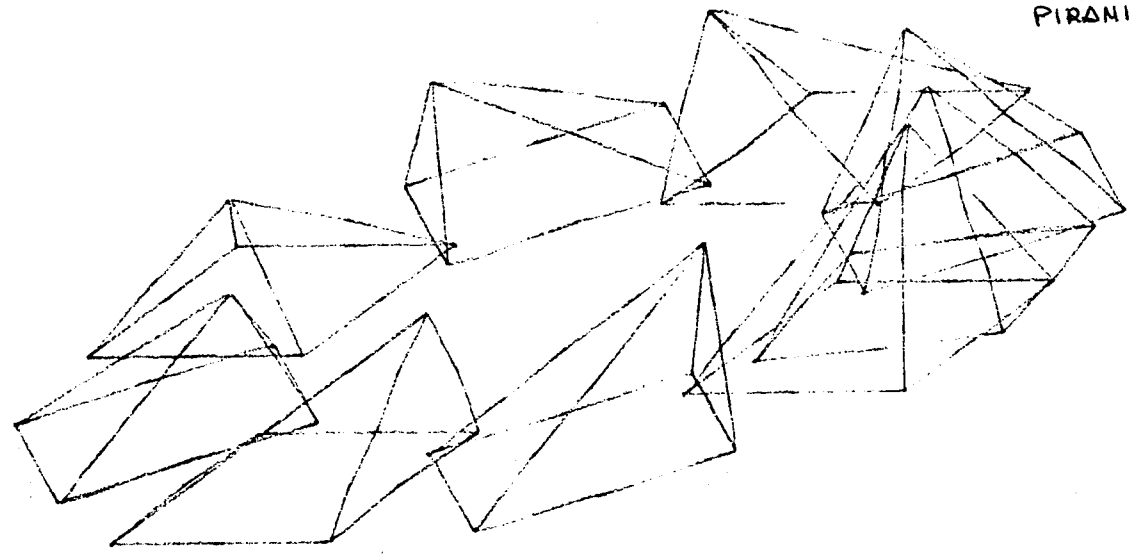
PIRAMIDE (ROTACION EJE Y)



X

y

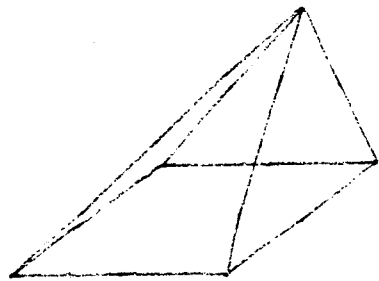
PIRÁMIDE (ROTACION EJE Z)



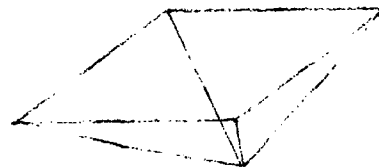
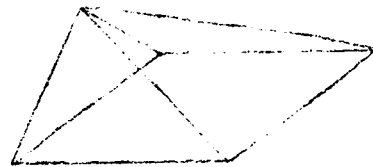
x



Y



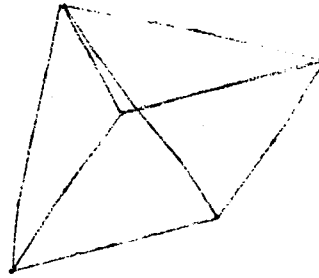
PIRAMIDE (REFLEXION)



X

Y

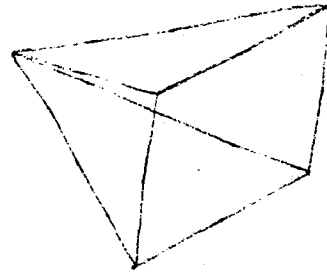
PIRAMIDE (DIMETRICO)



X

Y

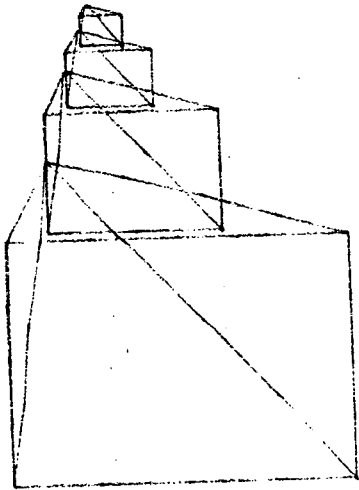
PIRAMIDE (ISOMETRICO)



X

y

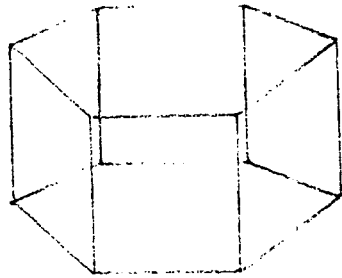
PIRAMIDE (CONICA)



x

Y

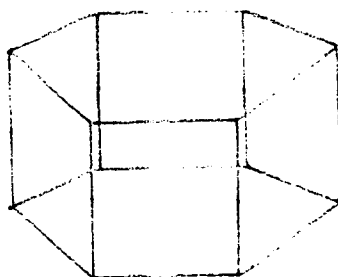
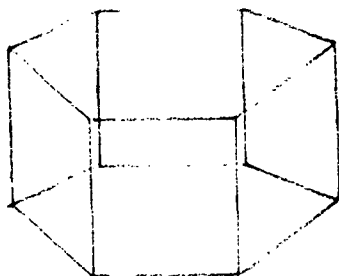
POLIGONO DE (FIGURA BASICA)  
BASE PENTAGONAL



X

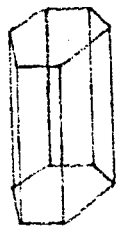
Y

POLIGONO DE  
BASE PENTAGONAL (TRASLACION)



X

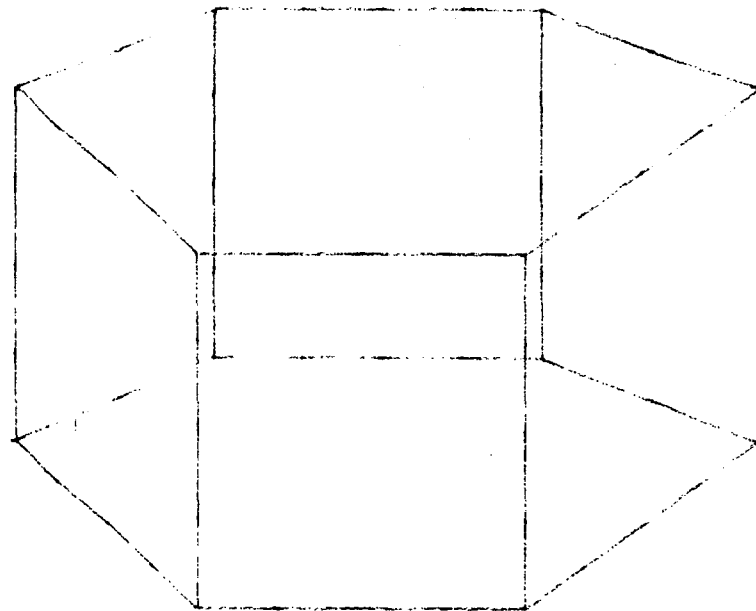
Y



POLIGONO DE (ESTRECHAMIENTO)  
BASE PENTAGONAL

X

Y



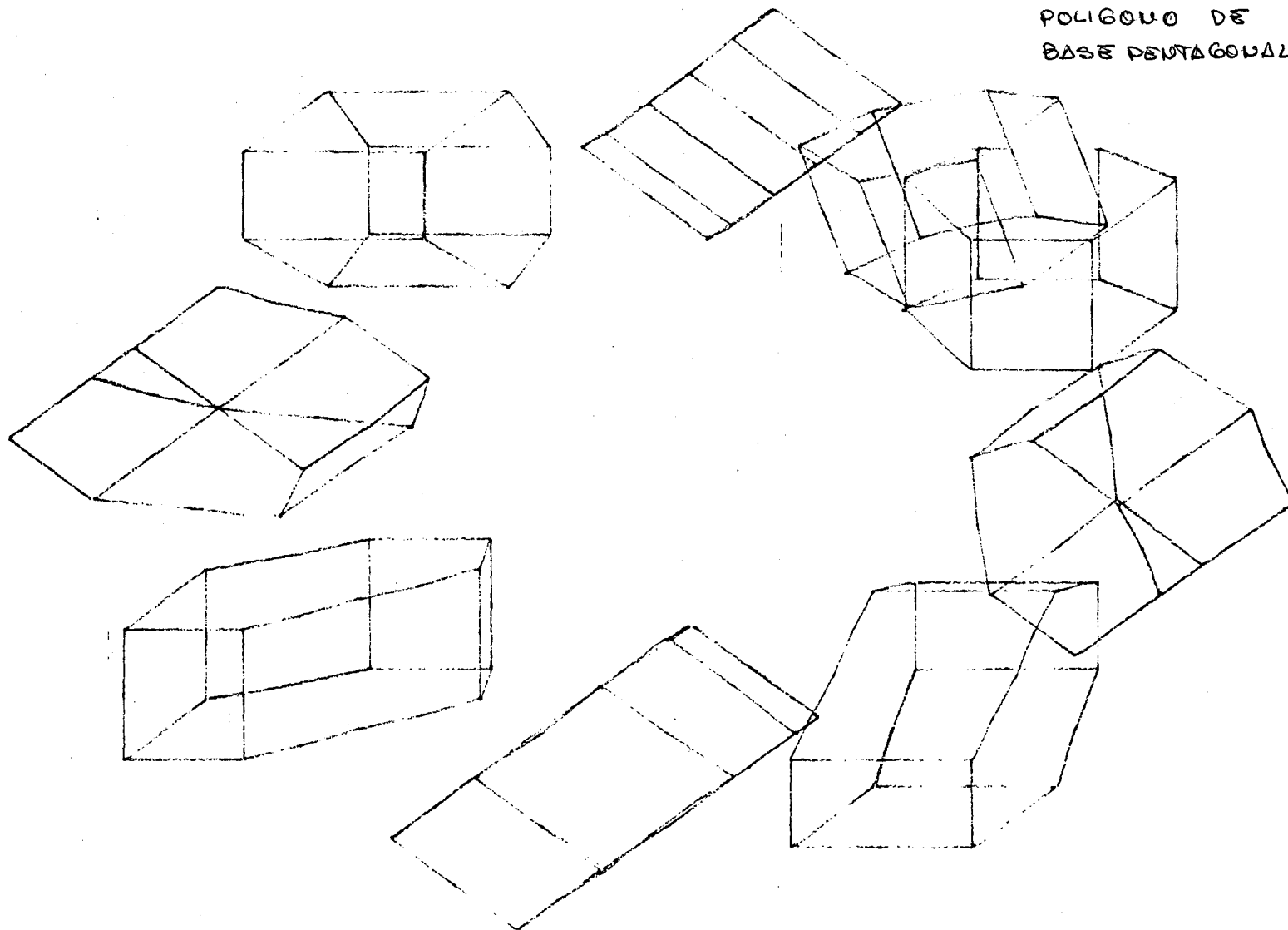
POLIGONO DE (ENSANCHAMIENTO)  
BASE PENTAGONAL

X



Y

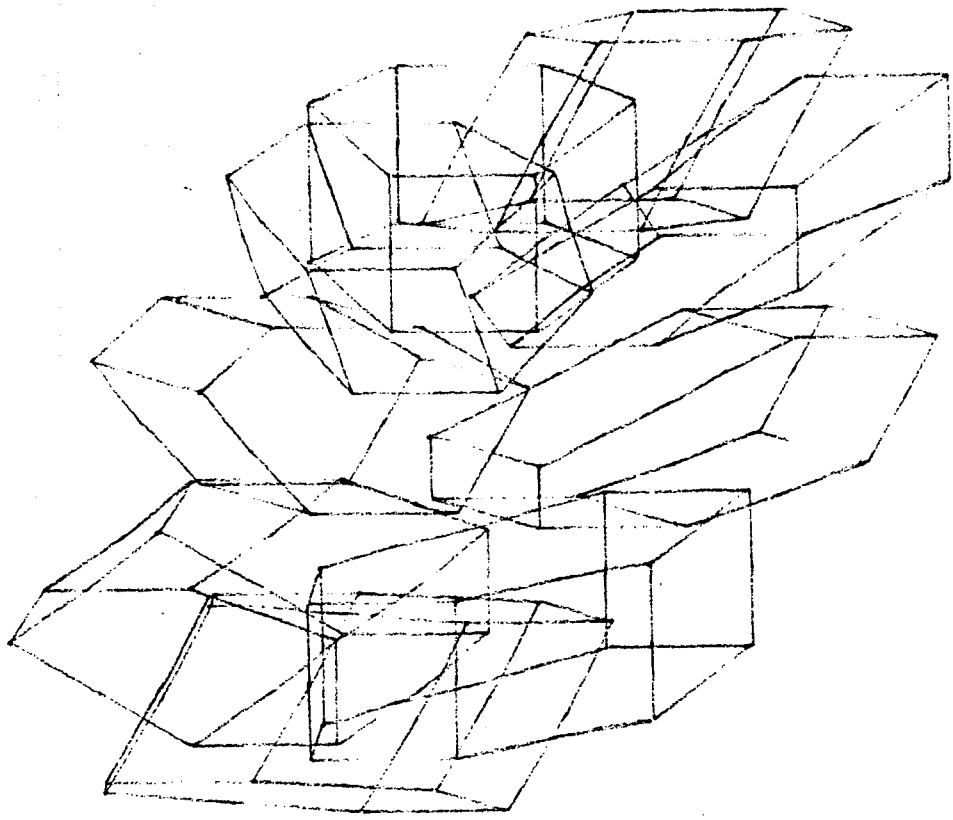
POLIGONO DE  
BASE PENTAGONAL (ROTACION EJE X)



X

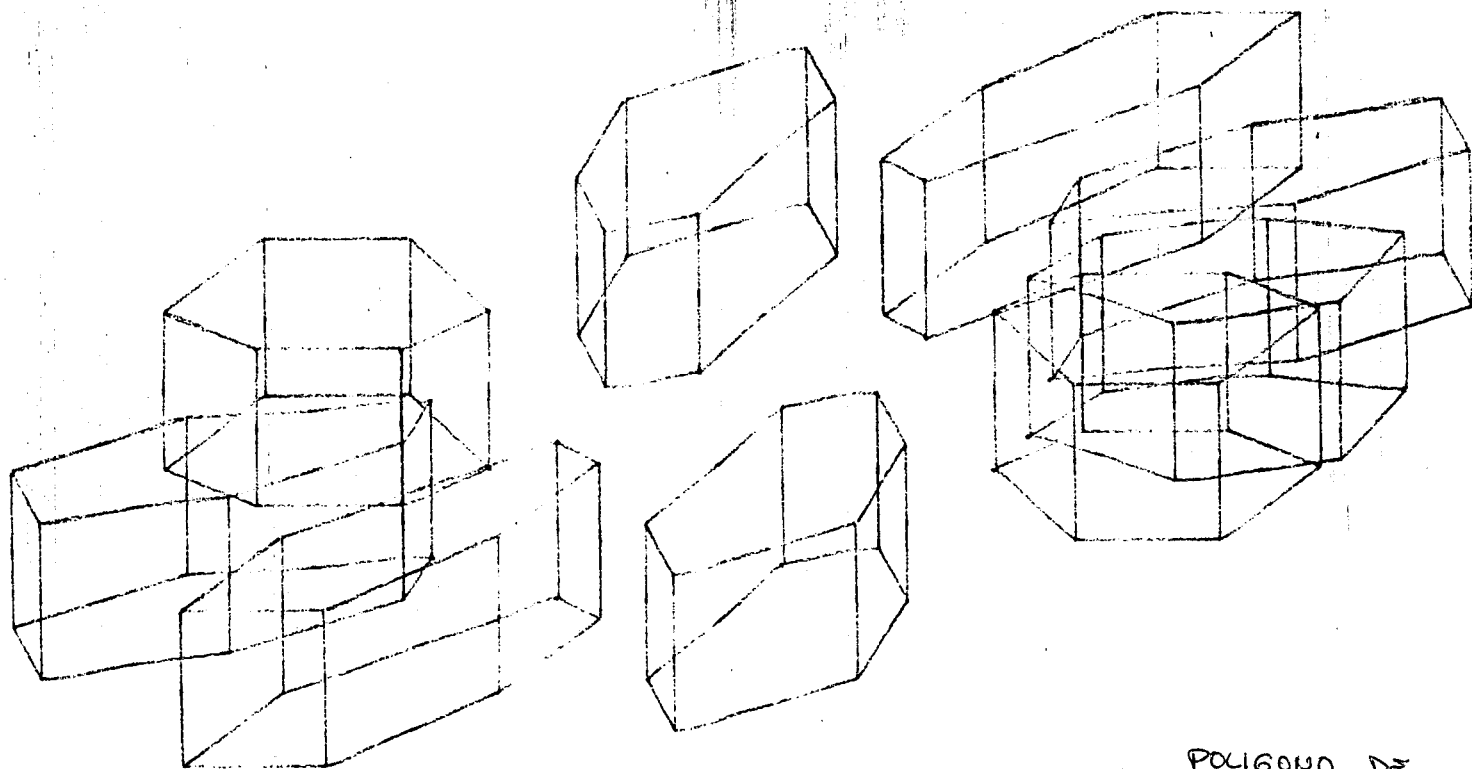
Y

POLIGONO DE (ROTACION EJE Y)  
BASE PENTAGONAL



X

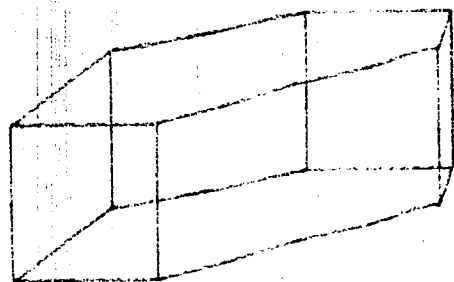
y



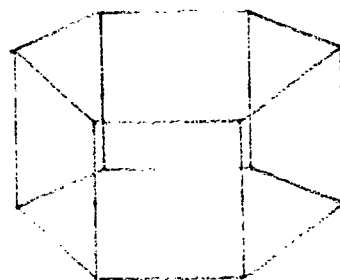
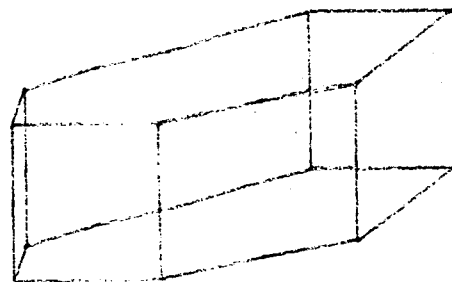
POLIGONO DE  
BASE PENTAGONAL (ROTACION BSEZ)

x

Y



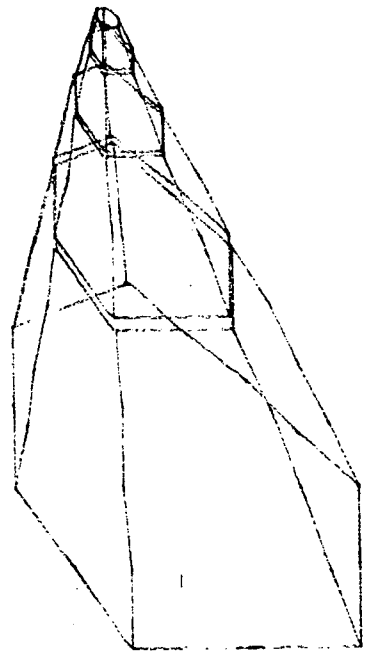
POLIGONO DE (REFLEXION)  
BASE PENTAGONAL



X

Y

POLIGONO DE (CONICA)  
BASE PENTAGONAL



X

Y

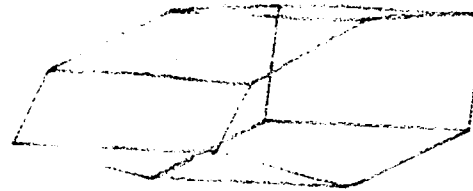
POLIGONO DE (DIESTRICO)  
BASE PENTAGONAL



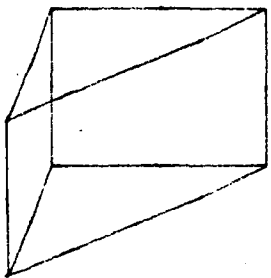
X

Y

POLIGONO DE (ISOMETRICO)  
BASE PENTAGONAL



X



POLIGONO (FIGURA BASICA)

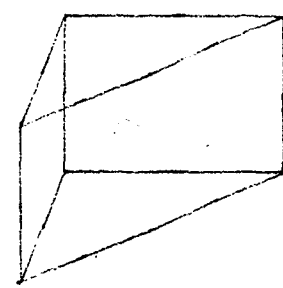
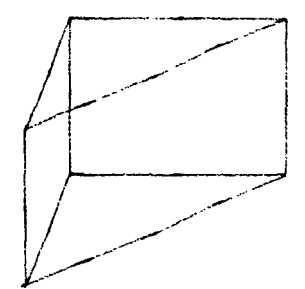
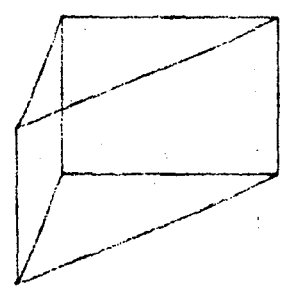
Y

X



Y

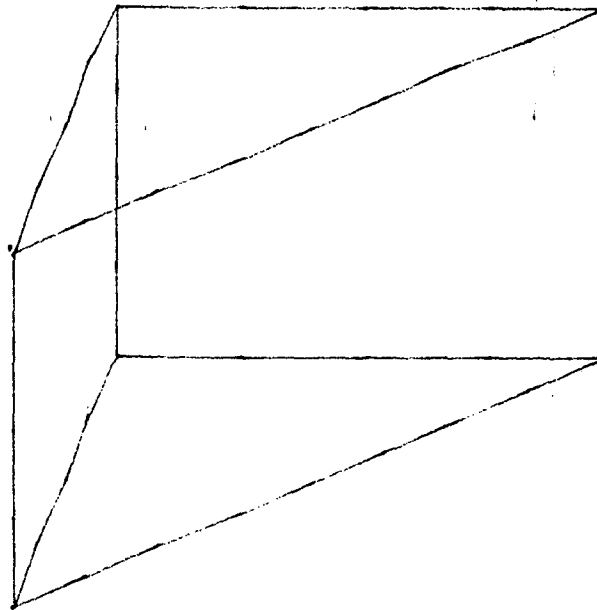
POLIGONO (TRASLACION)



X

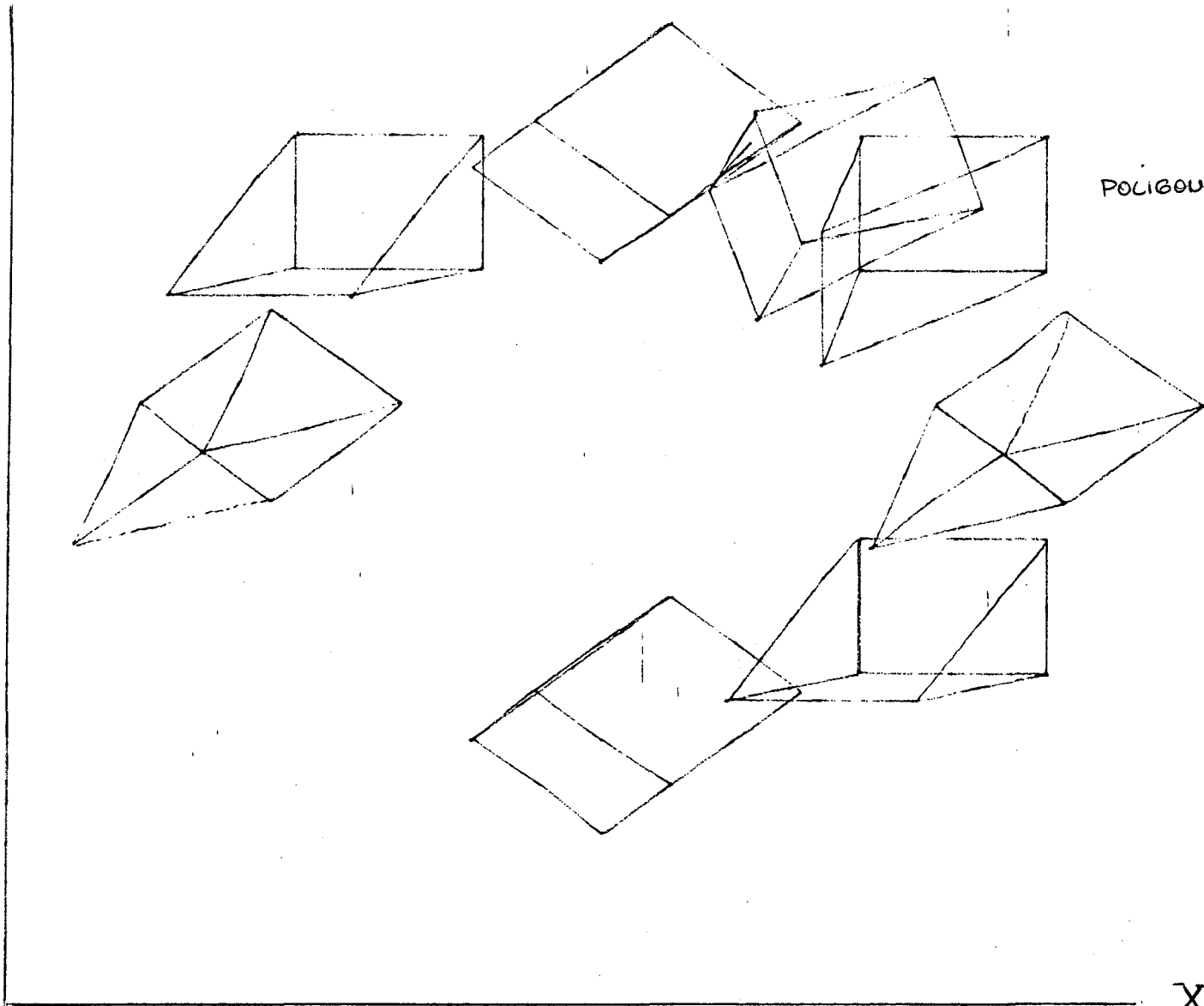
Y

POLIGONO (ENSANCHAMIENTO)



X

Y

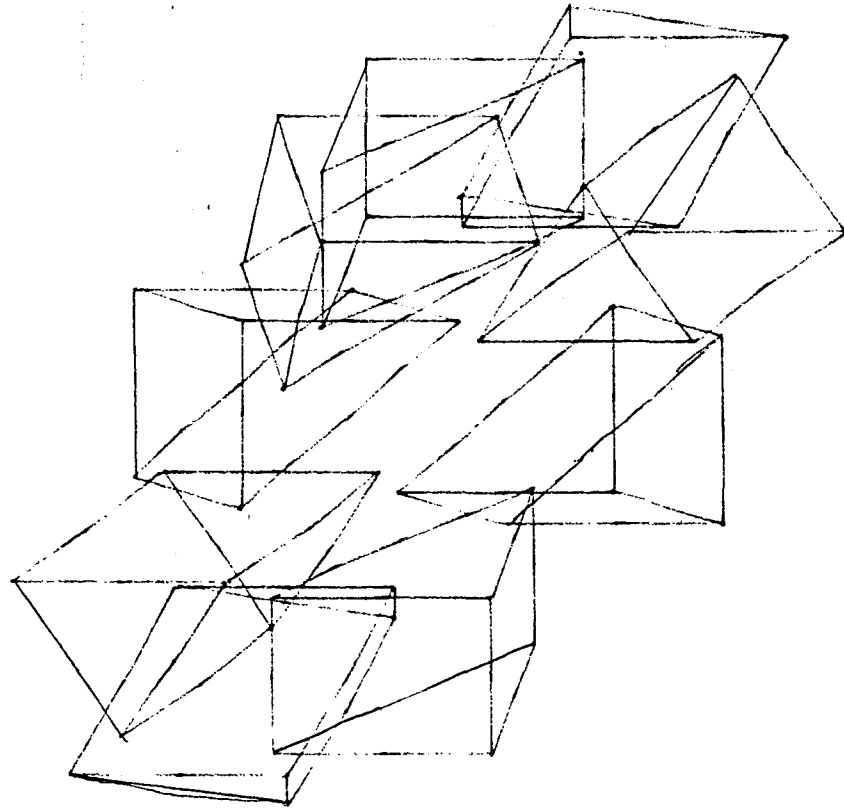


POLIGONO (ROTACION EN X)

X

Y

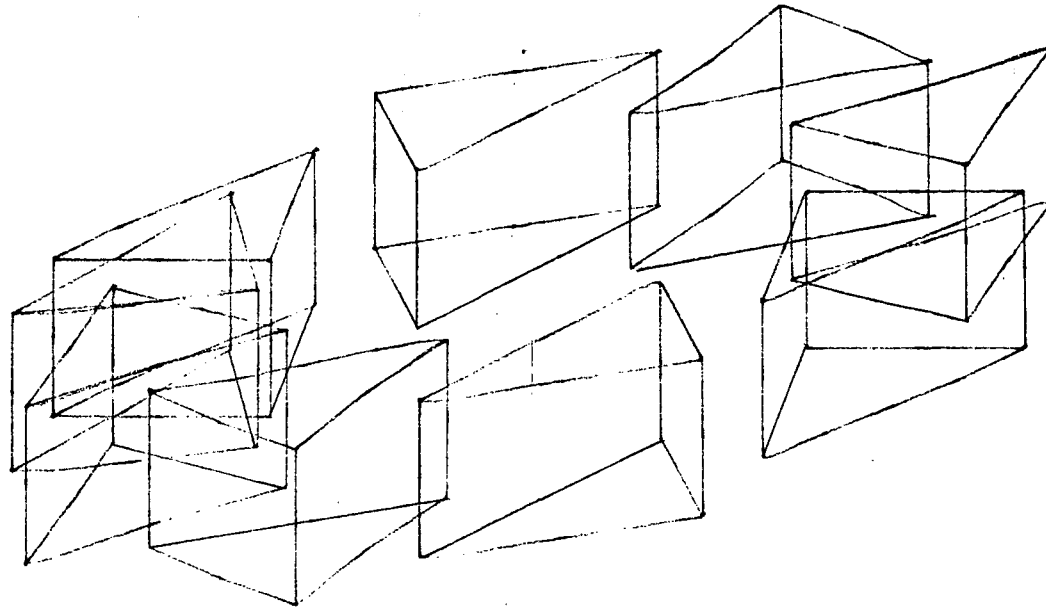
POLIGONO (ROTACION EN Y)



X

Y

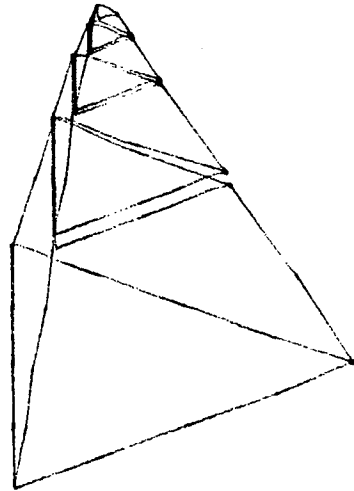
POLIGONO (ROTACION EJE Z)



X

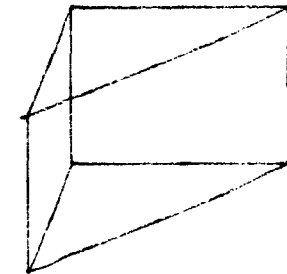
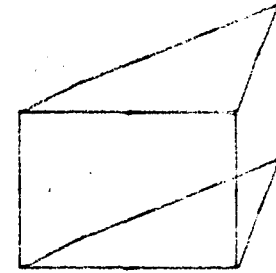
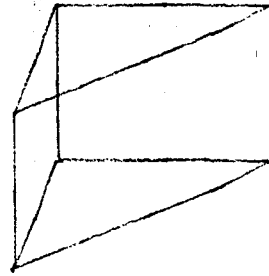
Y

POLIGONO (CONICA)



X

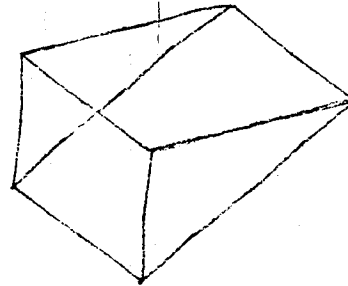
Y



POLIGONO (REFLEXION)

X

Y



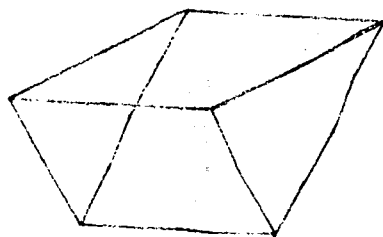
POLIGONO (DIMETRICO)

X



Y

POLIGONO (ISOMETRICO)



X

## 2. PROGRAMA EXA

### 2.1 ANALISIS DEL PROGRAMA EXA

En esta sección se estudia el funcionamiento del programa EXA, cuya estructuración en módulos independientes permite un cómodo análisis de cada una de sus funciones.

### 2.2. CARACTERISTICAS GENERALES

El programa EXA está concebido para dibujar cualquier función en coordenadas cartesianas en el PLOTTER. Se compone de un programa de control y siete módulos independientes entre sí que realizan las funciones de traslación, reducción, rotación, reflexión, isométrico, dimétrico, cónica. Tanto el programa de control como los distintos módulos están codificados en lenguaje BASIC / 3000.

Una de las peculiaridades del programa es la utilización de una función recursiva.

Una función recursiva es una función que se llama así misma; por ejemplo el factorial de un número se define como dicho número por el factorial del número anterior.

$$5! = 5 * 4! \\ = 5 * 4 * 3!$$

$$4! = 4 * 3!$$

Su propia definición se incluye así misma.

Hay que tener en cuenta que el PLOTTER se considera a la hora de dibujar como un cuadrado de  $256 * 256$  puntos, y que cuando la recta es mayor que 20 pierde linealidad. Para resolverlo es por lo que se aplica la función recursiva.

Con la función recursiva se divide la recta por la mitad y se vuelve a comparar, y así sucesivamente hasta conseguir una recta de distancia adecuada. Trazándose posteriormente la recta de división en división hasta llegar a la recta final deseada.

## 2.3 MODULO 1

NOMBRE DEL PROGRAMA: EXA

EXPLICACION DEL ORGANIGRAMA

### BLOQUE 1

Definición de la función recursiva como FNT (L9, X8, Y8, X9, Y9). Variables :

X8 , Y8 : Punto inicial

X9 , Y9 : Punto final

L9 : corresponde a un comando, y nos selecciona las distintas posibilidades que se pueden plantear.

L9 = 0 Traza una recta desde el punto inicial al punto final con el lápiz abajo.

L9 = 1 Se desplaza desde el punto inicial al punto final con el lápiz arriba.

L9 = 2 Inicialización de las variables.

L9 = 3 Vuelca las variables en un buffer hasta un final de fichero.

Para el caso L9 = 0 es donde se puede observar el uso de la función recursiva.

#### BLOQUE 2

Llamada a la subrutina recursiva.

#### BLOQUE 3

Explora la posibilidad de volcar las variables en un buffer hasta un final de fichero . En caso afirmativo bloque 20 donde finaliza la función . En caso negativo al bloque 4.

#### BLOQUE 4

Explora la posibilidad de inicialización de las variables . En caso afirmativo bloque 20 donde finaliza la función . En caso negativo al bloque 5.

#### BLOQUE 5

Explora la optimización del programa mediante el ahorro de posiciones en la base de datos para poner más caracteres . En caso afirmativo al bloque 2 . En caso

negativo al bloque 6.

#### BLOQUE 6

Explora la posibilidad de desplazarse desde el punto de inicial con el lápiz arriba. En caso afirmativo coloca el lápiz arriba y va al bloque 7. En caso negativo al bloque 9.

#### BLOQUE 7

Es una subrutina de escritura en fichero. En ella explora la posibilidad de que las variables almacenadas son mayor o igual que 80 caracteres. En caso afirmativo imprime en el fichero los 80 caracteres y vuelve al bloque 3. En caso negativo retorna donde se hizo la llamada .

#### BLOQUE 8

Analiza las variables X8 , Y8 .

#### BLOQUE 9

Es una llamada a la subrutina de formateo de variable . Las variables X8 , Y8 que esta en decimal se convierten en hexadecimal mediante la subrutina de conversión decimal- hexadecimal.

#### BLOQUE 10

Explora la posibilidad de trazar un recta desde el punto inicial con el lápiz abajo. En caso afirmativo coloca el lápiz abajo va al bloque 7. En caso negativo al bloque 11.

#### BLOQUE 11

Explora la posibilidad de desplazarse desde el punto final con el lápiz abajo. En caso afirmativo al bloque 12. En caso negativo al bloque 14.

#### BLOQUE 12

Se explora la posibilidad de que la ecuación de la recta sea mayor que 20. En caso afirmativo se hace una optimización de la distancia mediante el siguiente cambio.

$$( X8 , Y8, (X9-X8)/(2+X8) , (Y9-Y8)/(2+X9) )$$

Para el caso que  $X9, Y9$  sean mayores que  $X8, Y8$ .

$$( (X9-X8)/(2+X8), (Y9-Y8)/(2+X9) , X9 , Y9 )$$

Para el caso que  $X8, Y8$  sean mayores que  $X9, Y9$ .

Y retorna al bloque 3. En caso negativo lápiz abajo y va al bloque 7.

#### BLOQUE 13

Explora la posibilidad de trazar desde el punto de final con el lápiz arriba. En caso afirmativo bloque 7. En caso negativo bloque 21 donde finaliza la función.

#### BLOQUE 14

La variable X8 que esta en decimal se convierte en hexadecimal mediante la subrutina de conversión decimal hexadecimal.

#### BLOQUE 15

Es una subrutina de conversión decimal hexadecimal . Esta conversión se realiza mediante la división del número entero que corresponde a la variable X8 , Y8 , X9 , Y9 por 16 . Se explora la posibilidad de que el resto de la división sea mayor que 9 . En caso negativo se ha realizado el proceso de conversión y retorna donde se hizo la llamada . En caso negativo se explora la posibilidad de que el cociente es mayor que 16 . En caso positivo vuelve a realizarse la división del número por 16 . En caso negativo el valor del resto es igual al valor del cociente , vuelve a explorar la posibilidad de que el resto de la división es mayor que 9.

#### BLOQUE 16

La variable Y8 que esta en decimal se convierte en hexadecimal mediante la subrutina de conversión decimal hexadecimal.

#### BLOQUE 17

Analiza las variables X9 . Y9.

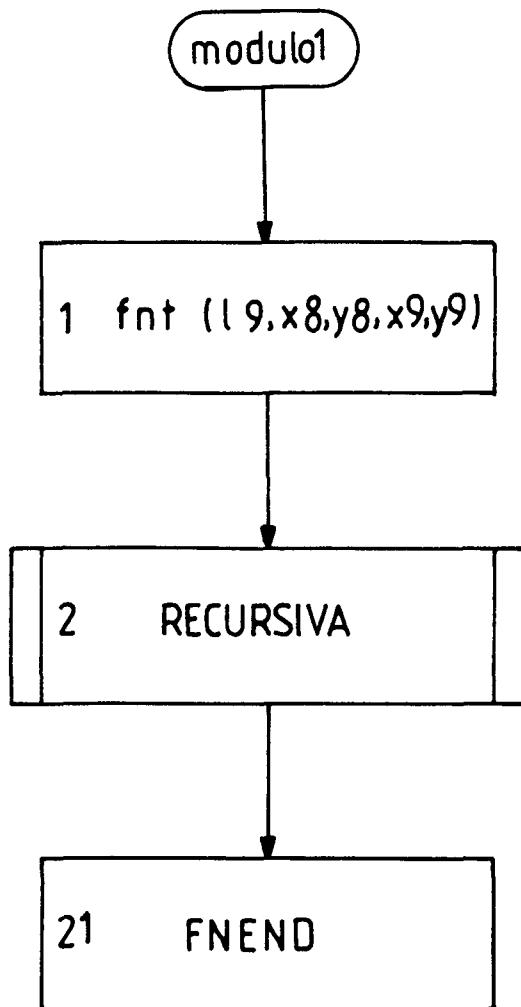
#### BLOQUE 18

La variable X9 que esta en decimal se convierte en hexadecimal mediante la subrutina de conversión decimal hexadecimal.

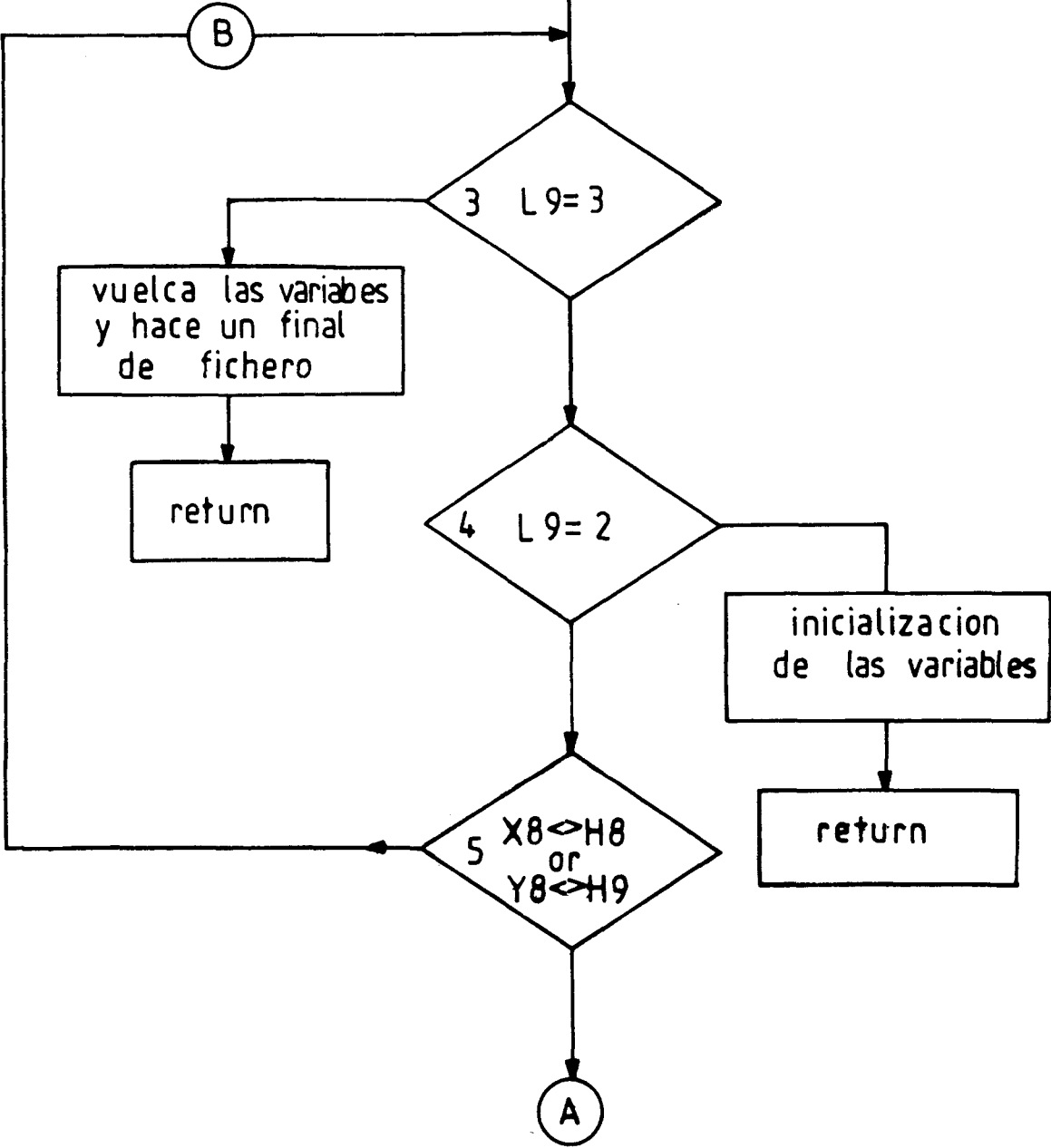
#### BLOQUE 19

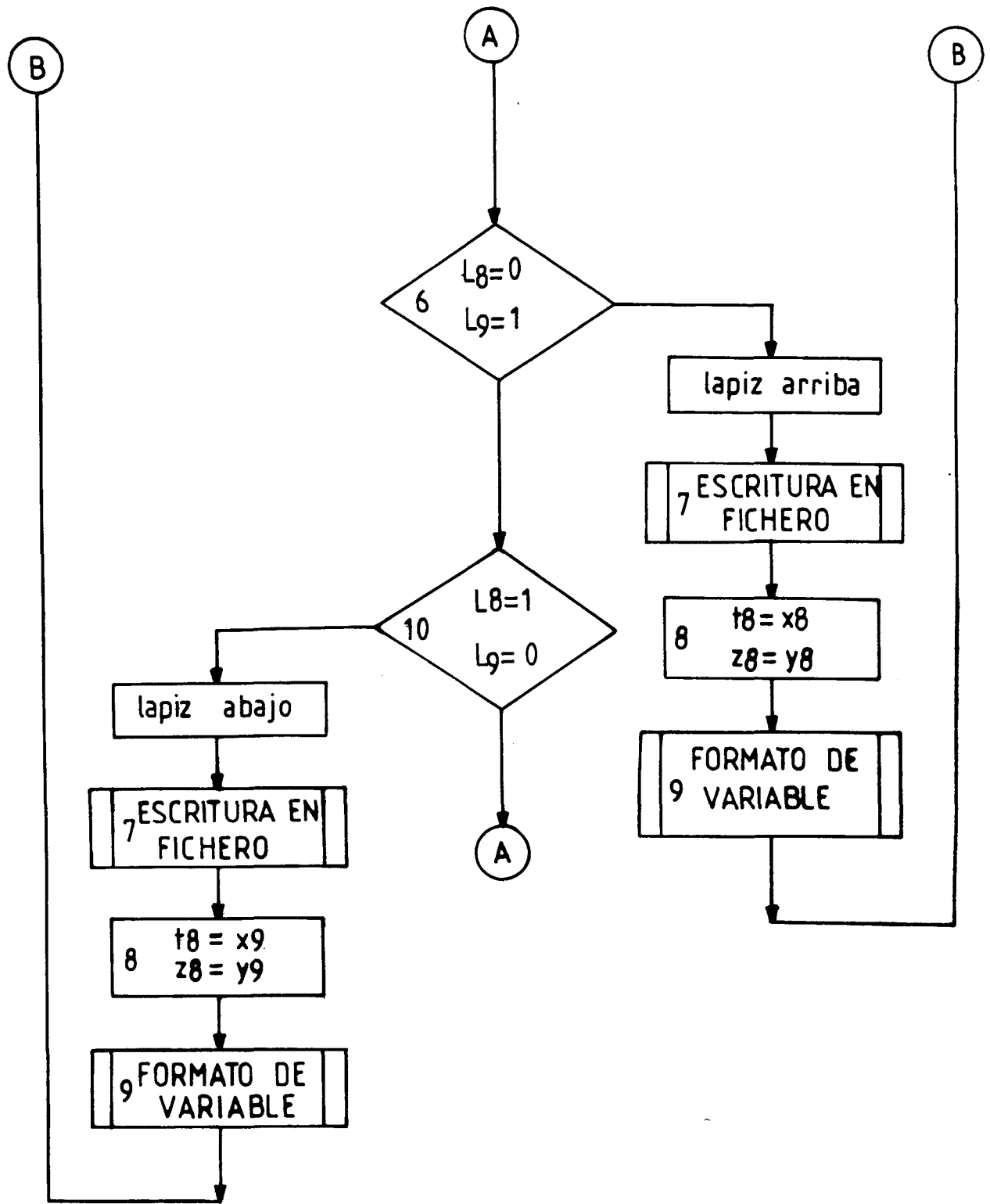
La variable Y9 que esta en decimal se convierte en hexadecimal mediante la subrutina de conversión decimal hexadecimal.

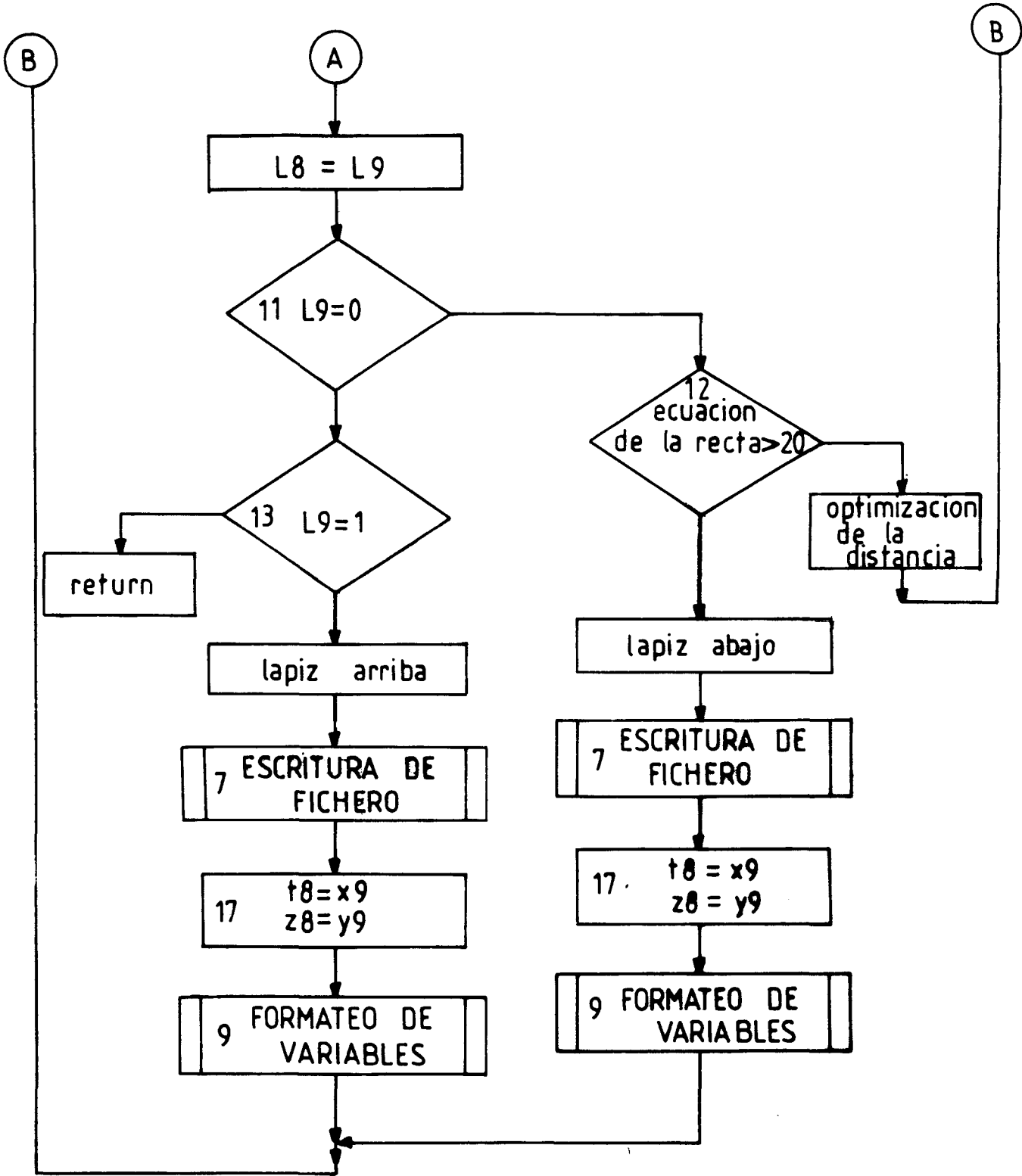


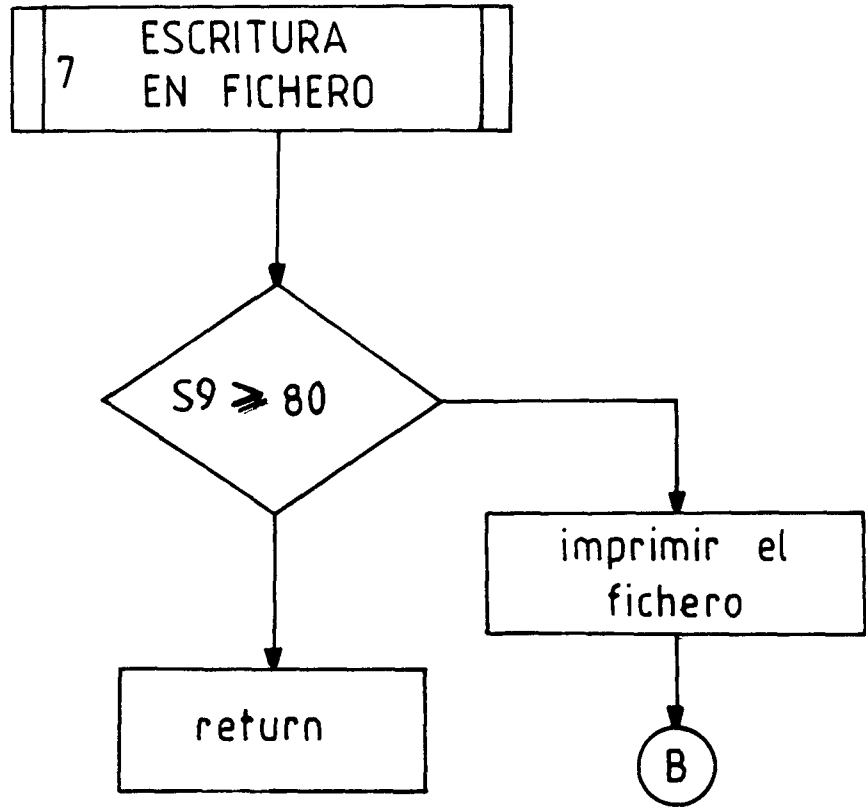


2 RECURSIVA









9 FORMATEO DE VARIABLES

14  $v8 = x8, x9$  18

15 CONVERSION DECIMAL HEXADECIMAL

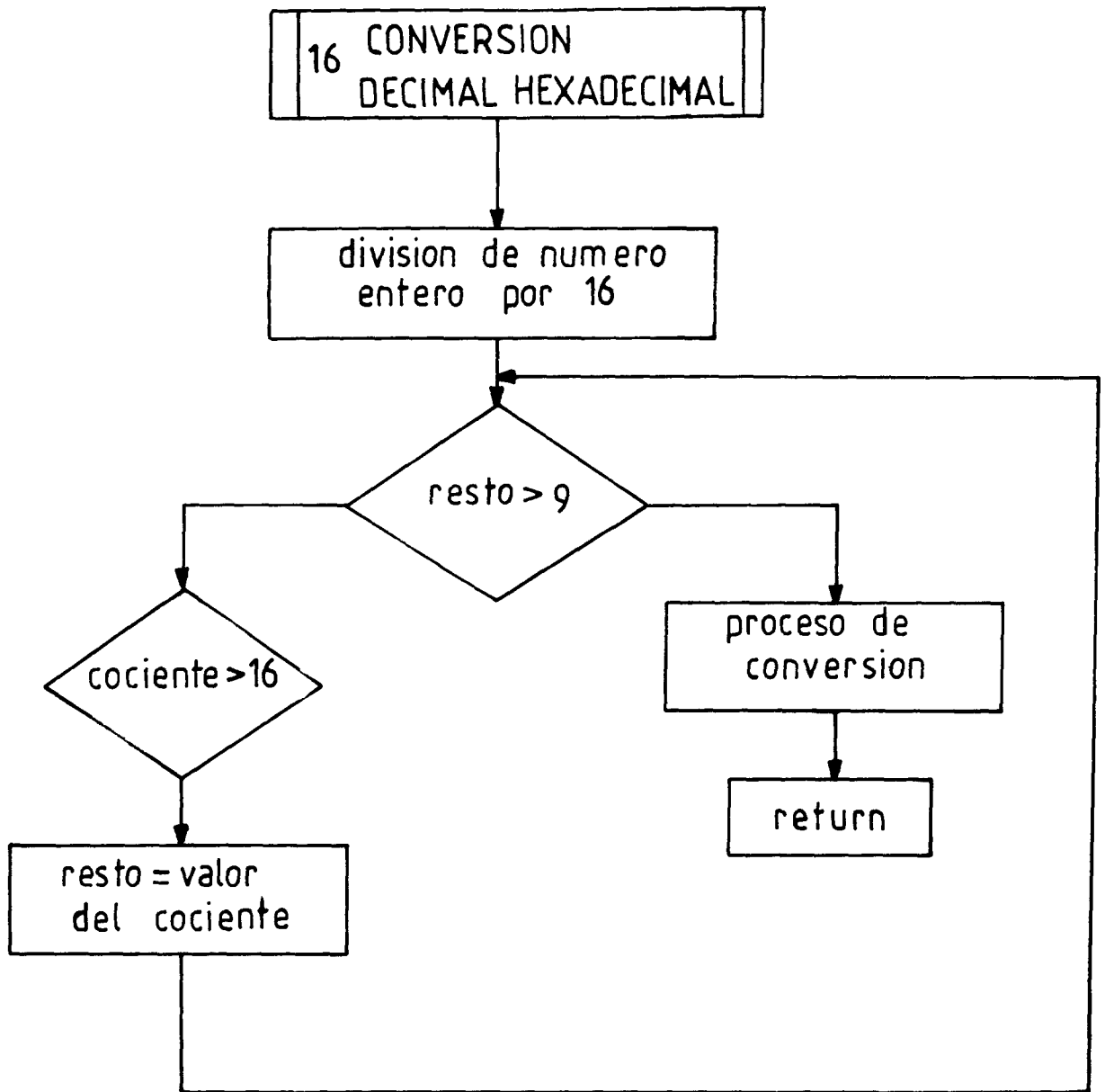
7 ESCRITURA EN FICHERO

16  $v8 = y8, y9$  19

15 CONVERSION DECIMAL HEXADECIMAL

7 ESCRITURA EN FICHERO

B



## 2.4 MODULO 2. CONFIGURACION DEL DIBUJO

Este módulo permite la configuración del dibujo cualquiera que sea la forma.

### EXPLICACION DEL ORGANIGRAMA

#### BLOQUE 1.

FICHERO 2: OUTPUT1 Localiza la posición de grabación correspondiente a la introducción de datos en tres dimensiones de la figura .

FICHERO 3: INPUT1 localiza la posición de grabación correspondiente a la transformación del fichero OUTPUT1 en dos dimensiones.

FICHERO 4: GRAPH13 Localiza la posición de grabación correspondiente a la transformación del fichero INPUT1 mediante la función  $K = \text{FNT} ( L, X3, X4, X1, X2 )$  .

Inicialización de las variables X1,X2,X3,X4.

VARIABLES:

L: Indica la posición de el lápiz del plotter.

L=0 Lápiz abajo

L=1 Lápiz arriba

X1,X2: Corresponde al valor inicial de X,Y.

X3,X4: Corresponde al valor final de X',Y'.



## BLOQUE 2

Una llamada a la función  $K = FNT ( L9, X8, Y8, X9, Y9 )$  para inicializar las variables haciendose de la siguiente forma  $K = FNT ( 2, 0, 0, 0, 0 )$ .

## BLOQUE 3

Una llamada a la función  $K = FNT ( L9, X8, Y8, X9, Y9 )$  mediante la cual coloca al lápiz del plotter en el origen, en espera de datos.

## BLOQUE 4

L=1 Indicarà a los bloques posteriores que estarà en espera del L=0 para la grabación de los datos en el FICHERO 1.

## BLOQUE 5

Se introducen los puntos de la figura a dibujar .

VARIABLES:

I: contiene el número de puntos + 1 .

E1: valor de X

E2: valor de Y

E3: valor de Z

## BLOQUE 6

Se imprimen los puntos correspondiente de la figura en el FICHERO 2.

#### BLOQUE 7

Se leen los puntos correspondiente a la figura del FICHERO2 y pasa a la subrutina del bloque 8

#### BLOQUE 8

Se selecciona un cambio de coordenadas de tres a dos dimensiones.

VARIABLES:

X1: Valor de X en dos dimensiones

X2: Valor de Y en dos dimensiones

Y se retorna al programa.

#### BLOQUE 9

Se imprimen los nuevos puntos X , Y de la figura en el FICHERO3.

#### BLOQUE 10

L=0 Indicar  a los bloques posteriores que todos los datos que se lean seran imprimidos en el fichero de salida que es el FICHERO1.

#### BLOQUE 11

Se leen los puntos iniciales X3, X4 del FICHERO1.

VARIABLES:

X3: Corresponde X' cuyo valor inicialmente es 0

X4: Corresponde Y' cuyo valor inicialmente es 0

#### BLOQUE 12

Explora la existencia de final de fichero. En caso afirmativo pasa al bloque 4. En caso negativo al bloque 13.

#### BLOQUE 13

Se leen los puntos del X , Y del FICHERO 1.

#### BLOQUE 14

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante  $K = \text{FNT} ( 3, 0, 0, 0, 0 )$  donde se vuelca las variables en un buffer hasta un final de fichero.

#### BLOQUE 15

Termina el programa.

#### BLOQUE 15

Explora los límites que no puede exceder la X ni la Y . En caso afirmativo pasa al bloque 17 . En caso negativo al bloque 18.

El valor de X no puede exceder más de 256 y menos de 0 y lo mismo ocurre para la Y .

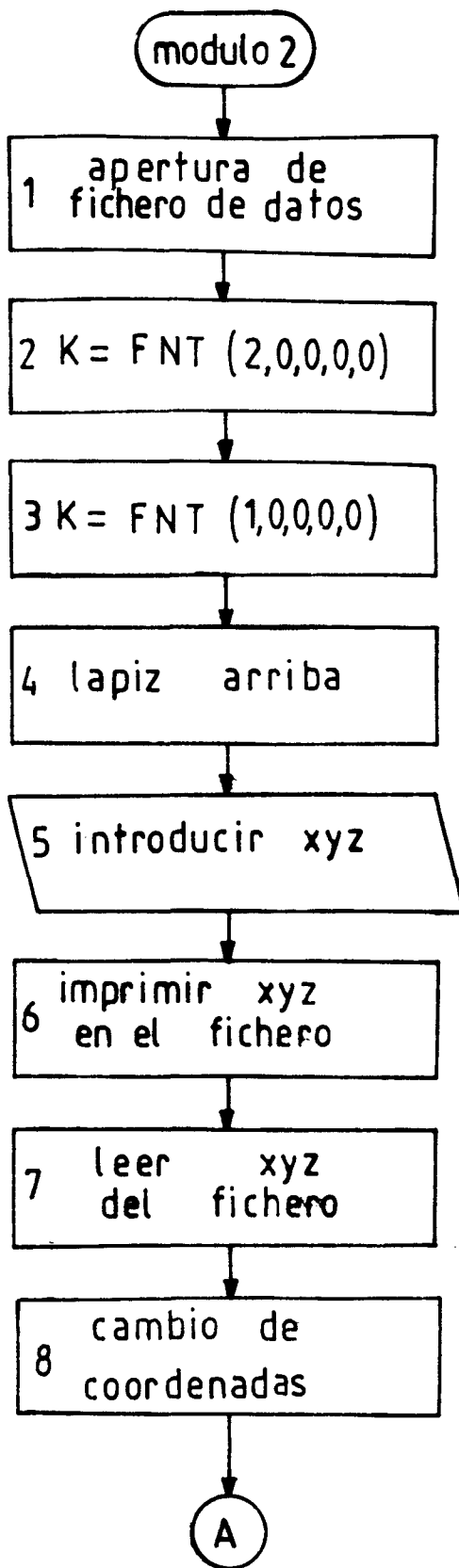
No pueden exceder de este valor porque son los límites máximos del plotter.

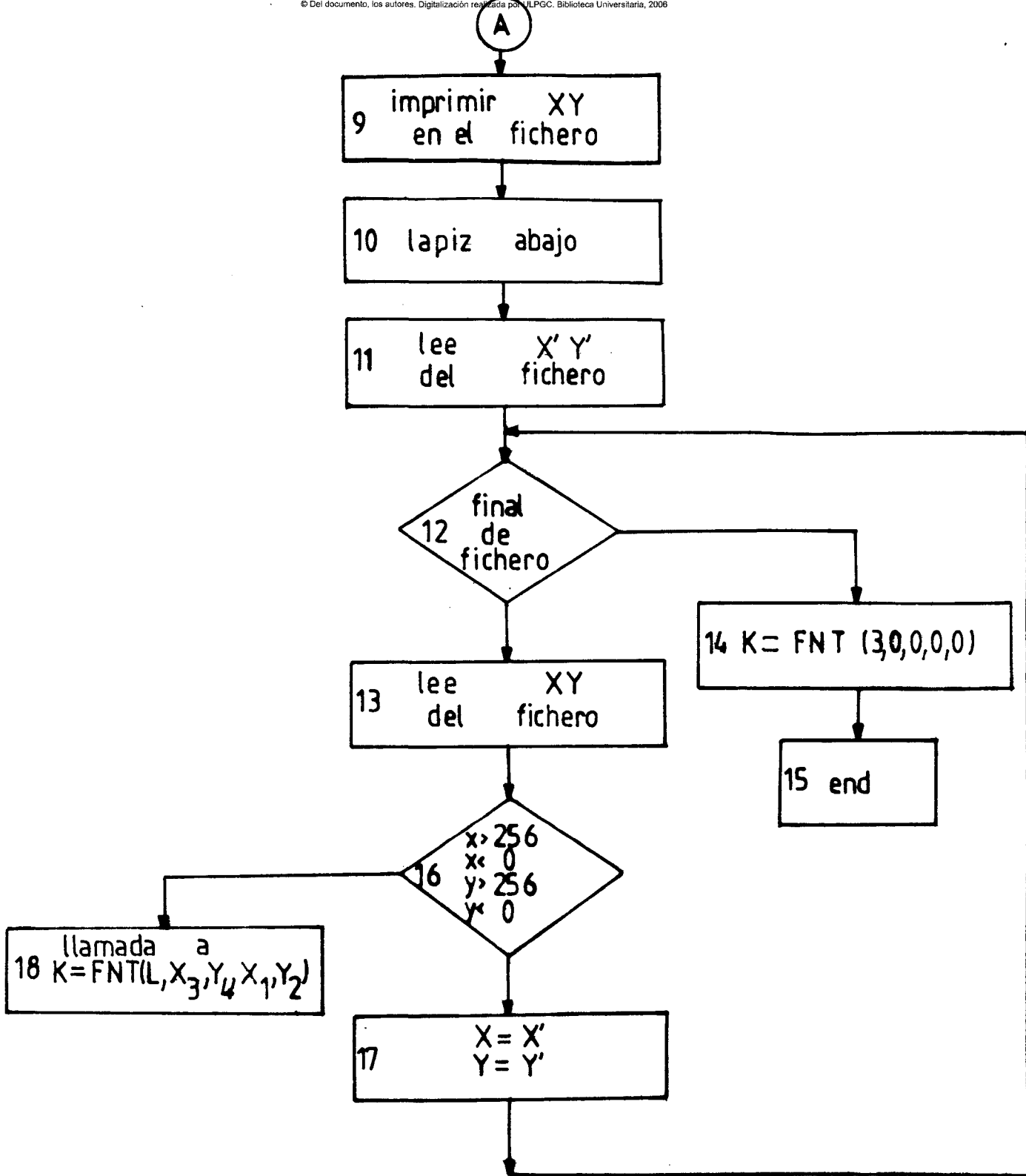
#### BLOQUE 17

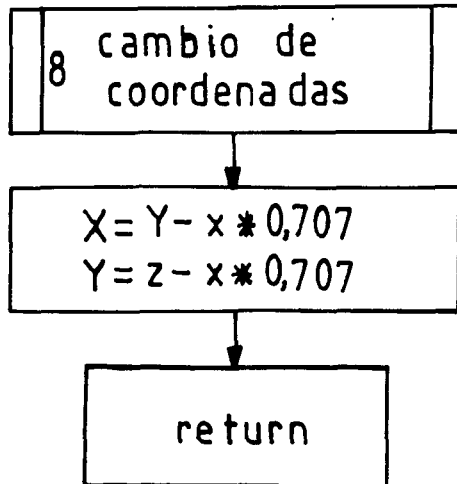
Las variable X tomará el valor X' y la variable Y tomará el valor Y'. Y se vuelve al bloque 12.

#### BLOQUE 18

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante la función  $K = \text{FNT} ( L, X3, X4, X1, X2 )$  donde las variables son equivalentes.







## 2.5 MODULO 3. TRASLACION DEL DIBUJO

Este módulo permite la traslación del dibujo cualquiera que sea la forma.

### EXPLICACION DEL ORGANIGRAMA

#### BLOQUE 1.

FICHERO 2: OUTPUT1 Localiza la posición de grabación correspondiente a la introducción de datos en tres dimensiones de la figura .

FICHERO 3: INPUT2 localiza la posición de grabación correspondiente a la transformación del fichero OUTPUT1 en dos dimensiones.

FICHERO 4: TRAS Localiza la posición de grabación correspondiente a la transformación del fichero INPUT2 en dos dimensiones.

FICHERO 1: GRAPH14 Localiza la posición de grabación correspondiente a la transformación del fichero INPUT1 mediante la función  $K = FNT ( L, X3, X4, X1, X2 )$  .

Inicialización de las variables X1, X2, X3, X4.

VARIABLES:

L: Indica la posición de el lápiz del plotter.



L=0 Lápiz abajo

L=1 Lápiz arriba

X1,X2: Corresponde al valor inicial de X,Y.

X3,X4: Corresponde al valor final de X',Y'.

## BLOQUE 2

Introducción de los factores de traslación de X , Y , Z que son L , M ,N.

## BLOQUE 3

L=1 Indicará a los bloques posteriores que estará en espera del L=0 para la grabación de los datos en el FICHERO 1.

## BLOQUE 4

Se leen los valores X, Y , Z del FICHERO 2.

## BLOQUE 5

En este bloque nos dice la cantidad de puntos que compone la figura . Si la cantidad de puntos es menor que la que corresponde nos dara error.

## BLOQUE 6

Una llamada a la función  $K = FNA ( P,$

$X(*), Y(*), Z(*), L, M, N$  )

#### BLOQUE 7

Definición de la matriz U. Cuyo valor sera los puntos X, Y, Z de la figura.

$$U = \begin{vmatrix} X & Y & Z & 1 \end{vmatrix}$$

#### BLOQUE 8

Definición de la matriz T. Cuyo valor será los parametros correspondiente a la traslación.

$$T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ L & m & n & 1 \end{vmatrix}$$

#### BLOQUE 9

Definición de la matriz V que tendra de valor el producto vectorial de U \* T.

$$V = U * T$$

#### BLOQUE 10

Finalización de la función  $K = FNA ( P, X(*), Y(*), Z(*), L, M, N )$

#### BLOQUE 11

Se devuelve el control al programa donde se hizo la llamada.

#### BLOQUE 12

Se imprime los puntos correspondientes de la figura en el FICHERO 3

#### BLOQUE 13

Se selecciona un cambio de coordenadas de tres a dos dimensiones.

#### VARIABLES:

X1: Valor de X en dos dimensiones.

X2: Valor de Y en dos dimensiones.

#### BLOQUE 14

Se imprimen los nuevos puntos X, Y de la figura en el FICHERO 4.

#### BLOQUE 15

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  para inicializar las variables haciendose de la siguiente forma  $K = \text{FNT} ( 2, 0, 0, 0, 0 )$ .

#### BLOQUE 16

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante la cual coloca el lápiz del plotter en el origen en espera de datos.

#### BLOQUE 17

Explora los límites que no puede exceder la X ni la Y. En caso afirmativo pasa la bloque 18 . En caso negativo al bloque 19.

El valor de X no puede exceder más de 256 y menos de 0 y los mismo ocurre para la Y.

No pueden exceder de este valor porque son los límites máximos del plotter.

#### BLOQUE 18

Las variable X tomará el valor X' y la variable Y tomará el valor Y' . Y se vuelve al bloque 17.

#### BLOQUE 19

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante la función  $K = \text{FNT} ( L, X3, X4, X1, X2 )$  donde las variables son equivalentes.

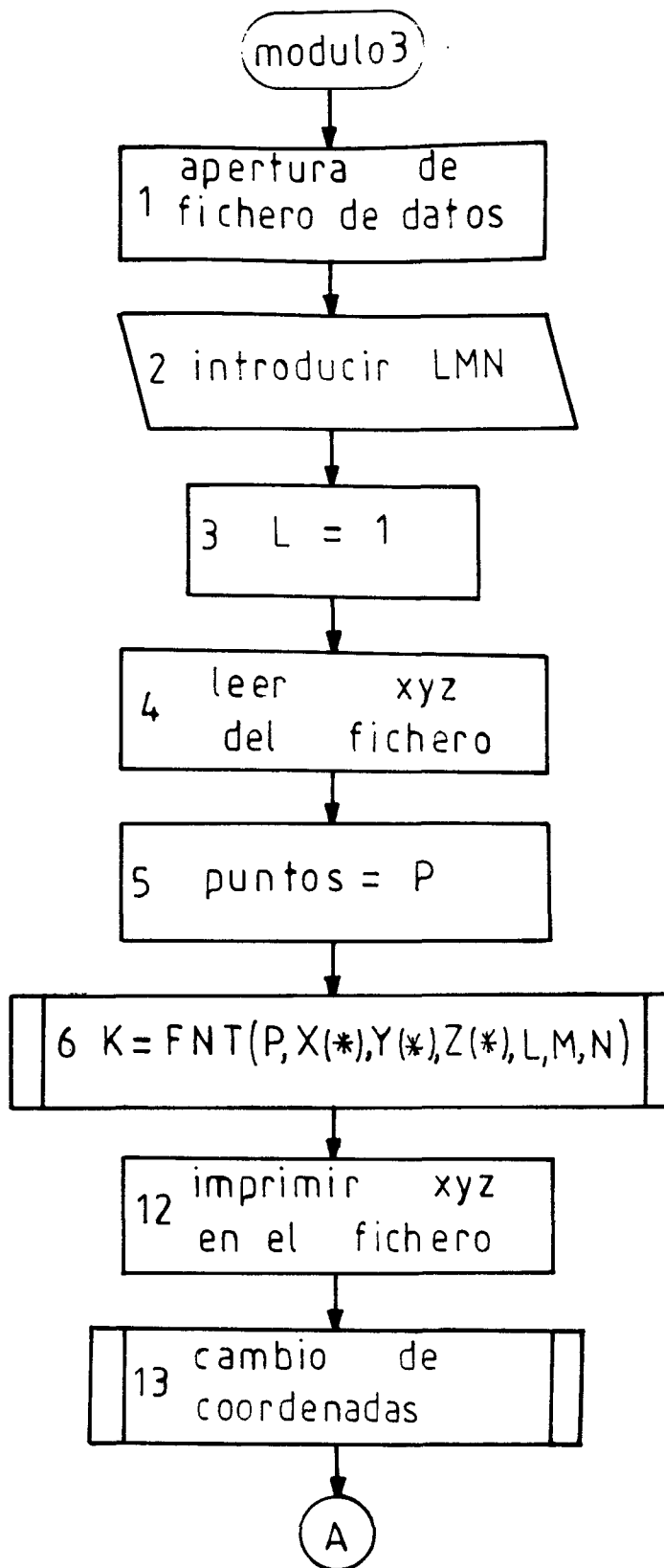
## BLOQUE 20

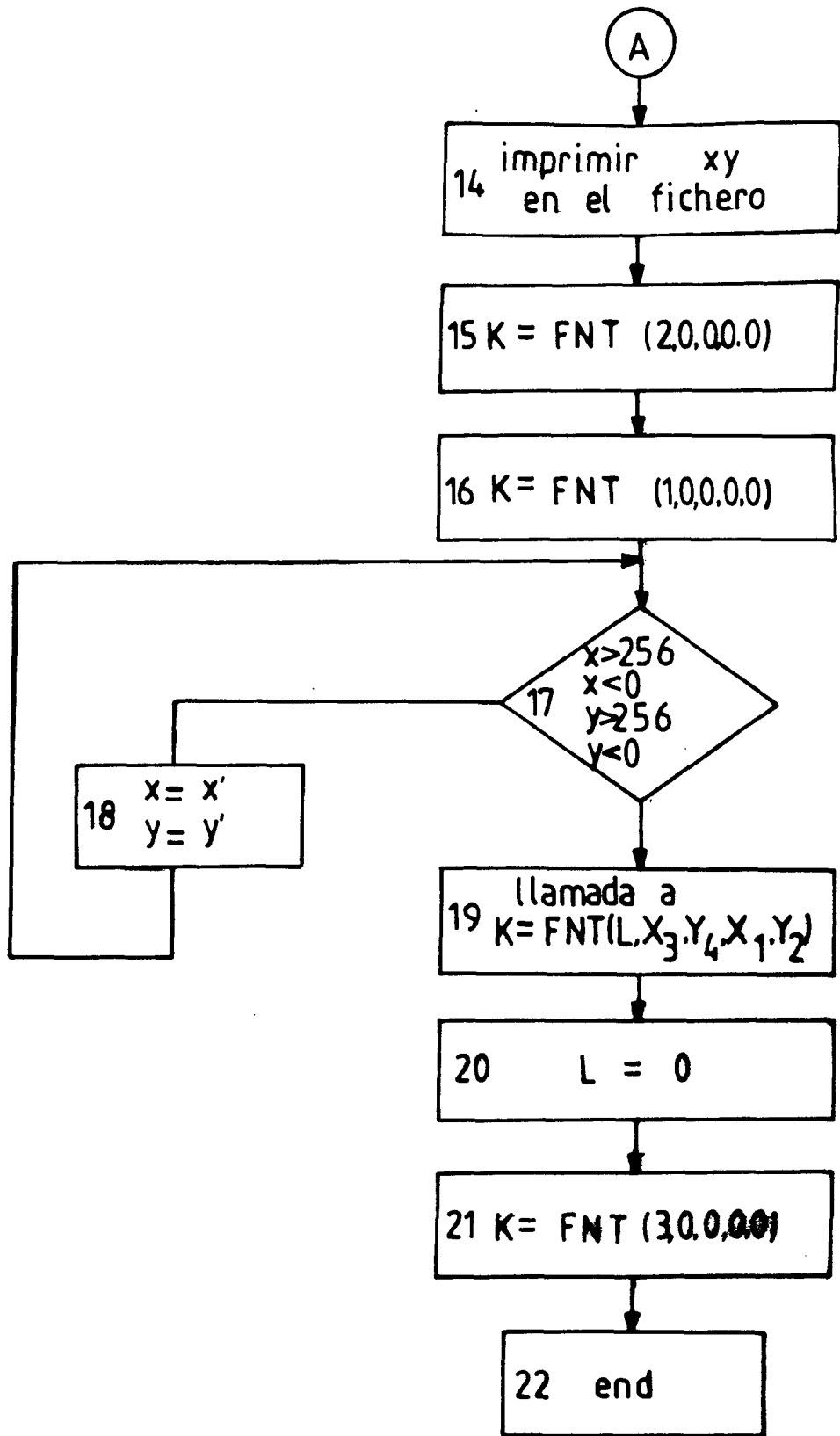
L=0 indica a los bloques posteriores que todos los datos que se lean serán imprimidos en el fichero de salida que es el FICHERO 1.

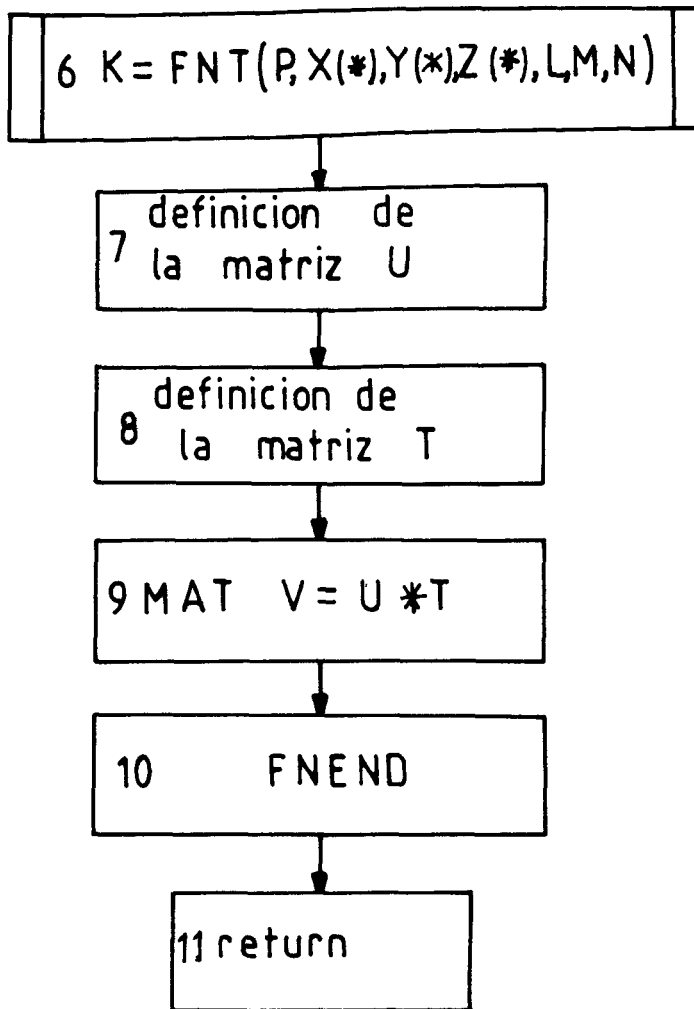
Una llamada a la función  $K = FNT ( L9, X8, Y8, X9, Y9 )$  mediante  $K = FNT ( 3, 0, 0, 0, 0 )$  donde se vuelca las variables en un buffer hasta un final de fichero.

## BLOQUE 22

Termina el programa.









## 2.6 MODULO 4. ESTRECHAMIENTO DEL DIBUJO

Este módulo permite la estrechamiento del dibujo cualquiera que sea la forma.

### EXPLICACION DEL ORGANIGRAMA

#### BLOQUE 1.

FICHERO 2: OUTPUT1 Localiza la posición de grabación correspondiente a la introducción de datos en tres dimensiones de la figura .

FICHERO 3: INPUT3 localiza la posición de grabación correspondiente a la transformación del fichero OUTPUT1 en dos dimensiones.

FICHERO 4: ESTR Localiza la posición de grabación correspondiente a la transformación del fichero INPUT2 en dos dimensiones.

FICHERO 1: GRAPH15 Localiza la posición de grabación correspondiente a la transformación del fichero INPUT1 mediante la función  $K = FNT ( L, X3, X4, X1, X2 )$  .

Inicialización de las variables X1, X2, X3, X4.

VARIABLES:

L: Indica la posición de el lápiz del plotter.

L=0 Lápiz abajo

L=1 Lápiz arriba

X1,X2: Corresponde al valor inicial de X,Y.

X3,X4: Corresponde al valor final de X',Y'.

## BLOQUE 2

Introducción de los factores de estrechamiento de X ,  
Y , Z que son A , E , J , S.

## BLOQUE 3

L=1 Indicará a los bloques posteriores que estará en  
espera del L=0 para la grabación de los datos en el  
FICHERO 1.

## BLOQUE 4

Se leen los valores X , Y , Z del FICHERO 2.

## BLOQUE 5

En este bloque nos dice la cantidad de puntos que

compone la figura . Si la cantidad de puntos es menor que la que corresponde nos dara error.

#### BLOQUE 6

Una llamada a la función  $K = FNA ( P, X(*), Y(*), Z(*), A, E, J, S )$

#### BLOQUE 7

Definición de la matriz U. Cuyo valor sera los puntos X, Y, Z de la figura.

$$U = \begin{vmatrix} x & y & z & 1 \end{vmatrix}$$

#### BLOQUE 8

Definición de la matriz T . Cuyo valor será los parametros correspondiente a la

$$T = \begin{vmatrix} a & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & j & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

### BLOQUE 9

Definición de la matriz  $V$  que tendrá de valor el producto vectorial de  $U * T$ .

$$V = U * T$$

### BLOQUE 10

Finalización de la función  $K = FNA ( P, X(*), Y(*), Z(*), A, E, J, S )$

### BLOQUE 11

Se devuelve el control al programa donde se hizo la llamada.

#### BLOQUE 12

Se imprime los puntos correspondientes de la figura en el FICHERO 3

#### BLOQUE 13

Se selecciona un cambio de coordenadas de tres a dos dimensiones.

#### VARIABLES:

X1: Valor de X en dos dimensiones.

X2: Valor de Y en dos dimensiones.

#### BLOQUE 14

Se imprimen los nuevos puntos X, Y de la figura en el FICHERO 4.

#### BLOQUE 15

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  para inicializar las variables haciendose de la siguiente forma  $K = \text{FNT} ( 2, 0, 0, 0, 0 )$ .

#### BLOQUE 16

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$

mediante la cual coloca el lápiz del plotter en el origen en espera de datos.

#### BLOQUE 17

Explora los límites que no puede exceder la X ni la Y. En caso afirmativo pasa al bloque 18. En caso negativo al bloque 19.

El valor de X no puede exceder más de 256 y menos de 0 y lo mismo ocurre para la Y.

No pueden exceder de este valor porque son los límites máximos del plotter.

#### BLOQUE 18

Las variables X tomará el valor X' y la variable Y tomará el valor Y'. Y se vuelve al bloque 17.

#### BLOQUE 19

Una llamada a la función  $K = FNT ( L9, X8, Y8, X9, Y9 )$  mediante la función  $K = FNT ( L, X3, X4, X1, X2 )$  donde las variables son equivalentes.

#### BLOQUE 20

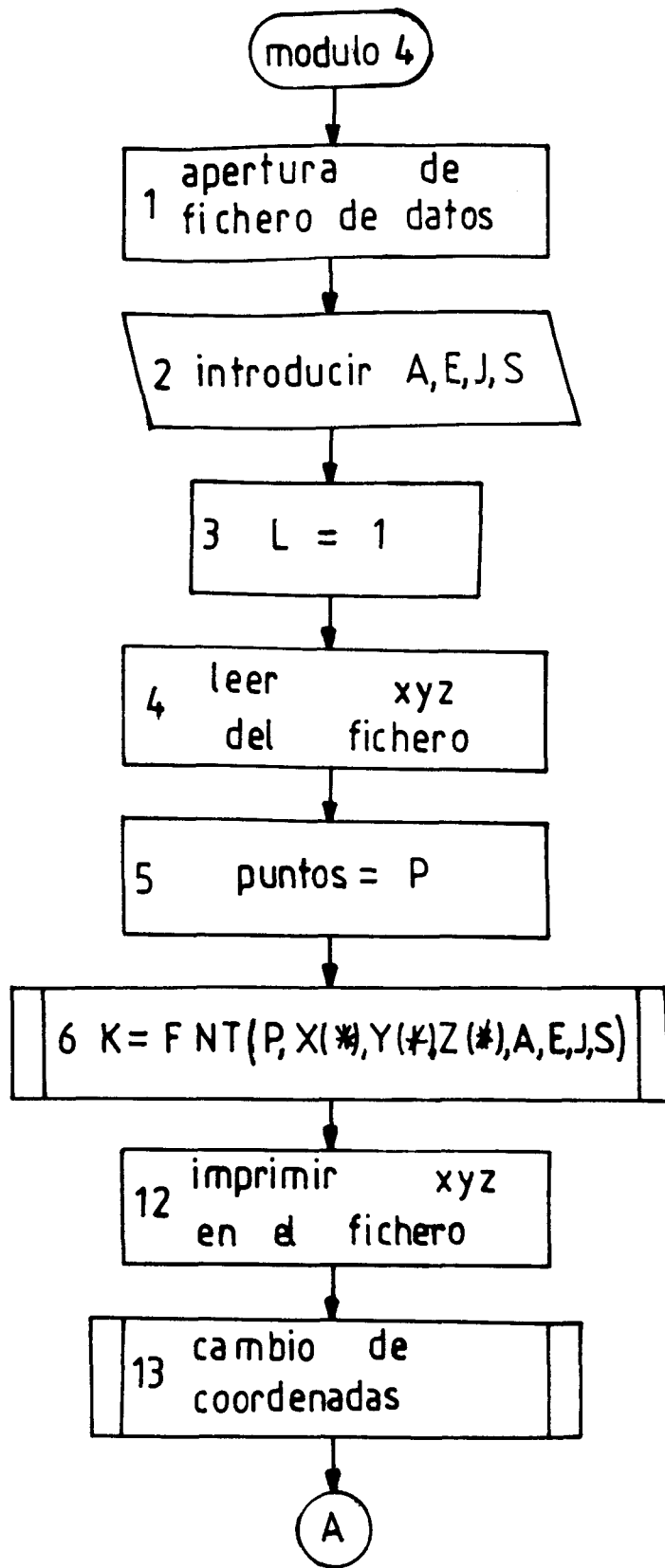
L=0 indica a los bloques posteriores que todos los datos que se lean serán imprimidos en el fichero de salida que es el FICHERO 1.

## BLOQUE 21

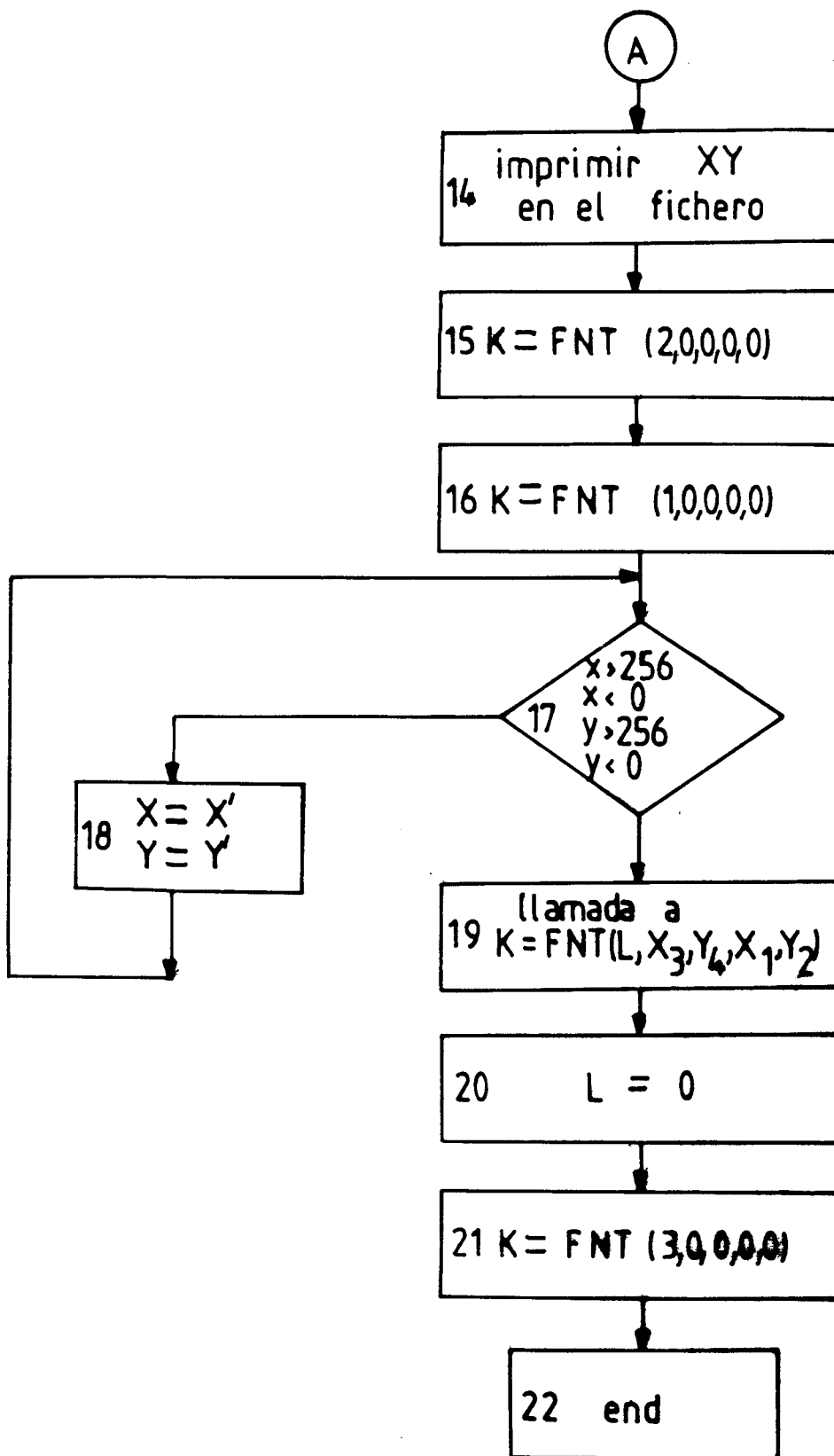
Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$   
mediante  $K = \text{FNT} ( 3, 0, 0, 0, 0 )$  donde se vuelca las  
variables en un buffer hasta un final de fichero.

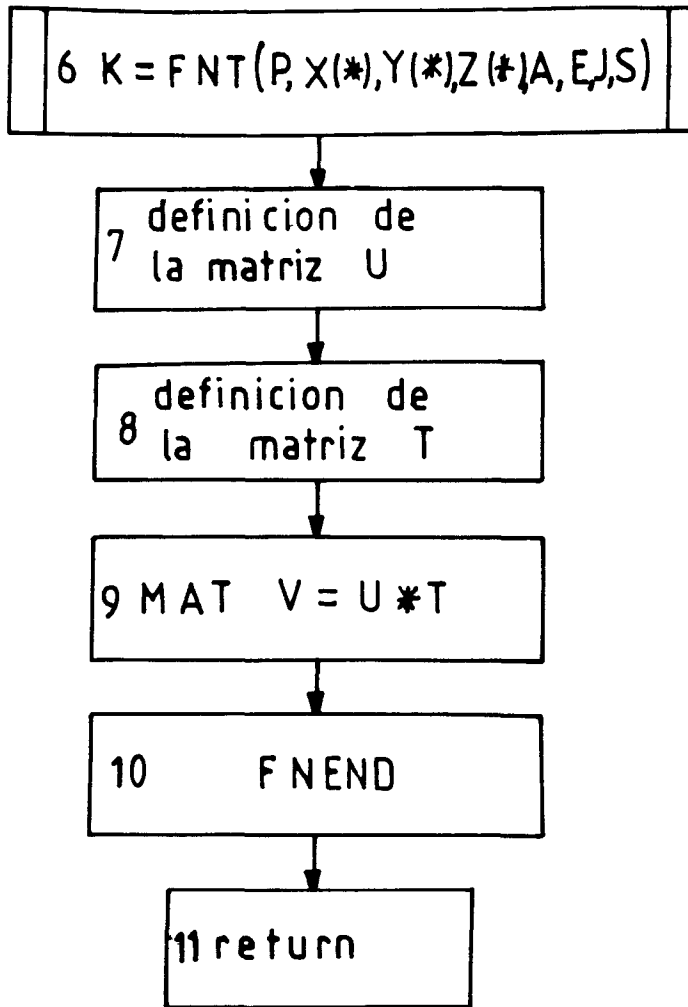
## BLOQUE 22

Termina el programa.









## 2.7 MODULO 5. ROTACION DEL DIBUJO

Este módulo permite la rotación del dibujo cualquiera que sea la forma.

### EXPLICACION DEL ORGANIGRAMA

#### BLOQUE 1.

FICHERO 2: OUTPUT1 Localiza la posición de grabación correspondiente a la introducción de datos en tres dimensiones de la figura .

FICHERO 3: INPUT2 localiza la posición de grabación correspondiente a la transformación del fichero OUTPUT1 en dos dimensiones.

FICHERO 4: ROT Localiza la posición de grabación correspondiente a la transformación del fichero INPUT2 en dos dimensiones.

FICHERO 1: GRAPH16 Localiza la posición de grabación correspondiente a la transformación del fichero INPUT1 mediante la función  $K = FNT ( L, X3, X4, X1, X2 )$  .

Inicialización de las variables X1, X2, X3, X4.

#### VARIABLES:

L: Indica la posición de el lápiz del plotter.

L=0 Lápiz abajo

L=1 Lápiz arriba

X1,X2: Corresponde al valor inicial de X,Y.

X3,X4: Corresponde al valor final de X',Y'.

#### BLOQUE 2

Introducción de los factores de rotación de X , Y , Z que son N1,N2,N3,T1.

#### BLOQUE 3

L=1 Indicará a los bloques posteriores que estará en espera del L=0 para la grabación de los datos en el FICHERO 1.

#### BLOQUE 4

Se leen los valores X, Y , Z del FICHERO 2.

#### BLOQUE 5

En este bloque nos dice la cantidad de puntos que

compone la figura . Si la cantidad de puntos es menor que la que corresponde nos dara error.

#### BLOQUE 6

Una llamada a la función  $K = FNA ( P, X(*), Y(*), Z(*), N1, N2, N3, T1)$

#### BLOQUE 7

Definición de la matriz U. Cuyo valor sera los puntos X, Y, Z de la figura.

$$U = \begin{vmatrix} X, Y, Z, 1 \end{vmatrix}$$

#### BLOQUE 8

Definición de la matriz T . Cuyo valor será los parametros correspondiente a la ..

$$\begin{array}{cccc|c}
 n_3^2 + (1 - n_1^2) \cos \theta & n_1 n_2 (1 - \cos \theta) + n_3 \sin \theta & n_1 n_3 (1 - \cos \theta) - n_2 \sin \theta & 0 & 0 \\
 n_1 n_2 (1 - \cos \theta) - n_3 \sin \theta & n_2^2 + (1 - n_1^2) \cos \theta & n_2 n_3 (1 - \cos \theta) + n_1 \sin \theta & 0 & 0 \\
 n_1 n_2 (1 - \cos \theta) + n_3 \sin \theta & n_2 n_3 (1 - \cos \theta) - n_1 \sin \theta & n_3^2 + (1 - n_1^2) \cos \theta & 0 & 0 \\
 0 & 0 & 0 & 0 & 1
 \end{array}$$

### BLOQUE 9

Definición de la matriz  $V$  que tendrá de valor el producto vectorial de  $U * T$ .

$$V = U * T$$

### BLOQUE 10

Finalización de la función  $K = FNA ( P, X(*), Y(*), Z(*), N1, N2, N3, T1 )$

#### BLOQUE 11

Se devuelve el control al programa donde se hizo la llamada.

#### BLOQUE 12

Se imprime los puntos correspondientes de la figura en el FICHERO 3

#### BLOQUE 13

Se selecciona un cambio de coordenadas de tres a dos dimensiones.

#### VARIABLES:

X1: Valor de X en dos dimensiones.

X2: Valor de Y en dos dimensiones.

#### BLOQUE 14

Se imprimen los nuevos puntos X, Y de la figura en el FICHERO 4.

#### BLOQUE 15

Una llamada a la función  $K = FNT (.L9, X8, Y8, X9, Y9)$  para inicializar las variables haciendose de la siguiente forma  $K = FNT ( 2, 0, 0, 0, 0 )$ .

## BLOQUE 16

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante la cual coloca el lápiz del plotter en el origen en espera de datos.

## BLOQUE 17

Explora los límites que no puede exceder la X ni la Y. En caso afirmativo pasa al bloque 18. En caso negativo al bloque 19.

El valor de X no puede exceder más de 256 y menos de 0 y lo mismo ocurre para la Y.

No pueden exceder de este valor porque son los límites máximos del plotter.

## BLOQUE 18

Las variables X tomará el valor X' y la variable Y tomará el valor Y'. Y se vuelve al bloque 17.

## BLOQUE 19

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante la función  $K = \text{FNT} ( L, X3, X4, X1, X2 )$  donde las variables son equivalentes.

## BLOQUE 20

$L=0$  indica a los bloques posteriores que todos los



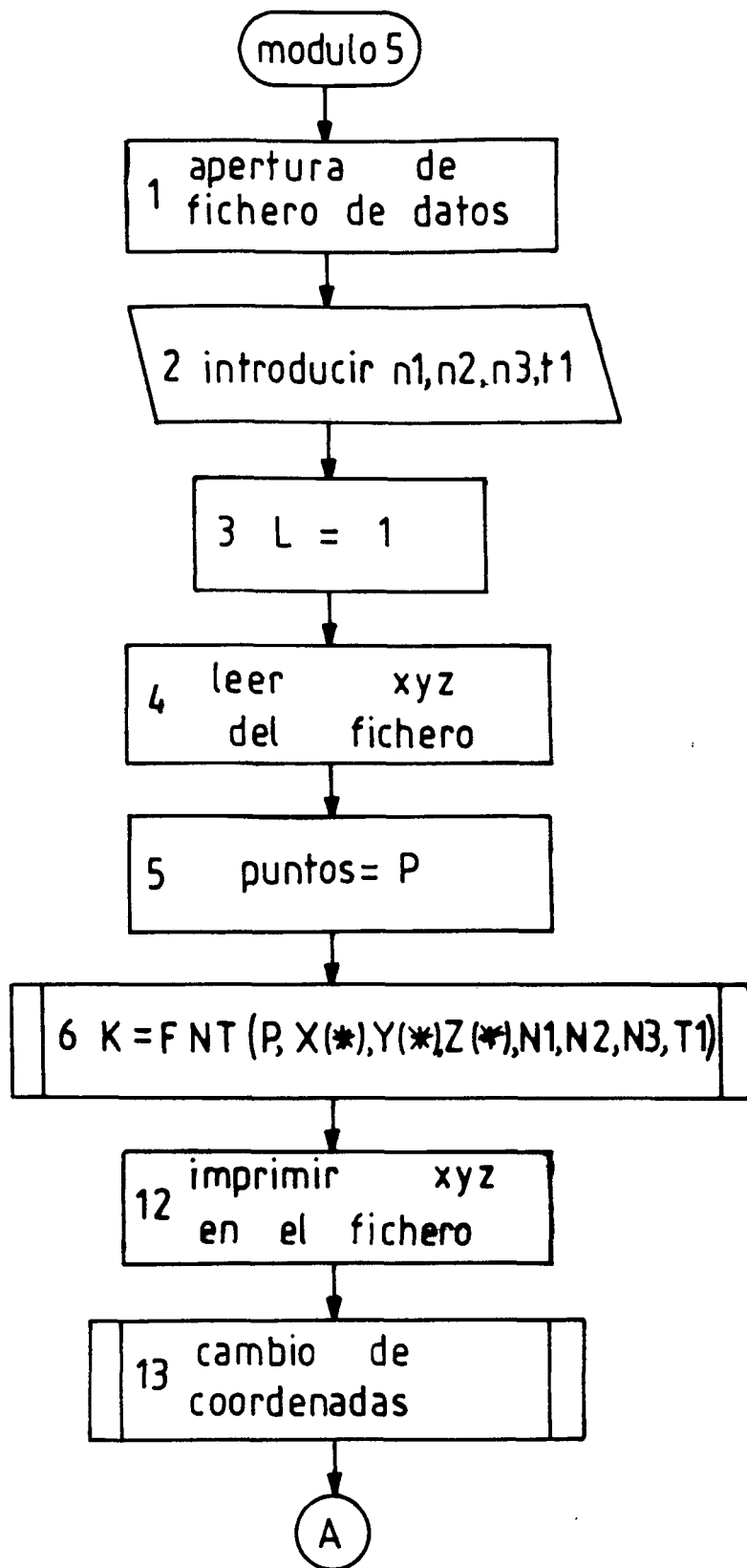
datos que se lean serán imprimidos en el fichero de salida que es el FICHERO 1.

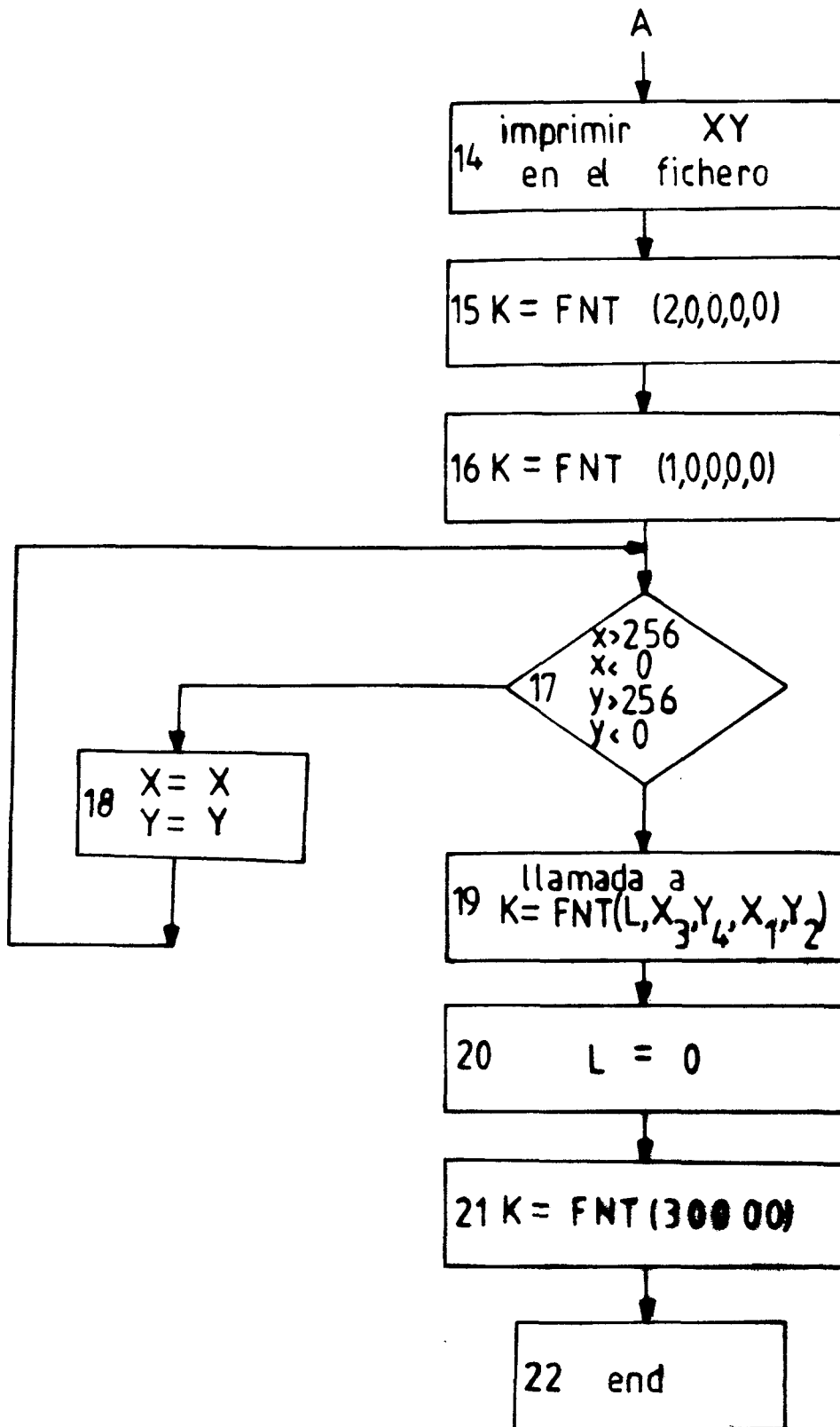
#### BLOQUE 21

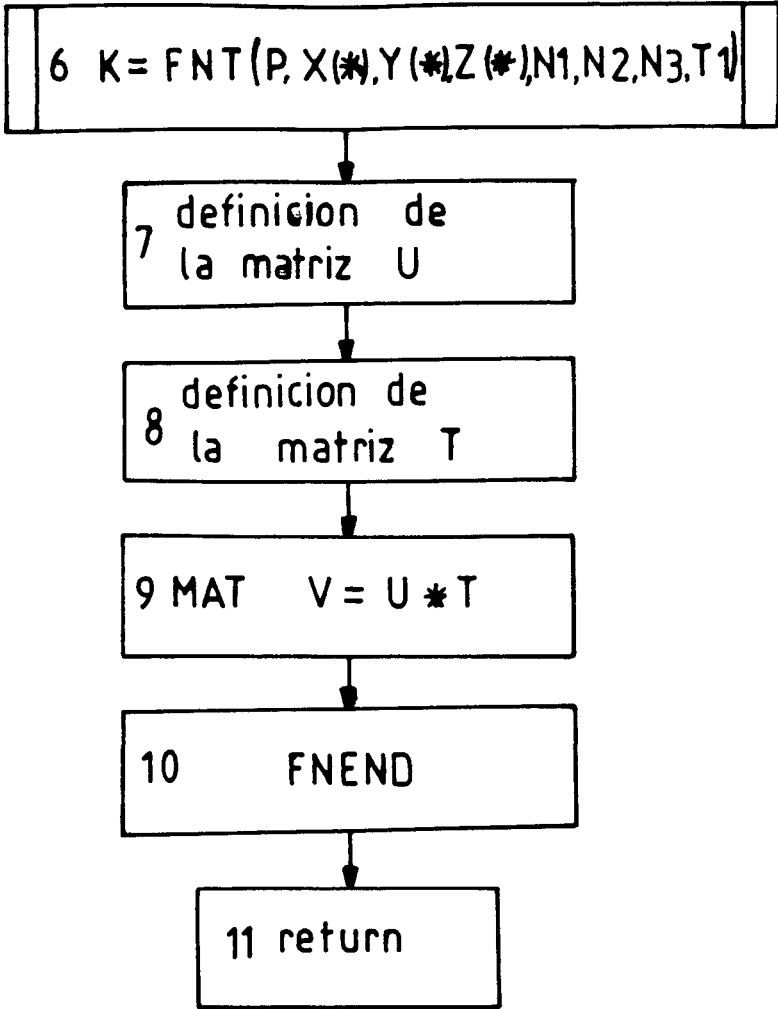
Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante  $K = \text{FNT} ( 3, 0, 0, 0, 0 )$  donde se vuelca las variables en un buffer hasta un final de fichero.

#### BLOQUE 22

Termina el programa.







## 2.8 MODULO 6. REFLEXION DEL DIBUJO

Este módulo permite la reflexión del dibujo cualquiera que sea la forma.

### EXPLICACION DEL ORGANIGRAMA

#### BLOQUE 1.

FICHERO 2: OUTPUT1 Localiza la posición de grabación correspondiente a la introducción de datos en tres dimensiones de la figura .

FICHERO 3: INPUT5 localiza la posición de grabación correspondiente a la transformación del fichero OUTPUT1 en dos dimensiones.

FICHERO 4: REFLE Localiza la posición de grabación correspondiente a la transformación del fichero INPUT2 en dos dimensiones.

FICHERO 1: GRAPH17 Localiza la posición de grabación correspondiente a la transformación del fichero INPUT1 mediante la función  $K = FNT ( L, X3, X4, X1, X2 )$  .

Inicialización de las variables X1, X2, X3, X4.

VARIABLES:

L: Indica la posición de el lápiz del plotter.

L=0 Lápiz abajo

L=1 Lápiz arriba

X1,X2: Corresponde al valor inicial de X,Y.

X3,X4: Corresponde al valor final de X',Y'.

## BLOQUE 2

Introducción de los factores de reflexion de X , Y ,  
Z que es N.

## BLOQUE 3

L=1 Indicará a los bloques posteriores que estará en  
espera del L=0 para la grabación de los datos en el  
FICHERO 1.

## BLOQUE 4

Se leen los valores X, Y , Z del FICHERO 2.

## BLOQUE 5

En este bloque nos dice la cantidad de puntos que compone la figura . Si la cantidad de puntos es menor que la que corresponde nos dara error.

#### BLOQUE 6

Una llamada a la función  $K = FNA ( P, X(*), Y(*), Z(*), N )$

#### BLOQUE 7

Definición de la matriz U. Cuyo valor sera los puntos X, Y, Z de la figura.

$$U = \begin{vmatrix} X, & Y, & Z, & 1 \end{vmatrix}$$

#### BLOQUE 8

Definición de la matriz T . Cuyo valor será los parametros correspondiente a la

$$T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad T = \begin{vmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

### BLOQUE 9

Definición de la matriz  $V$  que tendrá de valor el producto vectorial de  $U * T$ .

$$V = U * T$$

### BLOQUE 10

Finalización de la función  $K = FNA ( P, X(*), Y(*), Z(*), N )$



#### BLOQUE 11

Se devuelve el control al programa donde se hizo la llamada.

#### BLOQUE 12

Se imprime los puntos correspondientes de la figura en el FICHERO 3

#### BLOQUE 13

Se selecciona un cambio de coordenadas de tres a dos dimensiones.

#### VARIABLES:

X1: Valor de X en dos dimensiones.

X2: Valor de Y en dos dimensiones.

#### BLOQUE 14

Se imprimen los nuevos puntos X, Y de la figura en el FICHERO 4.

#### BLOQUE 15

Una llamada a la función  $K = FNT ( L9, X8, Y8, X9, Y9 )$  para inicializar las variables haciendose de la siguiente forma  $K = FNT ( 2, 0, 0, 0, 0 )$ .

#### BLOQUE 16

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante la cual coloca el lápiz del plotter en el origen en espera de datos.

#### BLOQUE 17

Explora los límites que no puede exceder la X ni la Y. En caso afirmativo pasa al bloque 18. En caso negativo al bloque 19.

El valor de X no puede exceder más de 256 y menos de 0 y lo mismo ocurre para la Y.

No pueden exceder de este valor porque son los límites máximos del plotter.

#### BLOQUE 18

Las variables X tomará el valor X' y la variable Y tomará el valor Y'. Y se vuelve al bloque 17.

#### BLOQUE 19

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante la función  $K = \text{FNT} ( L, X3, X4, X1, X2 )$  donde las variables son equivalentes.

#### BLOQUE 20

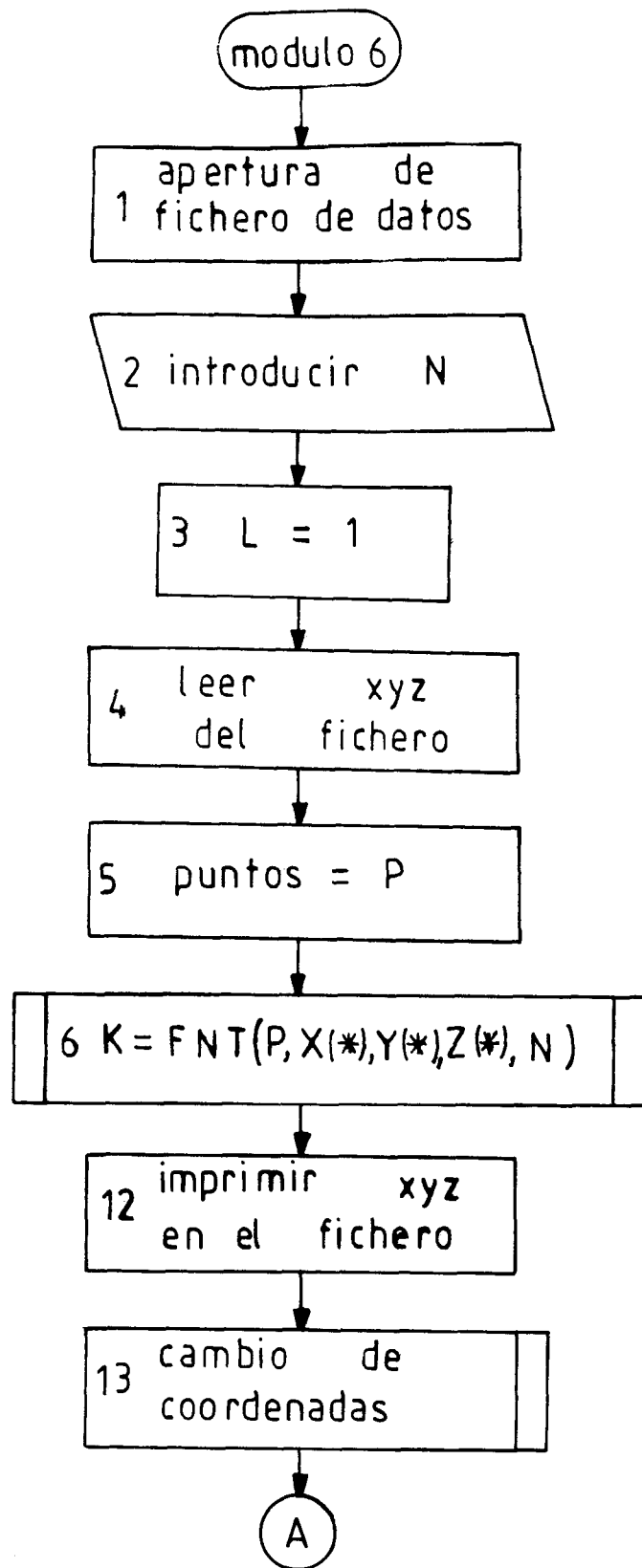
$L=0$  indica a los bloques posteriores que todos los datos que se lean serán imprimidos en el fichero de salida que es el FICHERO 1.

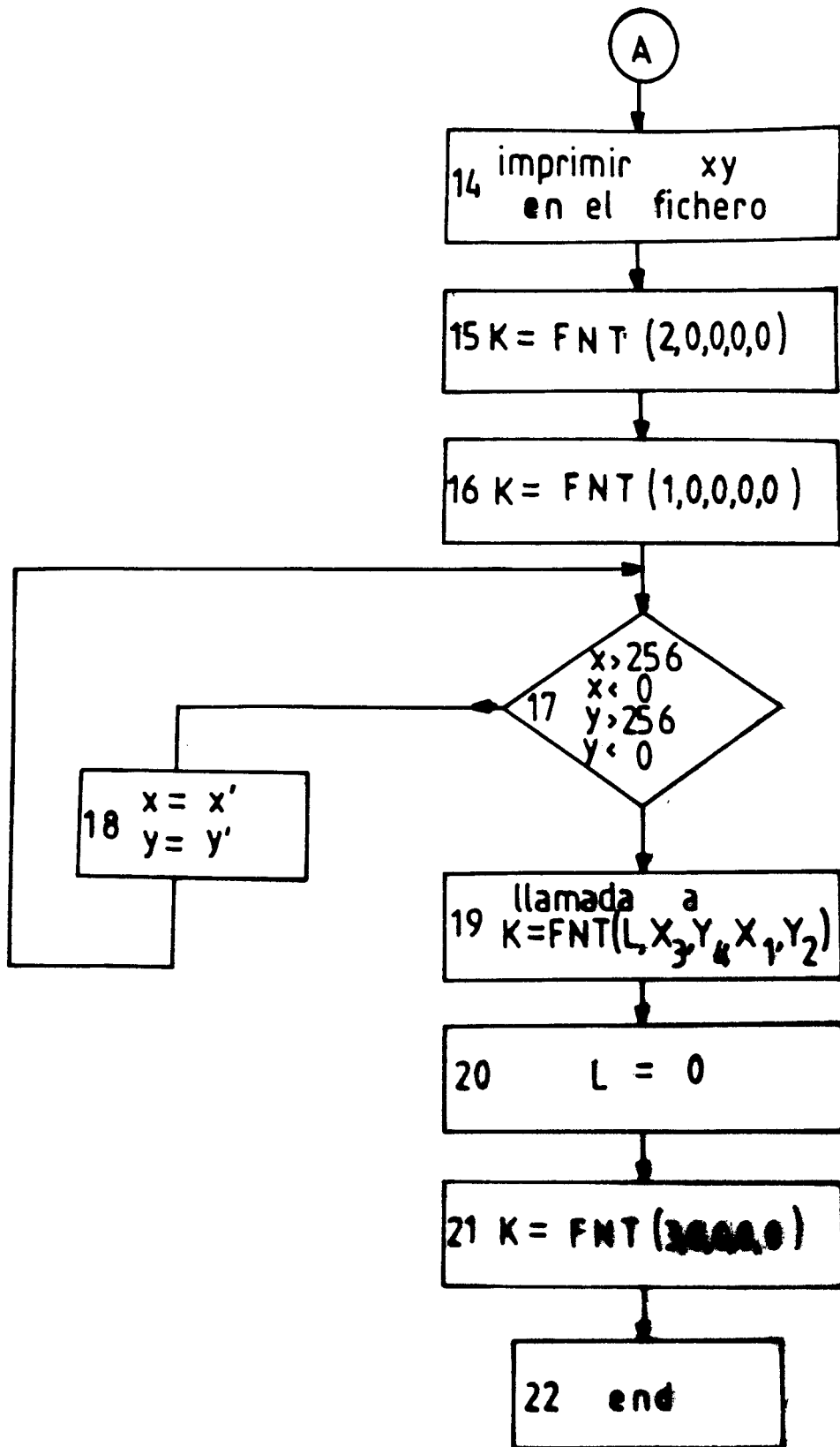
## BLOQUE 21

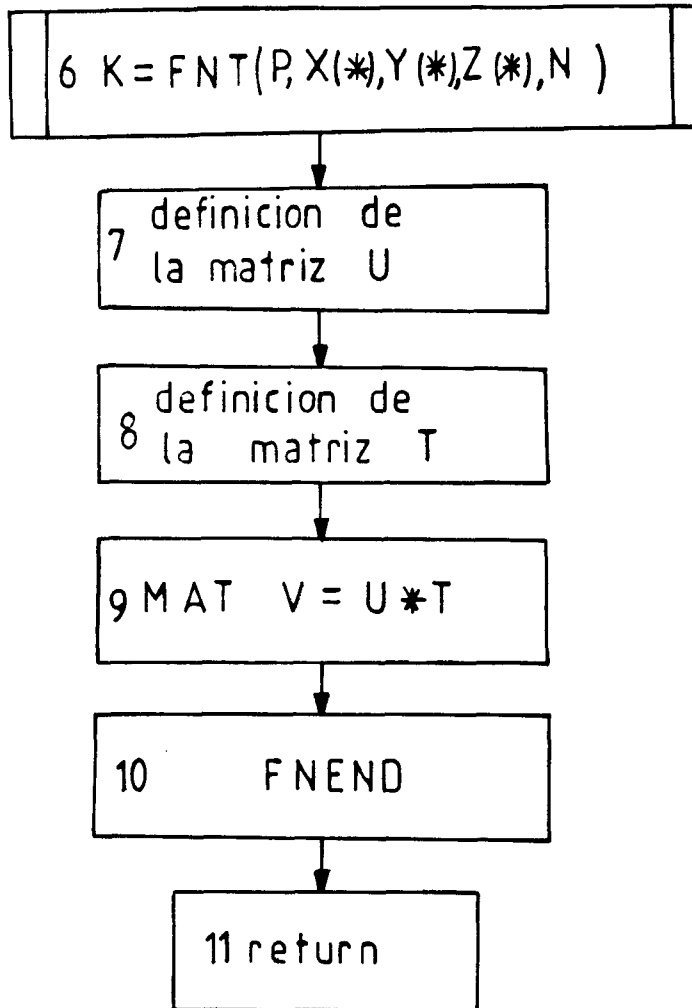
Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$   
mediante  $K = \text{FNT} ( 3, 0, 0, 0, 0 )$  donde se vuelca las  
variables en un buffer hasta un final de fichero.

## BLOQUE 22

Termina el programa.







## 2.9 MODULO 7. PROYECCION DIMETRICA DEL DIBUJO

Este modulo permite la proyección dimetrica del dibujo cualquiera que sea la forma.

### EXPLICACION DEL ORGANIGRAMA

#### BLOQUE 1.

FICHERO 2: OUTPUT1 Localiza la posición de grabación correspondiente a la introducción de datos en tres dimensiones de la figura .

FICHERO 3: INPUT6 localiza la posición de grabación correspondiente a la transformación del fichero OUTPUT1 en dos dimensiones.

FICHERO 4: DIME Localiza la posición de grabación correspondiente a la transformación del fichero INPUT2 en dos dimensiones.

FICHERO 1: GRAPH18 Localiza la posición de grabación correspondiente a la transformación del fichero INPUT1 mediante la función  $K = FNT ( L, X3, X4, X1, X2 )$  .

Inicialización de las variables X1,X2,X3,X4.

VARIABLES:

L: Indica la posición de el lápiz del plotter.

L=0 Lápiz abajo

L=1 Lápiz arriba

X1,X2: Corresponde al valor inicial de X,Y.

X3,X4: Corresponde al valor final de X',Y'.

## BLOQUE 2

L=1 Indicará a los bloques posteriores que estará en espera del L=0 para la grabación de los datos en el FICHERO 1.

## BLOQUE 3

Se leen los valores X, Y , Z del FICHERO 2.

## BLOQUE 4

En este bloque nos dice la cantidad de puntos que compone la figura . Si la cantidad de puntos es menor que la que corresponde nos dara error.



## BLOQUE 5

Una llamada a la función  $K = FNA ( P, X(*), Y(*), Z(*) )$

## BLOQUE 6

Definición de la matriz U. Cuyo valor sera los puntos X, Y, Z de la figura.

$$U = \begin{vmatrix} x & y & z & 1 \end{vmatrix}$$

## BLOQUE 7

Definición de la matriz T. Cuyo valor será los parametros correspondiente a la proyeccion dimetrica.

$$T = \begin{vmatrix} 0.707107 & 0.408248 & -0.577353 & 0 \\ 0 & 0.816597 & 0.577343 & 0 \\ 0.707107 & -0.408248 & 0.577353 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

## BLOQUE 8

Definición de la matriz  $V$  que tendrá de valor el producto vectorial de  $U * T$ .

$$V = U * T$$

## BLOQUE 9

Finalización de la función  $K = FNA ( P, X(*), Y(*), Z(*) )$

## BLOQUE 10

Se devuelve el control al programa donde se hizo la llamada.

## BLOQUE 11

Se imprime los puntos correspondientes de la figura en el FICHERO 3

## BLOQUE 12

Se selecciona un cambio de coordenadas de tres a dos dimensiones.

#### VARIABLES:

X1: Valor de X en dos dimensiones.

X2: Valor de Y en dos dimensiones.

#### BLOQUE 13

Se imprimen los nuevos puntos X, Y de la figura en el FICHERO 4.

#### BLOQUE 14

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  para inicializar las variables haciendose de la siguiente forma  $K = \text{FNT} ( 2, 0, 0, 0, 0 )$ .

#### BLOQUE 15

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante la cual coloca el lápiz del plotter en el origen en espera de datos.

#### BLOQUE 16

Explora los límites que no puede exceder la X ni la Y. En caso afirmativo pasa la bloque 178 . En caso negativo al bloque 18.

El valor de X no puede exceder más de 256 y menos de 0 y los mismo ocurre para la Y.

No pueden exceder de este valor porque son los

límites máximos del plotter.

#### BLOQUE 17

Las variable X tomará el valor X' y la variable Y tomara el valor Y'. Y se vuelve al bloque 16.

#### BLOQUE 18

Una llamada a la función  $K = FNT ( L9, X8, Y8, X9, Y9 )$  mediante la función  $K = FNT ( L, X3, X4, X1, X2 )$  donde las variables son equivalentes.

#### BLOQUE 19

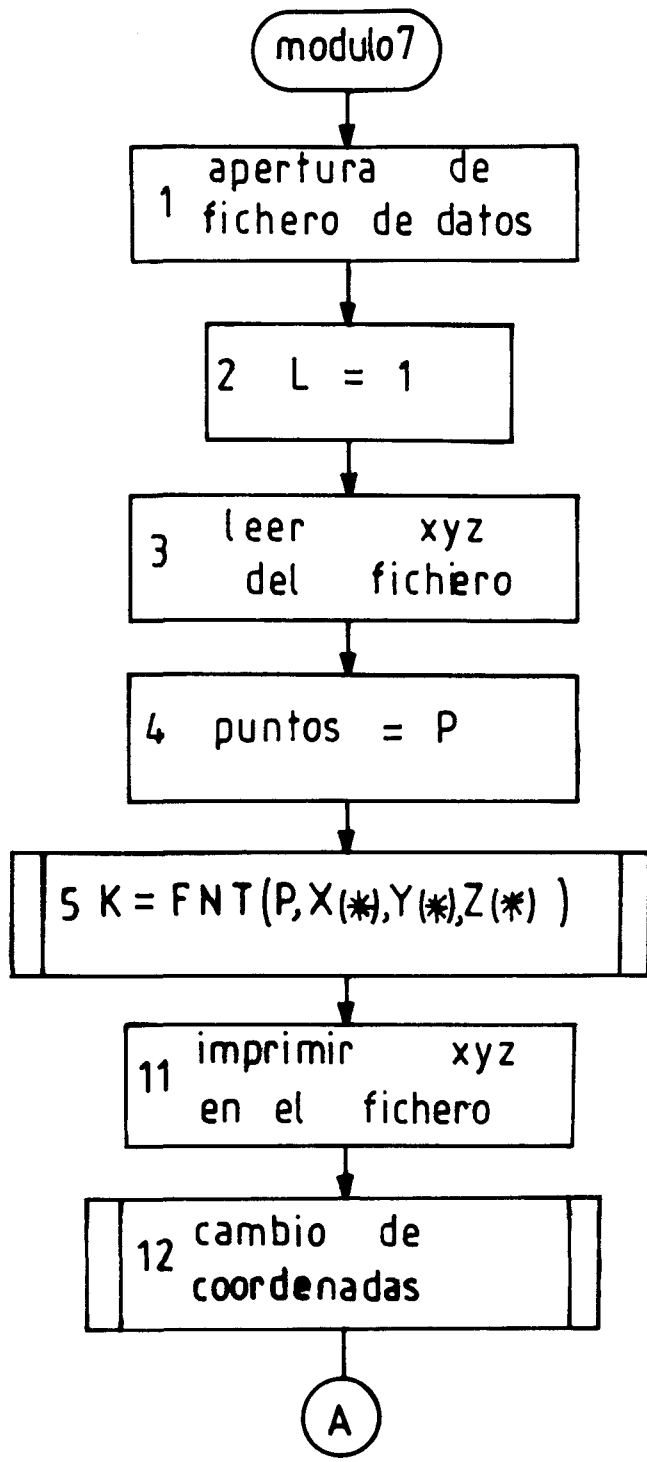
L=0 indica a los bloques posteriores que todos los datos que se lean sean imprimidos en el fichero de salida que es el FICHERO 1.

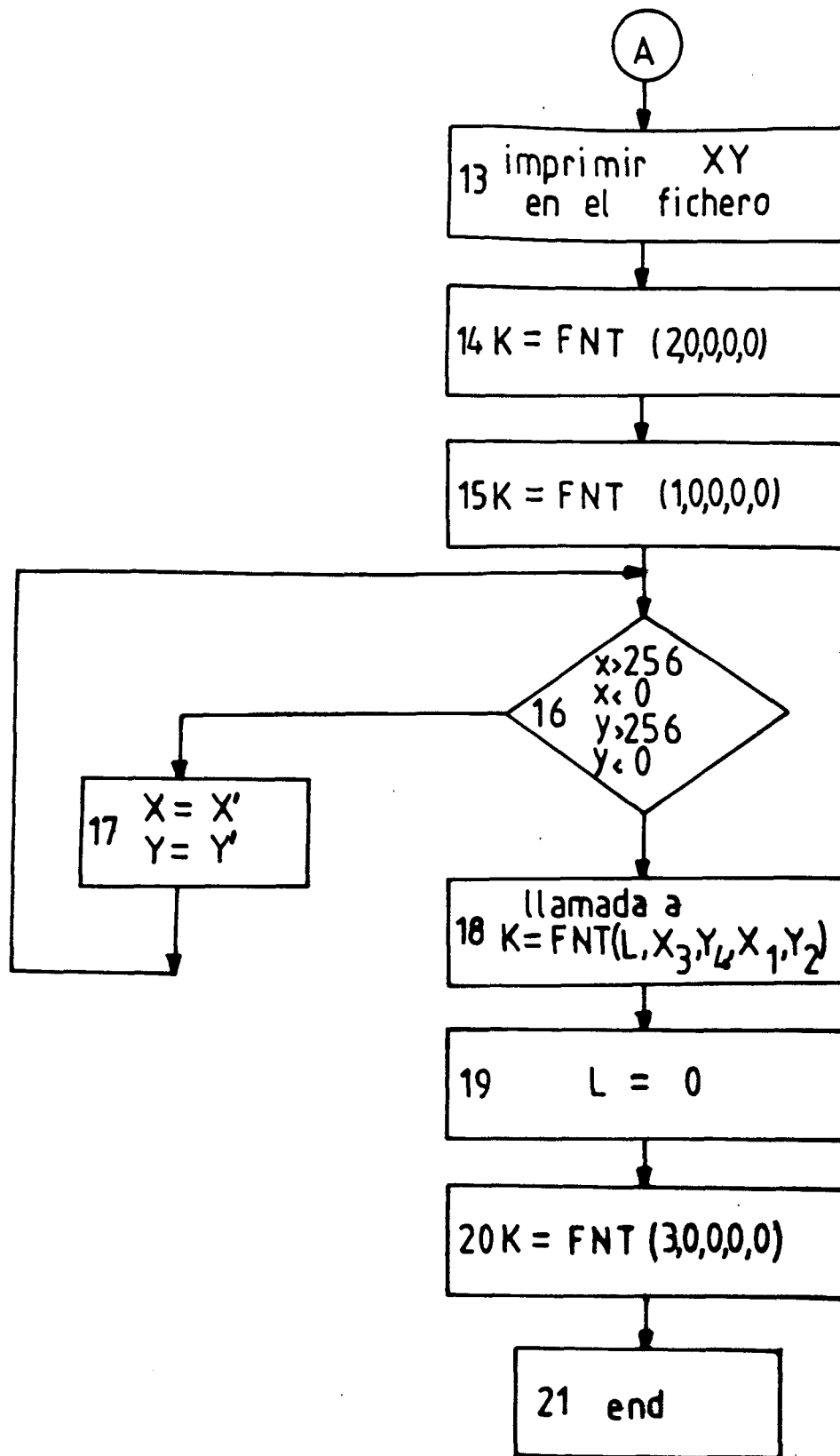
#### BLOQUE 20

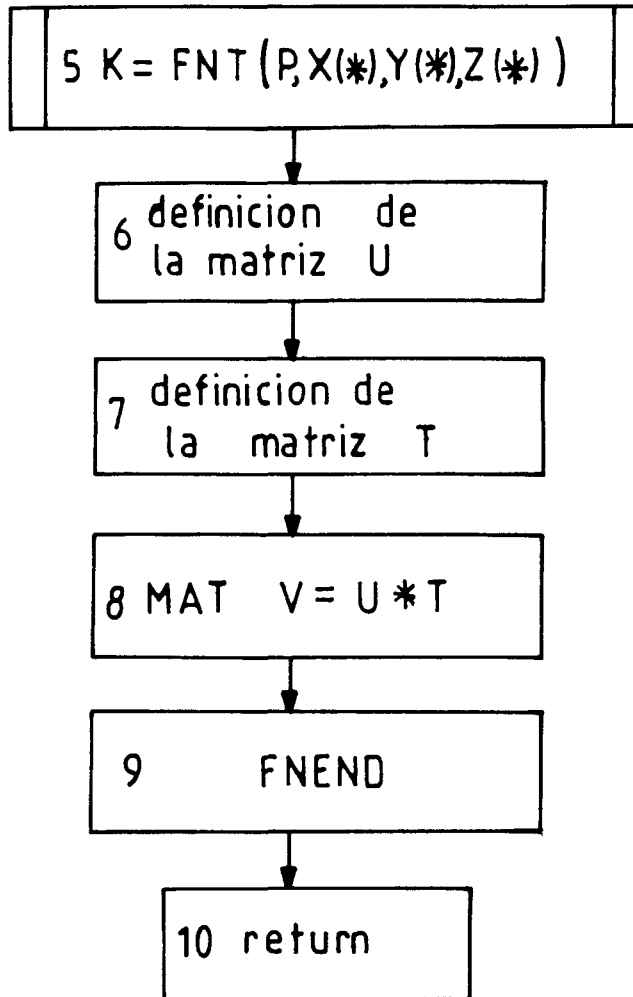
Una llamada a la función  $K = FNT ( L9, X8, Y8, X9, Y9 )$  mediante  $K = FNT ( 3, 0, 0, 0, 0 )$  donde se vuelca las variables en un buffer hasta un final de fichero.

#### BLOQUE 21

Termina el programa.







## 2.10 MODULO 8. PROYECCION ISOMETRICA DEL DIBUJO

Este módulo permite la proyección isométrica del dibujo cualquiera que sea la forma.

### EXPLICACION DEL ORGANIGRAMA

#### BLOQUE 1.

FICHERO 2: OUTPUT1 Localiza la posición de grabación correspondiente a la introducción de datos en tres dimensiones de la figura .

FICHERO 3: INPUT7 localiza la posición de grabación correspondiente a la transformación del fichero OUTPUT1 en dos dimensiones.

FICHERO 4: ISO Localiza la posición de grabación correspondiente a la transformación del fichero INPUT2 en dos dimensiones.

FICHERO 1: GRAPH19 Localiza la posición de grabación correspondiente a la transformación del fichero INPUT1 mediante la función  $K = FNT ( L, X3, X4, X1, X2 )$  .

Inicialización de las variables X1, X2, X3, X4.

VARIABLES:

L: Indica la posición de el lápiz del plotter.



L=0    Lápiz abajo

L=1    Lápiz arriba

X1,X2    Corresponde al valor inicial de X,Y.

X3,X4:    Corresponde al valor final de X',Y'.

## BLOQUE 2

L=1 Indicará a los bloques posteriores que estará en espera del L=0 para la grabación de los datos en el FICHERO 1.

## BLOQUE 3

Se leen los valores X, Y , Z del FICHERO 2.

## BLOQUE 4

En este bloque nos dice la cantidad de puntos que compone la figura . Si la cantidad de puntos es menor que la que corresponde nos dara error.

## BLOQUE 5

Una llamada a la función  $K = FNA ( P, X(*), Y(*), Z(*) )$

## BLOQUE 6

Definición de la matriz U. Cuyo valor sera los puntos X, Y, Z de la figura.

$$U = \begin{vmatrix} X, Y, Z, 1 \end{vmatrix}$$

## BLOQUE 7

Definición de la matriz T. Cuyo valor será los parametros correspondiente a la proyección isométrica.

$$\begin{vmatrix} 0.92580 & 0.13361 & -0.353553 & 0 \\ 0 & 0.935414 & 0.353553 & 0 \\ 0.377964 & -0.32751 & 0.866025 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

## BLOQUE 8

Definición de la matriz  $V$  que tendrá de valor el producto vectorial de  $U * T$ .

$$V = U * T$$

## BLOQUE 9

Finalización de la función  $K = FNA ( P, X(*), Y(*), Z(*) )$

## BLOQUE 10

Se devuelve el control al programa donde se hizo la llamada.

## BLOQUE 11

Se imprime los puntos correspondientes de la figura en el FICHERO 3

## BLOQUE 12

Se selecciona un cambio de coordenadas de tres a dos dimensiones.

#### VARIABLES:

X1: Valor de X en dos dimensiones.

X2: Valor de Y en dos dimensiones.

#### BLOQUE 13

Se imprimen los nuevos puntos X, Y de la figura en el FICHERO 4.

#### BLOQUE 14

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  para inicializar las variables haciendose de la siguiente forma  $K = \text{FNT} ( 2, 0, 0, 0, 0 )$ .

#### BLOQUE 15

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante la cual coloca el lápiz del plotter en el origen en espera de datos.

#### BLOQUE 16

Explora los límites que no puede exceder la X ni la Y. En caso afirmativo pasa la bloque 17 . En caso negativo al bloque 18.

El valor de X no puede exceder más de 256 y menos de

0 y los mismo ocurre para la Y.

No pueden exceder de este valor porque son los límites máximos del plotter.

#### BLOQUE 17

Las variable X tomara el valor X' y la variable Y tomara el valor Y' . Y se vuelve al bloque 16.

#### BLOQUE 18

Una llamada a la función  $K = \text{FNT} ( L9, X3, Y8, X9, Y9 )$  mediante la función  $K = \text{FNT} ( L, X3, X4, X1, X2 )$  donde las variables son equivalentes.

#### BLOQUE 19

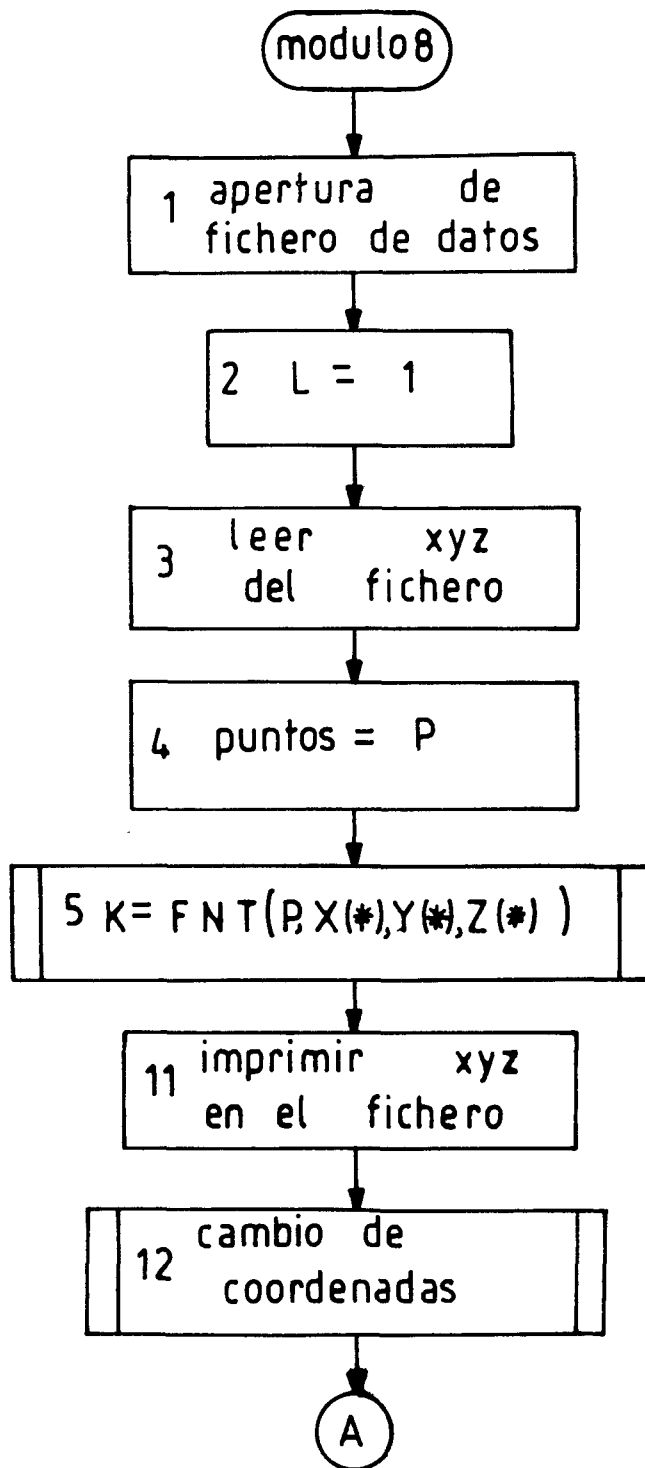
$L=0$  indica a los bloques posteriores que todos los datos que se lean serán imprimidos en el fichero de salida que es el FICHERO 1.

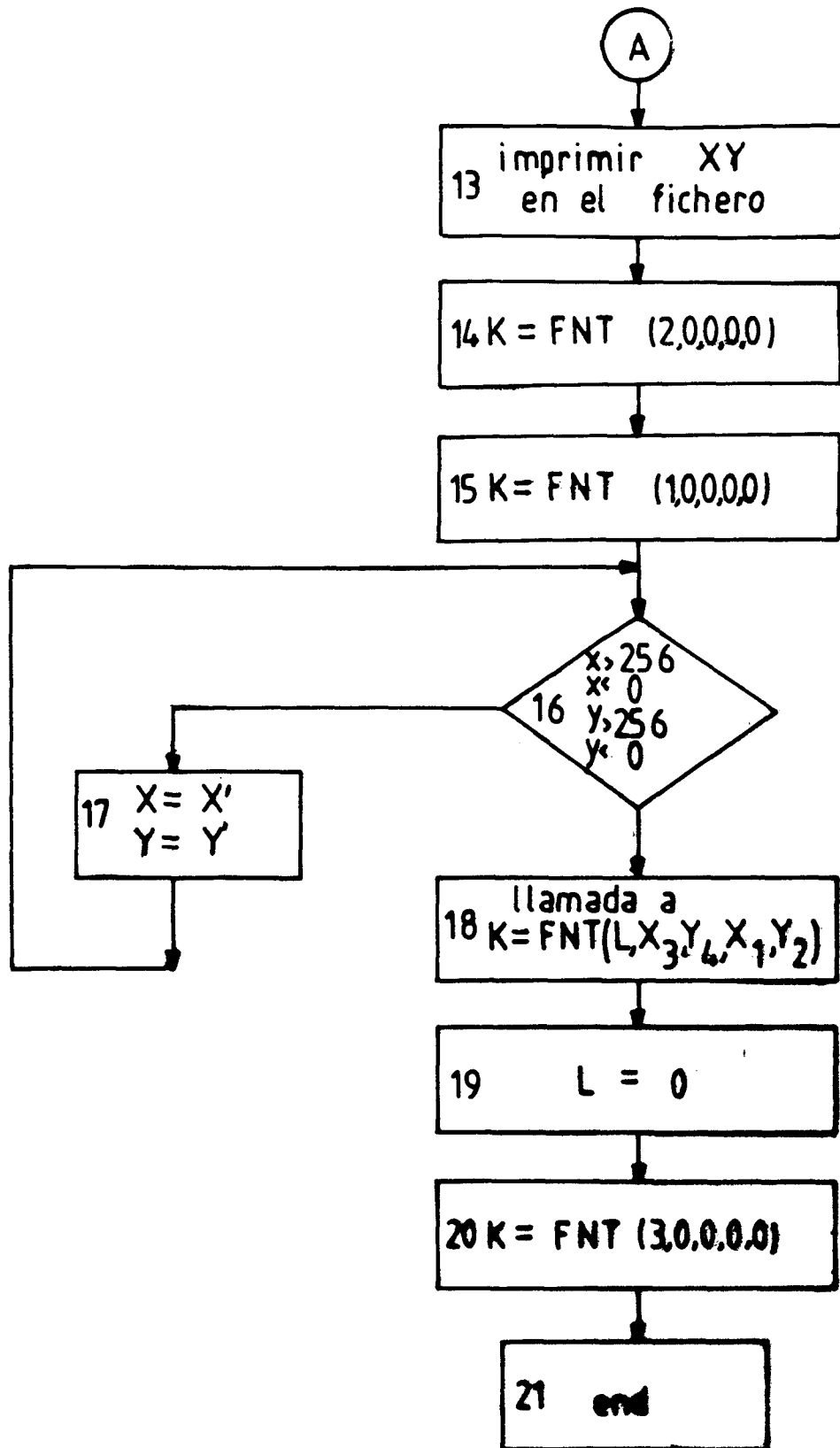
#### BLOQUE 20

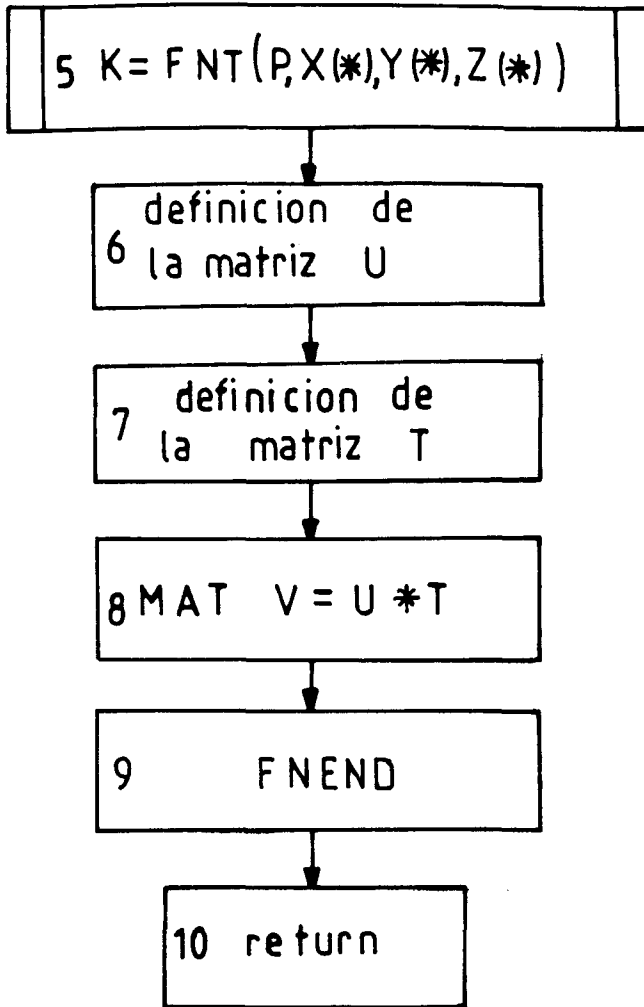
Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante  $K = \text{FNT} ( 3, 0, 0, 0, 0 )$  donde se vuelca las variables en un buffer hasta un final de fichero.

#### BLOQUE 21

Termina el programa.









## 2.11 MODULO 9. PROYECCION CONICA DEL DIBUJO

Este módulo permite la proyección cónica del dibujo cualquiera que sea la forma.

### EXPLICACION DEL ORGANIGRAMA

#### BLOQUE 1.

FICHERO 2: OUTPUT1 Localiza la posición de grabación correspondiente a la introducción de datos en tres dimensiones de la figura .

FICHERO 3: INPUT22 localiza la posición de grabación correspondiente a la transformación del fichero OUTPUT1 en dos dimensiones.

FICHERO 4: TRAS1 Localiza la posición de grabación correspondiente a la transformación del fichero INPUT22 en dos dimensiones.

FICHERO 1: GRAPH141 Localiza la posición de grabación correspondiente a la transformación del fichero INPUT1 mediante la función  $K = FNT ( L, X3, X4, X1, X2 )$  .

Inicialización de las variables X1, X2, X3, X4.

VARIABLES:

L: Indica la posición de el lápiz del plotter.

L=0 Lápiz abajo

L=1 Lápiz arriba

X1,X2: Corresponde al valor inicial de X,Y.

X3,X4: Corresponde al valor final de X',Y'.

## BLOQUE 2

Introducción de los factores de traslación de X , Y ,  
Z que son L , M ,N.

## BLOQUE 3

L=1 Indicará a los bloques posteriores que estará en  
espera del L=0 para la grabación de los datos en el  
FICHERO 1.

## BLOQUE 4

Se leen los valores X, Y , Z del FICHERO 2.

## BLOQUE 5

En este bloque nos dice la cantidad de puntos que  
compone la figura . Si la cantidad de puntos es menor  
que la que corresponde nos dara error.

## BLOQUE 6

Una llamada a la función  $K = FNA ( P, X(*), Y(*), Z(*), L, M, N )$

## BLOQUE 7

Definición de la matriz U. Cuyo valor sera los puntos X, Y, Z de la figura.

$$U = \begin{vmatrix} x & y & z & 1 \end{vmatrix}$$

## BLOQUE 8

Definición de la matriz T. Cuyo valor será los parametros correspondiente a la traslación.

$$T = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ L & m & n & 1 \end{vmatrix}$$

## BLOQUE 9

Definición de la matriz  $V$  que tendrá de valor el producto vectorial de  $U * T$ .

$$V = U * T$$

## BLOQUE 10

Finalización de la función  $K = FNA ( P, X(*), Y(*), Z(*), L, M, N )$

## BLOQUE 11

Se devuelve el control al programa donde se hizo la llamada.

## BLOQUE 12

Se imprime los puntos correspondientes de la figura en el FICHERO 3

### BLOQUE 13

Se selecciona un cambio de coordenadas de tres a dos dimensiones.

#### VARIABLES:

X1: Valor de X en dos dimensiones.

X2: Valor de Y en dos dimensiones.

### BLOQUE 14

Se imprimen los nuevos puntos X, Y de la figura en el FICHERO 4.

### BLOQUE 15

Una llamada a la función  $K = FNT ( L9, X8, Y8, X9, Y9 )$  para inicializar las variables haciendose de la siguiente forma  $K = FNT ( 2, 0, 0, 0, 0 )$ .

### BLOQUE 16

Una llamada a la función  $K = FNT ( L9, X8, Y8, X9, Y9 )$  mediante la cual coloca el lápiz del plotter en el origen en espera de datos.

### BLOQUE 17

Explora los límites que no puede exceder la X ni la Y. En caso afirmativo pasa la bloque 18 . En caso

negativo al bloque 19.

El valor de X no puede exceder más de 256 y menos de 0 y lo mismo ocurre para la Y.

No pueden exceder de este valor porque son los límites máximos del plotter.

#### BLOQUE 18

Las variables X tomará el valor X' y la variable Y tomará el valor Y'. Y se vuelve al bloque 17.

#### BLOQUE 19

Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante la función  $K = \text{FNT} ( L, X3, X4, X1, X2 )$  donde las variables son equivalentes.

#### BLOQUE 20

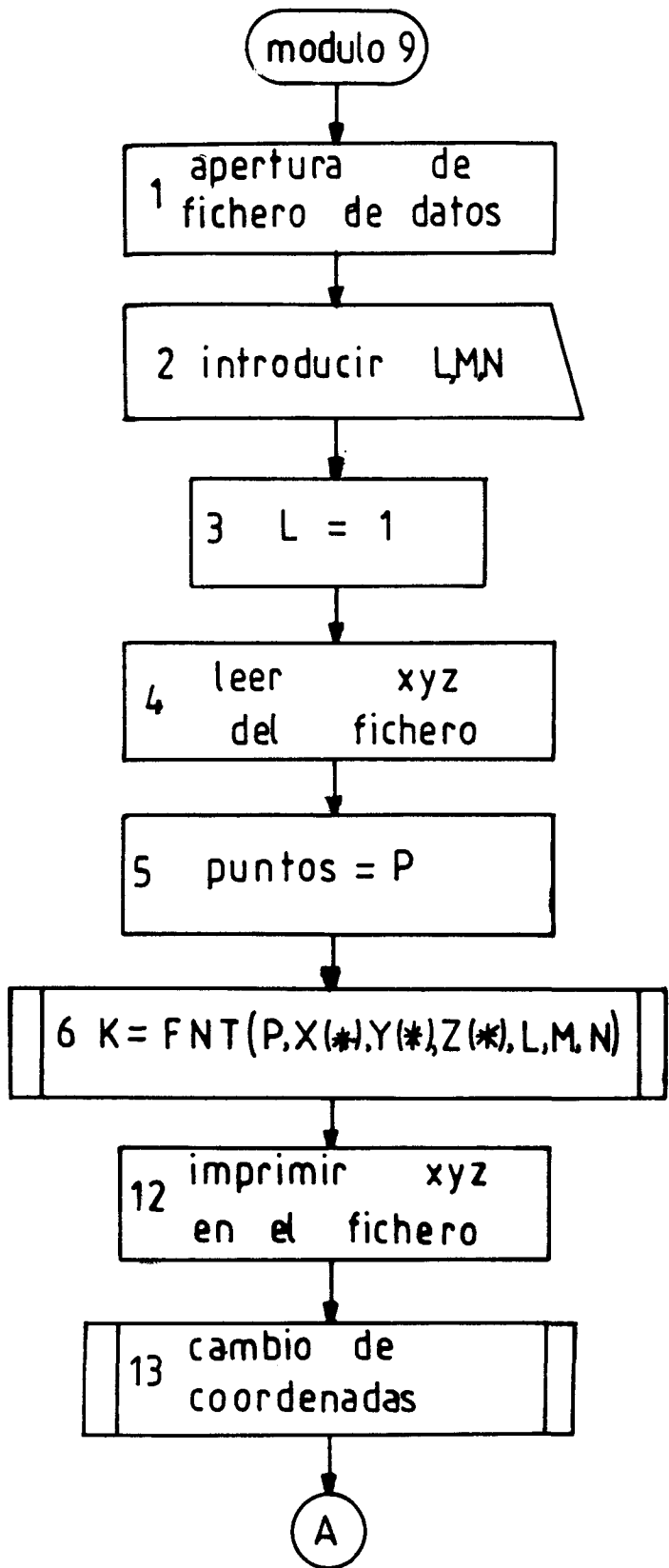
$L=0$  indica a los bloques posteriores que todos los datos que se lean serán imprimidos en el fichero de salida que es el FICHERO 1.

#### BLOQUE 21

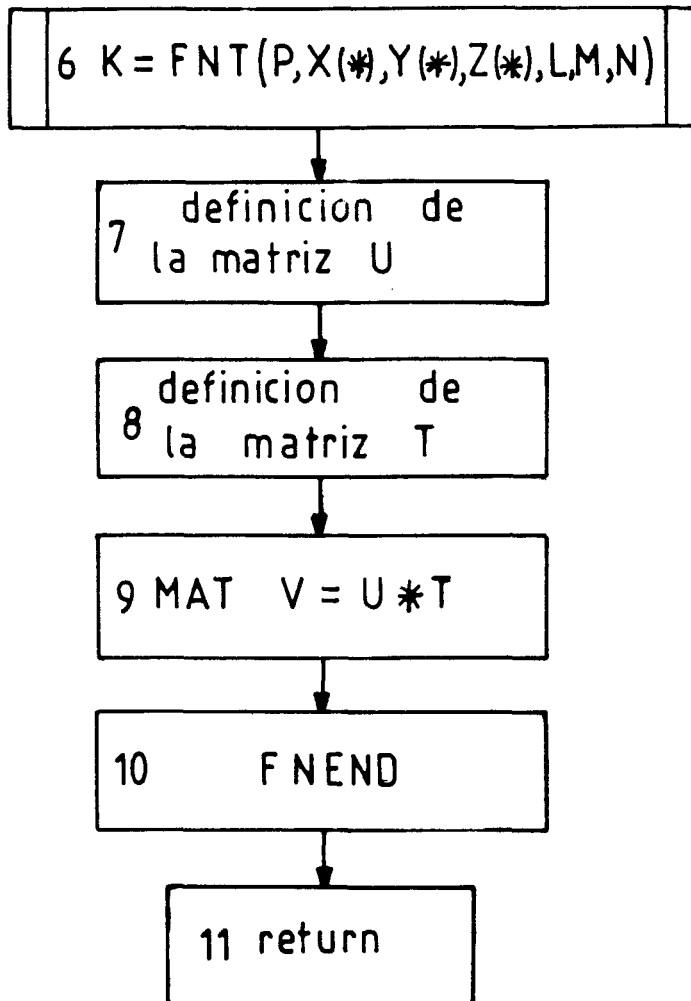
Una llamada a la función  $K = \text{FNT} ( L9, X8, Y8, X9, Y9 )$  mediante  $K = \text{FNT} ( 3, 0, 0, 0, 0 )$  donde se vuelca las variables en un buffer hasta un final de fichero.

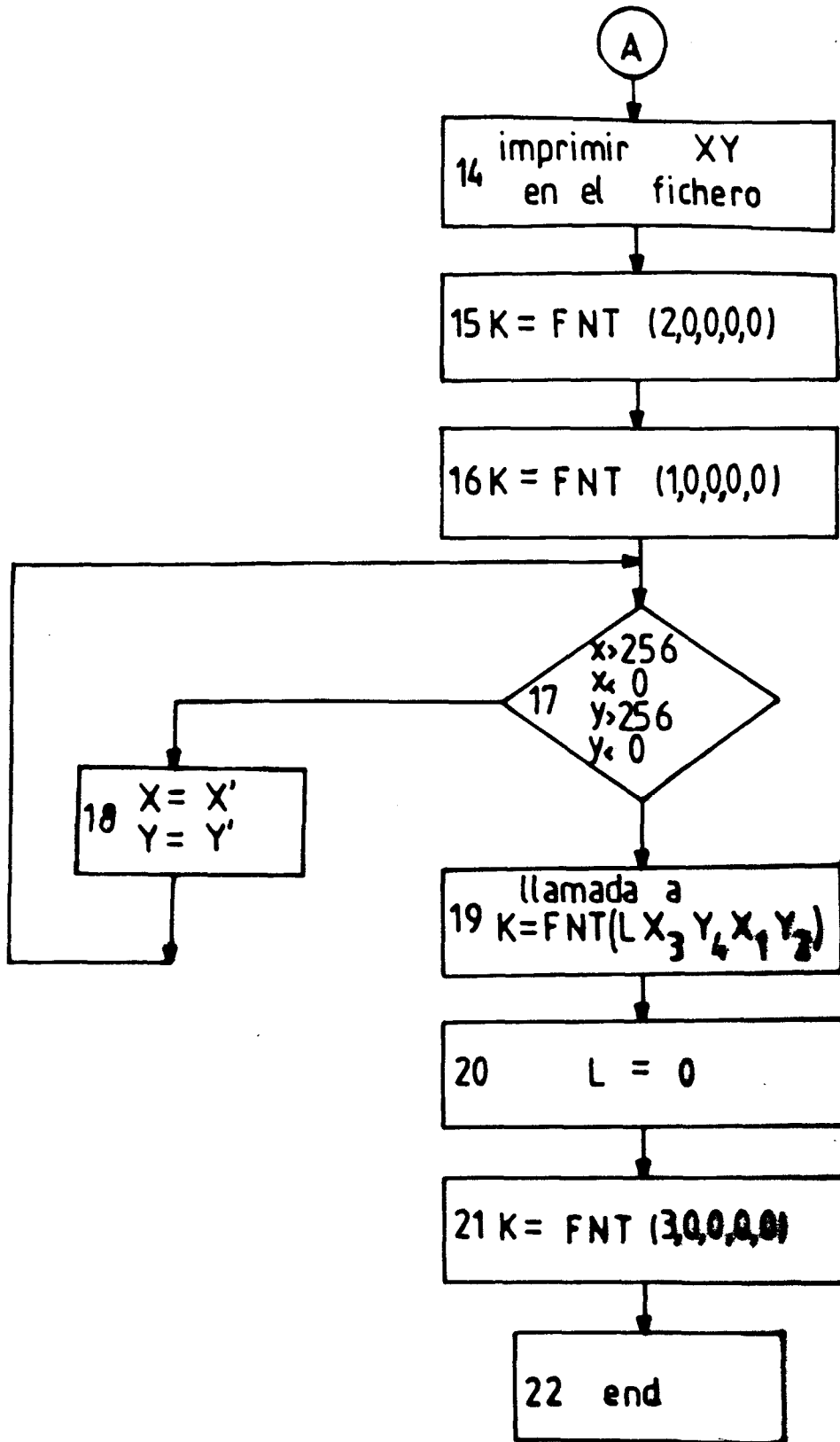
#### BLOQUE 22

Termina el programa.









### 3. PROGRAMA TRIDIMENSIONAL

#### 3.1 ANALISIS DEL PROGRAMA TRIDIMENSIONAL

En esta sección se estudia el funcionamiento del programa TRIDIMENSIONAL, cuya estructuración en módulos independientes permite un cómodo análisis de cada una de sus funciones.

#### 3.2 CARACTERISTICAS GENERALES

El programa TRIDIMENSIONAL está codificado en el lenguaje PLM/80. La característica de este programa es la interpretación de los ficheros de las correspondientes figuras realizados por el programa de control EXA y sus correspondientes módulos.

Se modifica la configuración I/O para que sea uno de los conectores del Sistema de Desarrollo, concretamente el PUNCH COMMAND (salida para tarjeta perforada) la salida.

#### 3.3 PROGRAMA MENU

Este módulo tiene la misión de seleccionar al comando deseado y transferir el control al módulo correspondiente, lo cual se realiza por concatenación devolviendo al final de cada módulo el control al programa menú.

### 3.4 PROGRAMA TRIDIMENSIONAL

#### EXPLICACION DEL ORGANIGRAMA

##### BLOQUE 1

Selecciona el comando correspondiente al módulo que desea utilizarse

##### BLOQUE 2

Selecciona el comando correspondiente al módulo que desea utilizarse.

##### BLOQUE 3

Transfiere el control al módulo seleccionado.

##### BLOQUE 4

Explora si el comando seleccionado es de salida . En caso afirmativo termina el programa . En caso negativo pasa al bloque 1.

##### MODULO 2

#### EXPLICACION DEL ORGANIGRAMA

### BLOQUE 1

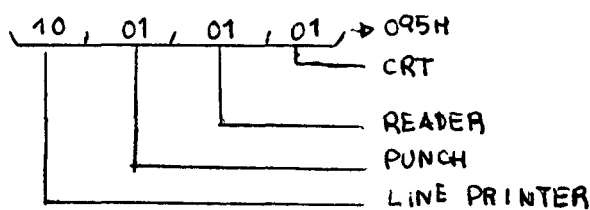
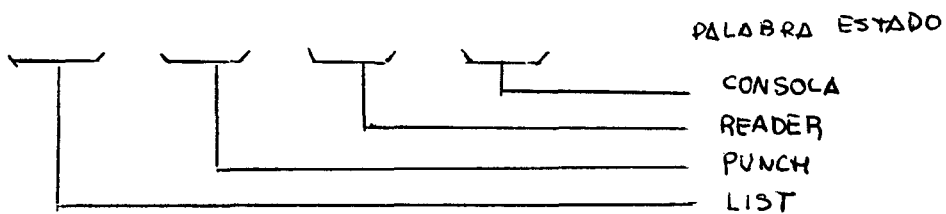
Se selecciona el nombre del fichero que se va a utilizar mediante una llamada a la subrutina DATA\$IN y nos permite utilizar por programa la tecla Rubout para poder borrar .

### BLOQUE 2

Se selecciona que uno de los conectores del Sistema de Desarrollo concretamente el PUNCH COMMAND como salida . Se denomina IOSET.

VALOR	CONSOLA	READER	PUNCH	LIST
00	TTY	TTY	TTY	TTY
01	CRT	HS READER	HS PUNCH	CRT
10	BATCH	U.D	U.D	LINE PRINTER
11	U.D	U.D	U.D	U.D

### BIT S



### BLOQUE 3

Apertura de fichero.

### BLOQUE 4

Lectura de los primeros 128 bytes.

### BLOQUE 5

Lápiz inicial ARRIBA para evitar errores.

### BLOQUE 6

Llamada a la subrutina.

### BLOQUE 7

Se explora el valor inicial del lápiz que sea igual al valor actual de lápiz . En caso afirmativo pasa al bloque 8 . En caso negativo pasa al bloque 9.

### BLOQUE 8

Transmisión del valor actual del lápiz.

### BLOQUE 9

Transmisión del valor actual de lápiz . Pero la diferencia esta en que si el nuevo valor es distinto del anterior se introduce un retraso mayor ( CAAL TIME ) para evitar movimientos bruscos del lápiz.

#### BLOQUE 10

Transmisión de valor de X y el de Y.

#### BLOQUE 11

Mediante el procedimiento PO ( Punch output routine )  
Toma un byte y lo transmite al dispositivo PUNCH.

#### BLOQUE 12

Llamada a la Subrutina COGEB.

#### BLOQUE 13

Llamada a la Subrutina ACCFIL.

#### BLOQUE 14

Explora si los bytes son mayores que 27 . En caso  
negativo retorna donde se hizo la llamada. Y pasa al  
bloque 16 . En caso positivo pasa al bloque 15.

#### BLOQUE 15

Lectura de los 128 bytes siguientes y retorna donde  
se hizo la llamada y pasa al bloque 16.

#### BLOQUE 16

Llamada a la Subrutina HEXADECIMAL.

#### BLOQUE 17

Explora si el byte es mayor o igual que 0 o mayor o igual que 9 . En caso afirmativo pasa al bloque 18 . En caso negativo pasa al bloque 19.

#### BLOQUE 18

Realización de una And entre el byte y 0FH para tener solo la parte menos significativa y retorna donde se hizo la llamada y pasa al bloque 28.

#### BLOQUE 19

Explora si el byte es mayor o igual que A , o mayor o igual que F. En caso afirmativo pasa al bloque 20 . En caso negativo pasa al bloque 21.

#### BLOQUE 20

Realización de una AND entre el byte y 0FH para tener solo la parte menos significativa y retorna donde se hizo la llamada y pasa al bloque 28.

#### BLOQUE 21

Explora si el byte es igual a X . En caso afirmativo pasa al bloque 22 . En caso negativo pasa al bloque 23.

#### BLOQUE 22

Asigna a la variable FLAG el valor 2 y retorna donde



se hizo la llamada y pasa al bloque 28.

#### BLOQUE 23

Explora si el byte es igual a L . En caso afirmativo pasa al bloque 24. En caso negativo pasa al bloque 25.

#### BLOQUE 25

Explora si el byte es igual a U . En caso afirmativo pasa al bloque 26. En caso negativo pasa al bloque 27.

#### BLOQUE 26

Asigna a la variable FLAG el valor 4 y retorna donde se hizo la llamada y pasa al bloque 28.

#### BLOQUE 27

Explora si el byte es igual al CR . Tanto si el caso es positivo como si es negativo retorna donde se hizo la llamada y pasa al bloque 28.

#### BLOQUE 28

Explora si la variable es igual a 2 en caso afirmativo termina el programa . En caso negativo pasa al bloque 29.

#### BLOQUE 29

Explora si la variable FLAG es igual a 3 en caso

afirmativo pasa al bloque 30 . En caso negativo pasa al bloque 31.

#### BLOQUE 30

Indica la posición del lápiz que es en este caso LAPIZ ARRIBA.

#### BLOQUE 31

Explora si la variable es igual a 4 en caso afirmativo pasa al bloque 32. en caso negativo pasa al bloque 12.

#### BLOQUE 32

Indica la posición del lápiz que es en este caso LAPIZ ABAJO.

#### BLOQUE 33

Analiza el valor X1 haciendo una llamada a la subrutina COGEB que es el bloque 12.

#### BLOQUE 34

Analiza el valor X2 haciendo una llamada a la subrutina COGEB que es bloque 12.

#### BLOQUE 35

El verdadero valor del desplazamiento en el eje X es

el valor X2 más el desplazamiento de X1 cuatro posiciones a la izquierda.

#### BLOQUE 36

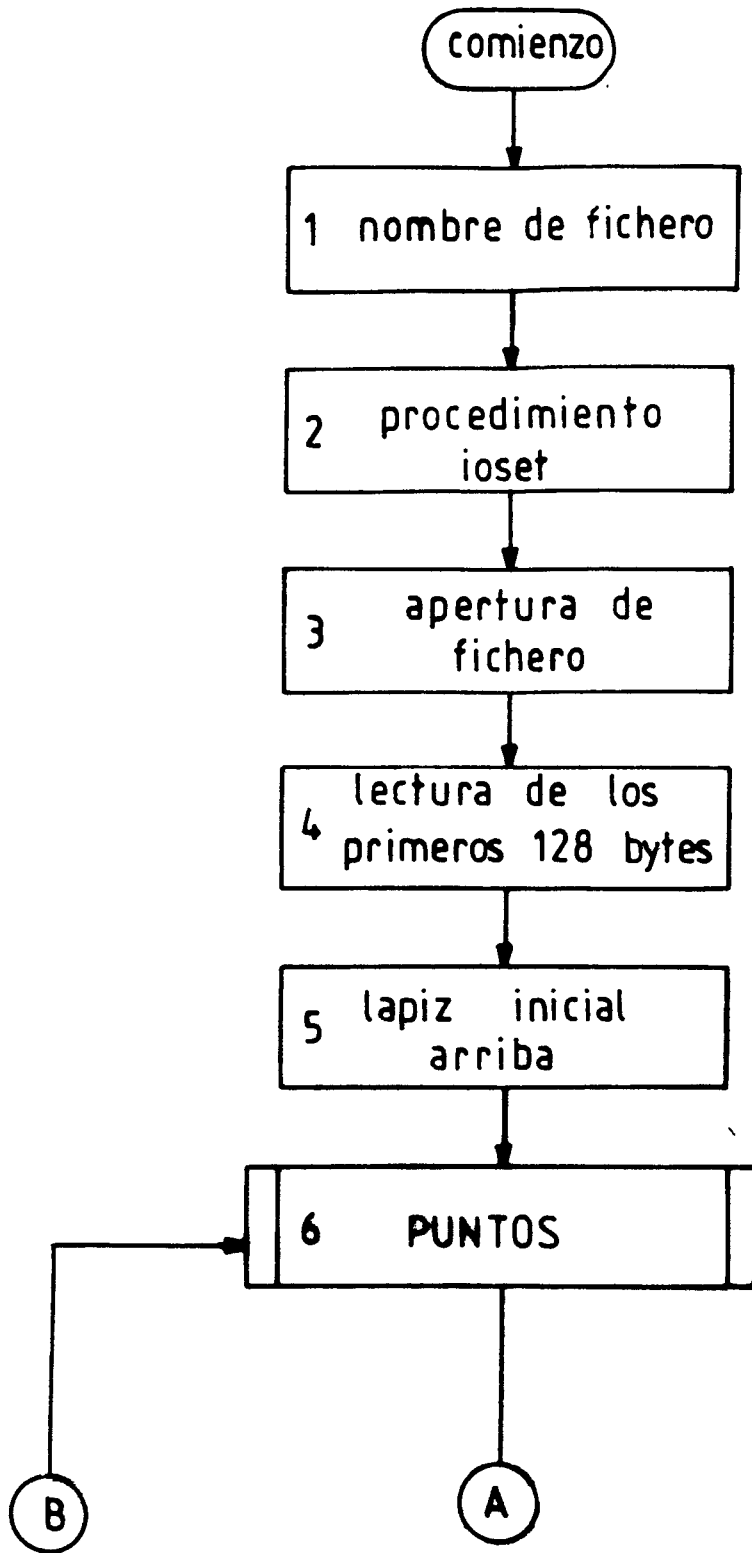
Analiza el valor Y1 haciendo una llamada a la subrutina COGEB que es el bloque 12.

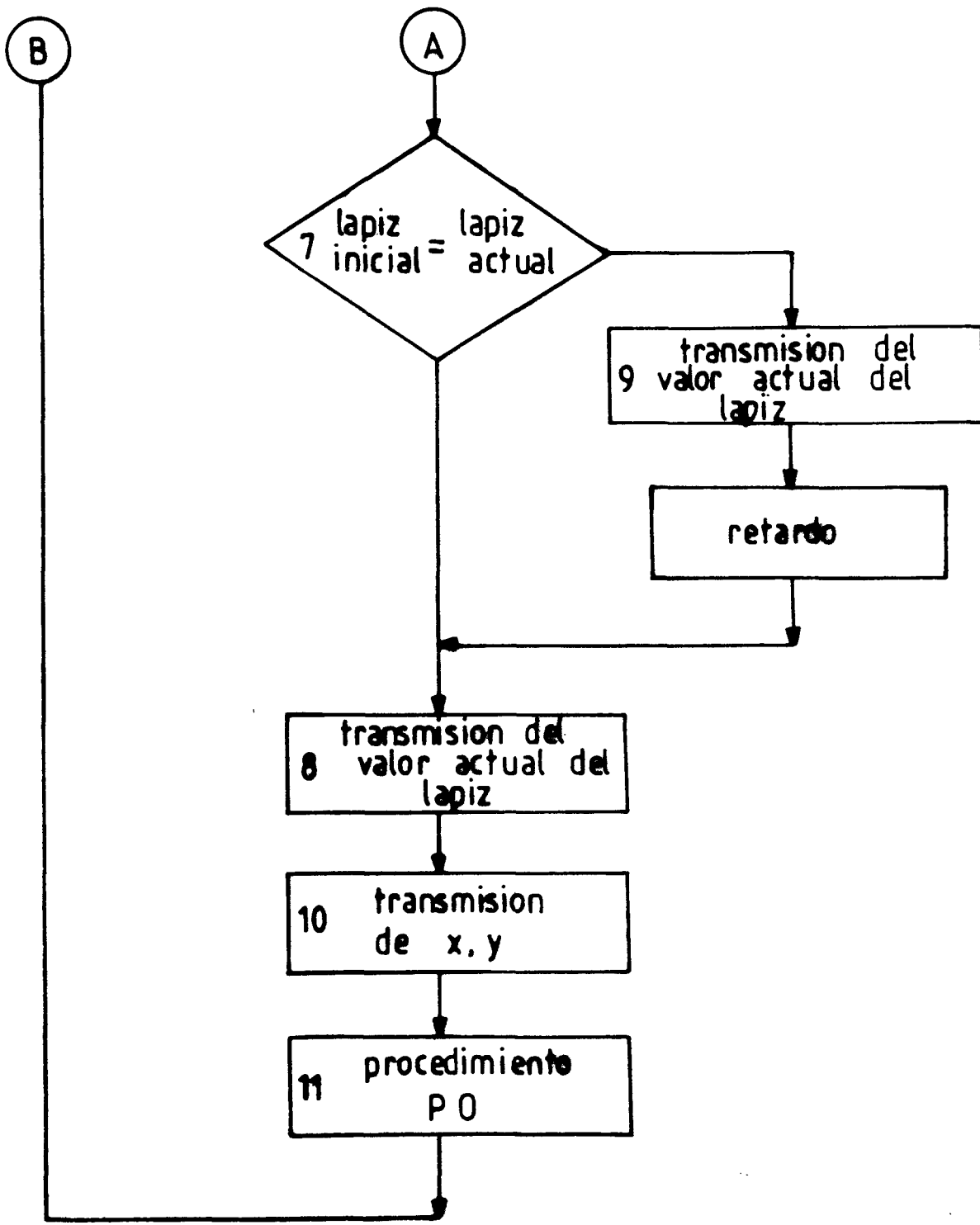
#### BLOQUE 37

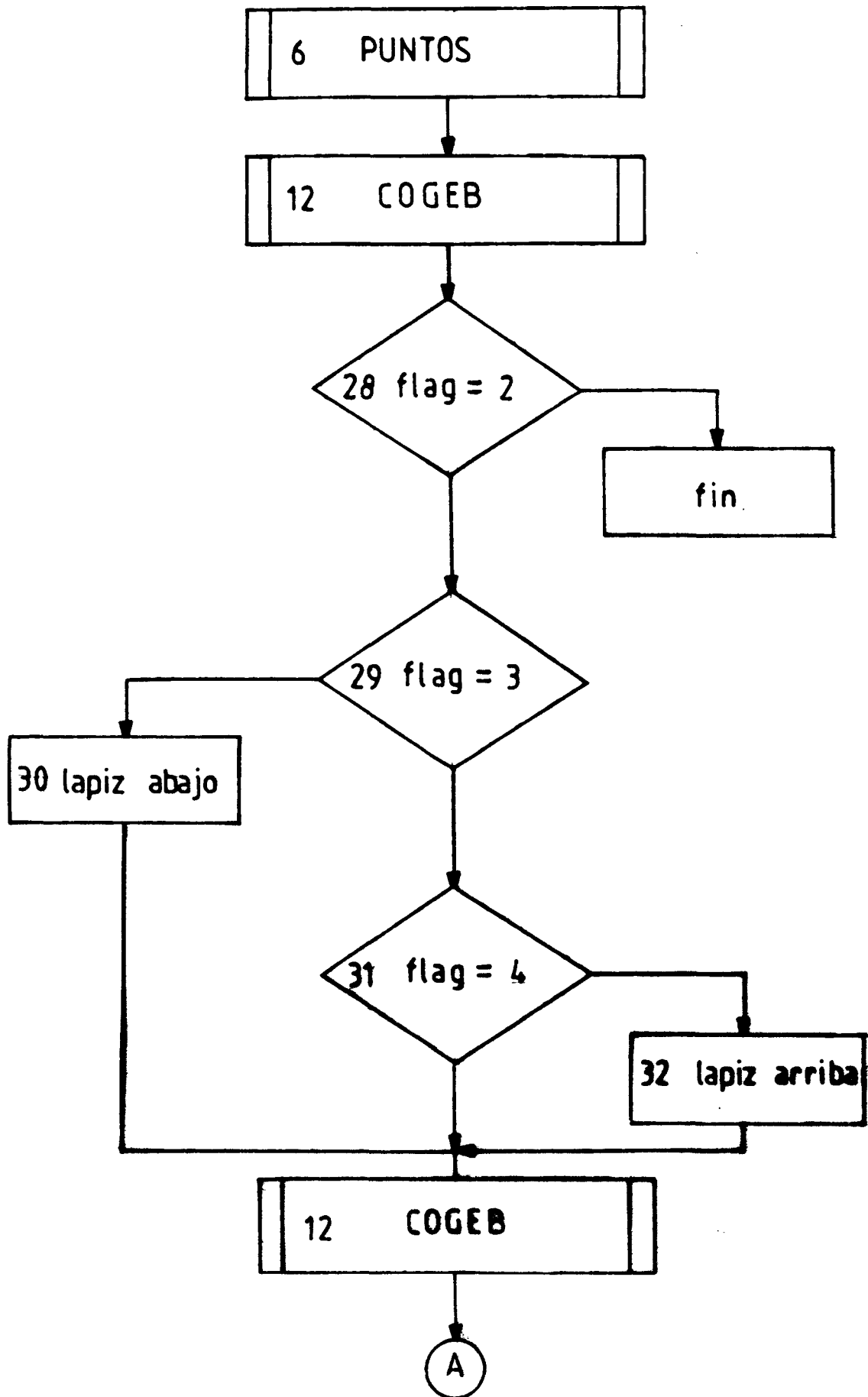
Analiza el valor Y2 haciendo una llamada a la subrutina COGEB que es igual al bloque 12.

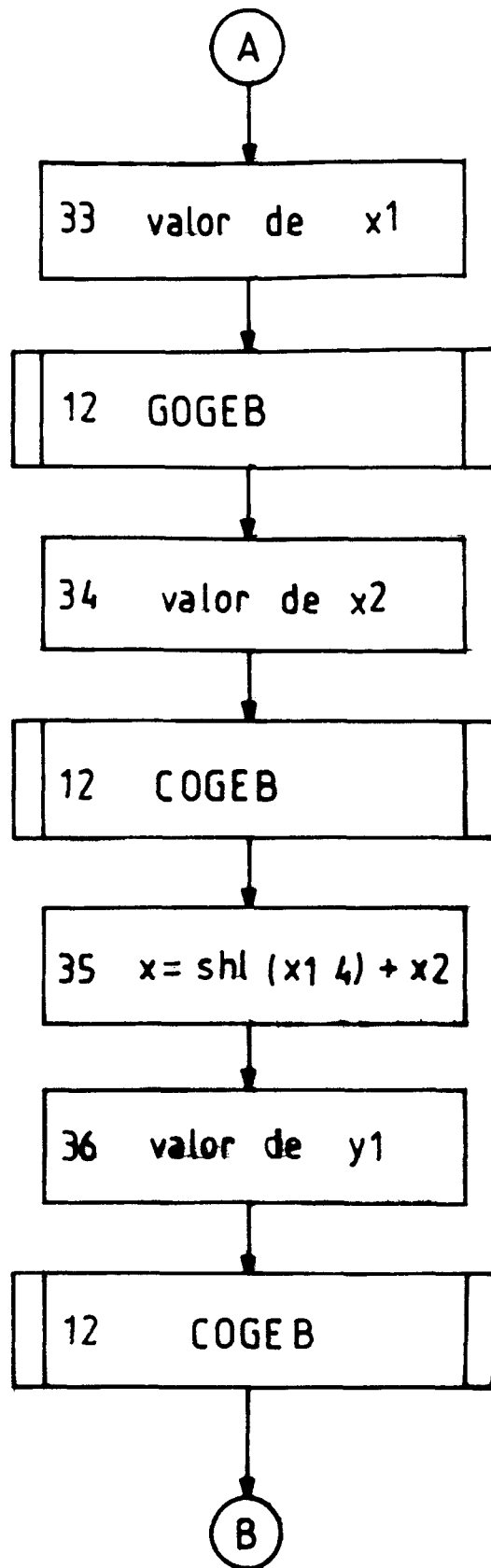
#### BLOQUE 38

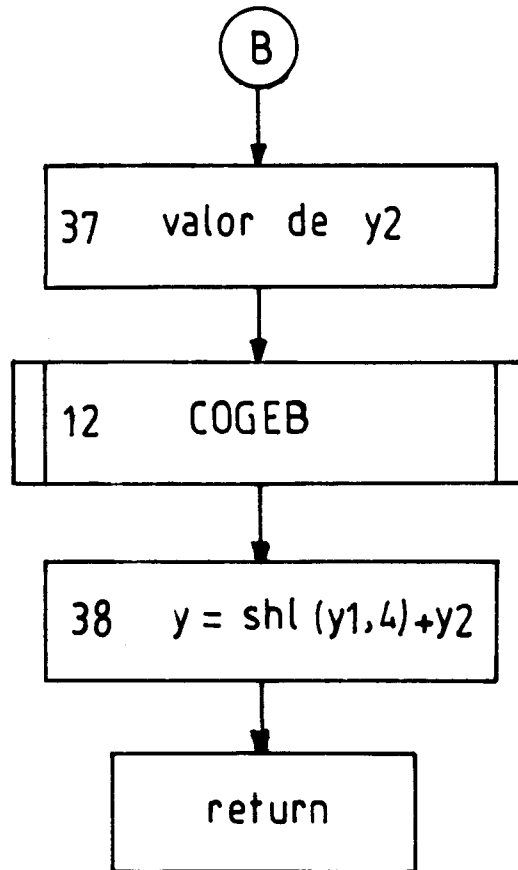
El verdadero valor del desplazamiento en el eje Y es el valor Y2 más el valor de Y1 cuatro posiciones a la izquierda y pasa al bloque 7.



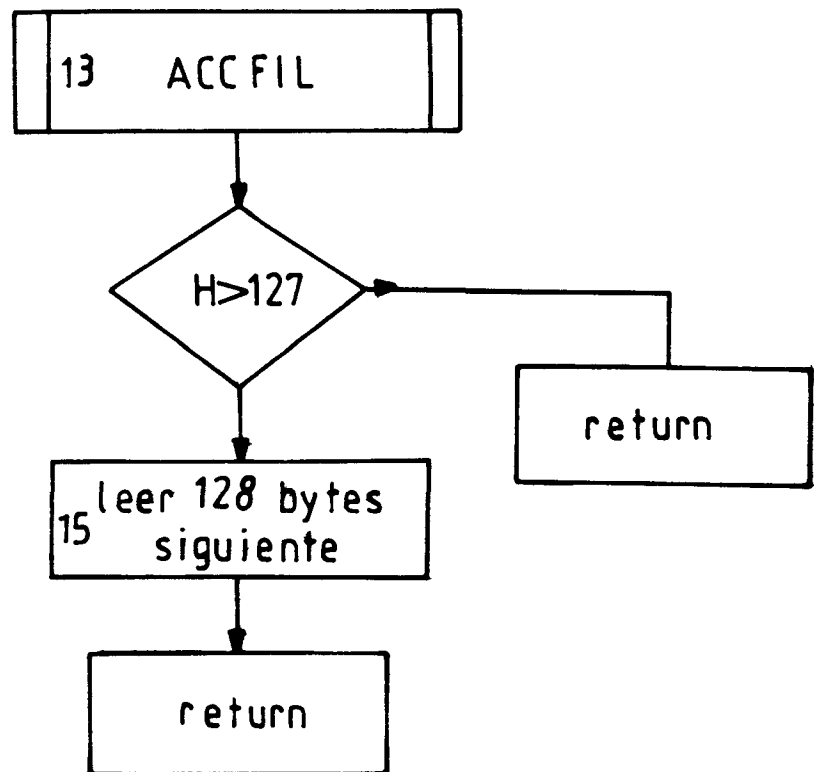
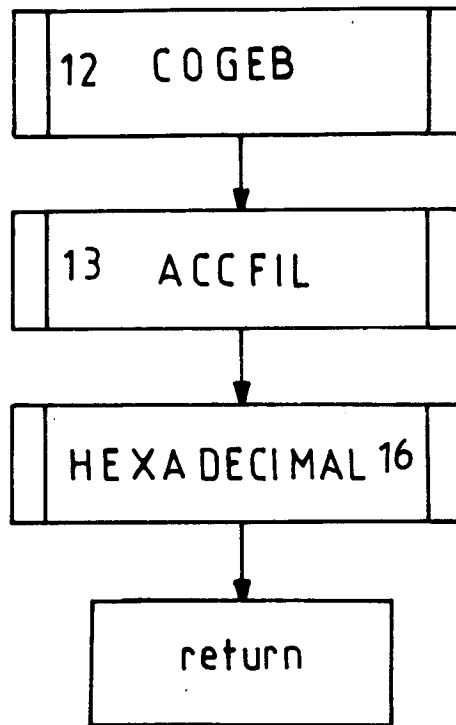


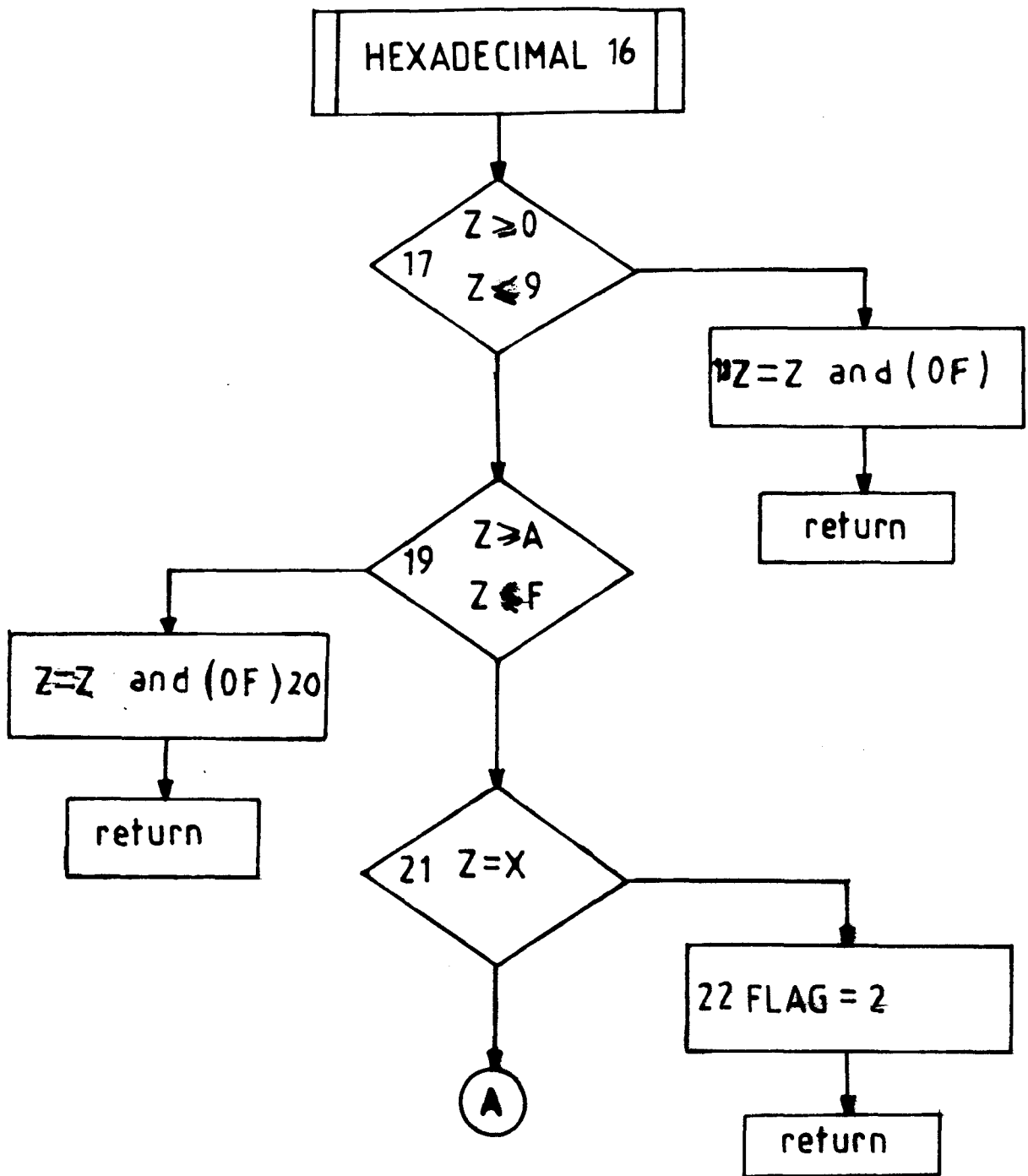


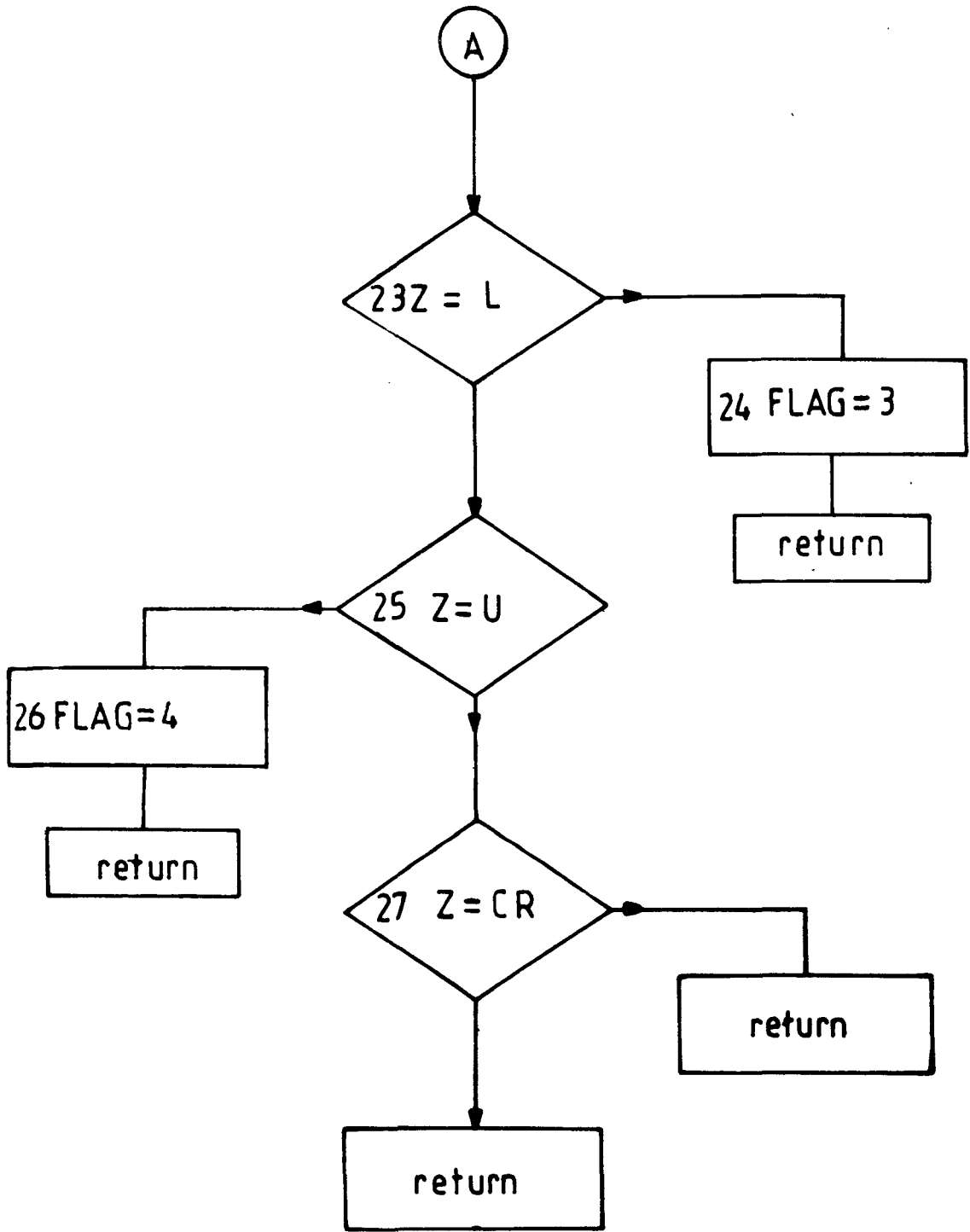












## LISTADO DEL PROGRAMA EXA. Y SUS MODULOS

EXA

```
10 DEF FNT(L9,X8,Y8,X9,Y9)
20 IF L9=3 THEN DO
30 A9=1
40 O9#="O9#"+"X"
50 GOSUB 1020
60 PRINT #1;O9#
70 GOTO 1100
80 DOEND
90 IF L9=2 THEN DO
100 S9=0
110 L8=0
120 O9#=""
130 H8=J8=0
140 GOTO 1100
150 DOEND
160 IF X8<>H8 OR Y8<>J8 THEN DO
170 Z9=FNT(1,H8,J8,X8,Y8)
180 DOEND
190 IF L8=0 AND L9=1 THEN DO
200 A9=1
210 O9#="O9#"+"U"
220 GOSUB 1020
230 A9=2
240 T8=X8
250 Z8=Y8
260 GOSUB 920
270 DOEND
280 IF L8=1 AND L9=0 THEN DO
290 A9=1
300 O9#="O9#"+"L"
310 GOSUB 1020
320 A9=2
330 T8=X8
340 Z8=Y8
350 GOSUB 920
360 DOEND
370 L8=L9
380 IF L9=0 THEN DO
390 IF SQR((X9-X8)**2+(Y9-Y8)**2)>20 THEN DO
400 Z9=FNT(0,X8,Y8,(X9-X8)/2+X8,(Y9-Y8)/2+Y8)
410 Z9=FNT(0,(X9-X8)/2+X8,(Y9-Y8)/2+Y8,X9,Y9)
420 DOEND
430 ELSE DO
440 A9=1
450 O9#="O9#"+"L"
460 GOSUB 1020
470 A9=2
480 T8=X9
490 Z8=Y9
500 GOSUB 920
510 DOEND
520 DOEND
530 IF L9=1 THEN DO
540 A9=1
550 O9#="O9#"+"U"
560 GOSUB 1020
570 A9=2
580 T8=X9
590 Z8=Y9
600 GOSUB 920
610 DOEND
620 H8=X9
```

```

630 JB=Y9
640 GOTO 1100
650 REM SUBROUTINA CONVERSION DECIMAL HEXADECIMAL
660 F9=0
670 A9$=""
680 V8=INT(V8)
690 V9=V8
700 R9=INT(V9/16)
710 W9=((V9/16)-R9)*16
720 REM GENERACION DE RESTO
730 IF W9>9 THEN DO
740     IF W9=10 THEN B9$="A"
750     IF W9=11 THEN B9$="B"
760     IF W9=12 THEN B9$="C"
770     IF W9=13 THEN B9$="D"
780     IF W9=14 THEN B9$="E"
790     IF W9=15 THEN B9$="F"
800 DOEND
810 ELSE CONVERT W9 TO B9$
820 A9$=B9$+A9$
830 IF R9>16 THEN DO
840     V9=R9
850     GOTO 700
860 DOEND
870 IF F9=1 THEN 910
880 F9=1
890 W9=R9
900 GOTO 720
910 RETURN
920 REM SUBROUTINA DE FORMATEADO DE VARIABLES
930 V8=T8
940 GOSUB 650
950 O9$=O9$+A9$
960 GOSUB 1020
970 V8=Z8
980 GOSUB 650
990 O9$=O9$+A9$
1000 GOSUB 1020
1010 RETURN
1020 REM SUBROUTINA DE EXCRITURA EN FICHERO
1030 S9=S9+A9
1040 IF S9>=80 THEN DO
1050     PRINT #1;O9$
1060     O9$=""
1070     S9=0
1080 DOEND
1090 RETURN
1100 RETURN 1
1110 FNEND

```

EXA1

```
1120 DIM A9#[2],B9#[2],O9#[80]
1130 REM NOMBRE DE LA BASE DE DATOS
1140 FILES GRAPH13,OUTPUT1,INPUT1
1160 K1=FNT(2,0,0,0,0)
1170 K1=FNT(1,0,0,0,0)
1210 X3=X4=0
1220 REM PROGRAMA QUE INTRODUCE LAS COORDENADAS EN EL FICHERO OUTPUT
1230 L=1
1240 FOR I=1 TO 16
1250   PRINT "DEME EL VALOR ";I;" ";
1260   INPUT E1,E2,E3
1270   PRINT #2;E1
1280   PRINT #2;E2
1290   PRINT #2;E3
1300 NEXT I
1310 REM
1320 RESTORE #2
1330 REM
1340 FOR I=1 TO 16
1350   READ #2;E1
1360   READ #2;E2
1370   READ #2;E3
1380   GOSUB 1610
1390   PRINT #3;X1
1400   PRINT #3;X2
1410 NEXT I
1420 REM
1430 X1=0
1440 X2=0
1450 RESTORE #3
1460 REM
1470 L=0
1480 READ #3;X3
1490 READ #3;X4
1500 ON END #3 THEN 1580
1510 READ #3;X1
1520 READ #3;X2
1530 IF X1>256 OR X1<0 OR X2>256 OR X2<0 THEN 1550
1540 K=FNT(L,X3,X4,X1,X2)
1550 X3=X1
1560 X4=X2
1570 GOTO 1500
1580 K=FNT(3,0,0,0,0)
1590 STOP
1600 END
1610 REM SUBROUTINA DE CAMBIO DE COORDENADAS
1620 REM
1630 X1=E2-E1*.707
1640 X2=E3-E1*.707
1650 REM
1660 RETURN
```

EXA2

```
1120 DIM A9#[2],B9#[2],O9#[80]
1130 DIM E1[16],E2[16],E3[16],X1[16],X2[16]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH14,OUTPUT1,INPUT2,TRAS
1160 INPUT "L = FACTOR DE TRASLACION X = ",L
1170 INPUT "M = FACTOR DE TRASLACION Y = ",M
1180 INPUT "N = FACTOR DE TRASLACION Z = ",N
1190 REM LECTURA DE LOS PUNTOS DEL FICHERO
1200 L=1
1210 FOR I=1 TO 16
1220   READ #2;E1[I]
1230   READ #2;E2[I]
1240   READ #2;E3[I]
1250 NEXT I
1260 REM CALULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1270 REM
1280 P=16
1290 K1=FNA(P,E1[*],E2[*],E3[*],L,M,N)
1300 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1310 FOR I=1 TO 16
1320   PRINT #3;E1[I]
1330   PRINT #3;E2[I]
1340   PRINT #3;E3[I]
1350 NEXT I
1360 REM CAMBIO DE COORDENADAS
1370 FOR I=1 TO 16
1380   X1[I]=E2[I]-E1[I]*.707
1390   X2[I]=E3[I]-E1[I]*.707
1400   REM
1410   PRINT #4;X1[I]
1420   PRINT #4;X2[I]
1430 NEXT I
1440 REM CAMBIO DE COORDENADAS A HEX.
1450 K1=FNT(2,0,0,0,0)
1460 K1=FNT(1,0,0,0,0)
1470 X3=X4=0
1480 L=1
1490 FOR I=1 TO 16
1500   IF X1[I]>256 OR X1[I]<0 OR X2[I]>256 OR X2[I]<0 THEN 1530
1510   K=FNT(L,X3,X4,X1[I],X2[I])
1520   L=0
1530   X3=X1[I]
1540   X4=X2[I]
1550 NEXT I
1560 K=FNT(3,0,0,0,0)
1570 STOP
1580 END
1590 DEF FNA(P,E1[*],E2[*],E3[*],L,M,N)
1600   DIM VC[100,4],UC[100,4],TC[4,4]
1610   MAT U=ZER(100,4)
1620   FOR I=1 TO P
1630     UC[I,1]=E1[I]
1640     UC[I,2]=E2[I]
1650     UC[I,3]=E3[I]
1660     UC[I,4]=1
1670   NEXT I
1680   MAT T=ZER(4,4)
1690   LET TC[1,1]=TC[2,2]=1
1700   LET TC[3,3]=TC[4,4]=1
1710   LET TC[4,1]=L
```



```
1720 LET T[4,2]=M
1730 LET T[4,3]=N
1740 MAT V=U*T
1750 FOR I=1 TO P
1760 LET E1[I]=V[I,1]
1770 LET E2[I]=V[I,2]
1780 LET E3[I]=V[I,3]
1790 NEXT I
1800 RETURN P
1810 FNEND
1820 RETURN
```

>

EXA5

```
1120 DIM A9%(2),B9%(2),O9%(80)
1130 DIM E1(16),E2(16),E3(16),X1(16),X2(16)
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH15,OUTPUT1,INPUT3,ESTR
1160 INPUT "A=FACTOR DE ESTRECHAMIENTO X=",A
1170 INPUT "E=FACTOR DE ESTRECHAMIENTO Y=",E
1180 INPUT "J=FACTOR DE ESTRECHAMIENTO Z=",J
1190 INPUT "S=FACTOR DE ESTRECHAMIENTOTOTAL =",S
1200 REM LECTURA DE LOS PUNTOS DEL FICHERO
1210 FOR I=1 TO 16
1220   READ #2:E1(I)
1230   READ #2:E2(I)
1240   READ #2:E3(I)
1250 NEXT I
1260 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1270 REM
1280 P=16
1290 K1=FNB(P,E1[*],E2[*],E3[*],A,E,J)
1300 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1310 FOR I=1 TO 16
1320   PRINT #3:E1(I)
1330   PRINT #3:E2(I)
1340   PRINT #3:E3(I)
1350 NEXT I
1360 REM CAMBIO DE COORDENADAS
1370 FOR I=1 TO 16
1380   X1(I)=E2(I)-E1(I)*.707
1390   X2(I)=E3(I)-E1(I)*.707
1400   REM
1410   PRINT #4;X1(I)
1420   PRINT #4;X2(I)
1430 NEXT I
1440 REM CAMBIO DE COORDENADAS A HEX.
1450 K1=FNT(2,0,0,0,0)
1460 K1=FNT(1,0,0,0,0)
1470 X3=X4=0
1480 L=1
1490 FOR I=1 TO 16
1500   IF X1(I)>256 OR X1(I)<0 OR X2(I)>256 OR X2(I)<0 THEN 1530
1510   K=FNT(L,X3,X4,X1(I),X2(I))
1520   L=0
1530   X3=X1(I)
1540   X4=X2(I)
1550 NEXT I
1560 K=FNT(3,0,0,0,0)
1570 STOP
1580 END
1590 DEF FNB(P,E1[*],E2[*],E3[*],A,E,J)
1600   DIM VC(100,4),UC(100,4),TC(4,4)
1610   MAT UC=ZER(100,4)
1620   FOR I=1 TO P
1630     LET UC(I,1)=E1(I)
1640     LET UC(I,2)=E2(I)
1650     LET UC(I,3)=E3(I)
1660     LET UC(I,4)=1
1670   NEXT I
1680   MAT T=ZER(4,4)
1690   LET T(1,1)=A
1700   LET T(2,2)=E
1710   LET T(3,3)=J
```

```
1720 LET T[4,4]=1/S
1730 MAT V=U*T
1740 FOR I=1 TO P
1750   LET E1[I]=V[I,1]/V[I,4]
1760   LET E2[I]=V[I,2]/V[I,4]
1770   LET E3[I]=V[I,3]/V[I,4]
1780 NEXT I
1790 RETURN P
1800 FNEND
1810 RETURN
```

EXA4

```
1120 DIM A9#[2],B9#[2],O9#[80]
1130 DIM E1[16],E2[16],E3[16],X1[16],X2[16]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH16,OUTPUT1,INPUT4,ROT
1160 INPUT "N1=COSENO DEL EJE DE ROTACION X=";N1
1170 INPUT "N2=COSENO DEL EJE DE ROTACION Y=";N2
1180 INPUT "N3=COSENO DEL EJE DE ROTACION Z=";N3
1190 INPUT "T1=ROTACION DEL ANGULO EN GRADOS=";T1
1200 REM LECTURA DE LOS PUNTOS DEL FICHERO
1210 L=1
1220 FOR I=1 TO 16
1230   READ #2:E1[I]
1240   READ #2:E2[I]
1250   READ #2:E3[I]
1260 NEXT I
1270 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1280 REM
1290 P=16
1300 K4=FNC(P,E1[*],E2[*],E3[*],N1,N2,N3)
1310 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1320 FOR I=1 TO 16
1330   PRINT #3:E1[I]
1340   PRINT #3:E2[I]
1350   PRINT #3:E3[I]
1360 NEXT I
1370 REM CAMBIO DE COORDENADAS
1380 FOR I=1 TO 16
1390   X1[I]=128+E2[I]-E1[I]*.707
1400   X2[I]=128+E3[I]-E1[I]*.707
1410   REM
1420   PRINT #4:X1[I]
1430   PRINT #4:X2[I]
1440 NEXT I
1450 REM CAMBIO DE COORDENADAS A HEX.
1460 K2=FNT(2,0,0,0,0)
1470 K2=FNT(1,0,0,0,0)
1480 X3=X4=0
1490 L=1
1500 FOR I=1 TO 16
1510   IF X1[I]>256 OR X1[I]<0 OR X2[I]>256 OR X2[I]<0 THEN 1540
1520   K=FNT(L,X3,X4,X1[I],X2[I])
1530   L=0
1540   X3=X1[I]
1550   X4=X2[I]
1560 NEXT I
1570 K2=FNT(3,0,0,0,0)
1580 STOP
1590 END
1600 DEF FNC(P,E1[*],E2[*],E3[*],N1,N2,N3)
1610   DIM U[100,4],V[100,4],T[4,4]
1620   MAT U=ZER(100,4)
1630   FOR I=1 TO P
1640     LET U[I,1]=E1[I]
1650     LET U[I,2]=E2[I]
1660     LET U[I,3]=E3[I]
1670   NEXT I
1680   MAT T=ZER(4,4)
1690   LET T2=T1/57.2958
1700   LET T[4,4]=1
1710   LET T[1,1]=N1*N1+(1-N1*N1)*COS(T2)
1720   LET T[1,2]=N1*N2*(1-COS(T2))+N3*SIN(T2)
```

```
1730 LET T[1,3]=N1*N3*(1-COS(T2))-N2*SIN(T2)
1740 LET T[2,1]=N1*N2*(1-COS(T2))-N3*SIN(T2)
1750 LET T[2,2]=N2*N2+(1-N2*N2)*COS(T2)
1760 LET T[2,3]=N2*N3*(1-COS(T2))+N1*SIN(T2)
1770 LET T[3,1]=N1*N3*(1-COS(T2))+N2*SIN(T2)
1780 LET T[3,2]=N2*N3*(1-COS(T2))-N1*SIN(T2)
1790 LET T[3,3]=N3*N3+(1-N3*N3)*COS(T2)
1800 MAT V=U*T
1810 FOR I=1 TO P
1820 LET E1[I]=V[I,1]
1830 LET E2[I]=V[I,2]
1840 LET E3[I]=V[I,3]
1850 NEXT I
1860 RETURN P
1870 FNEND
```

EXAD

```
1120 DIM A9#[2],B9#[2],O9#[80]
1130 DIM E1[16],E2[16],E3[16],X1[16],X2[16]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH17,OUTPUT1,INPUT5,REFLE
1160 INPUT "N=CODIGO DE REFLEXION N=",N
1170 REM LECTURA DE LOS PUNTOS DEL FICHERO
1180 FOR I=1 TO 16
1190   READ #2:E1[I]
1200   READ #2:E2[I]
1210   READ #2:E3[I]
1220 NEXT I
1230 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1240 REM
1250 P=16
1260 K4=FND(P,E1[*],E2[*],E3[*],N)
1270 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1280 FOR I=1 TO 16
1290   PRINT #3:E1[I]
1300   PRINT #3:E2[I]
1310   PRINT #3:E3[I]
1320 NEXT I
1330 REM CAMBIO DE COORDENADAS
1340 FOR I=1 TO 16
1350   X1[I]=180+E2[I]-E1[I]*.707
1360   X2[I]=180+E3[I]-E1[I]*.707
1370   REM
1380   PRINT #4:X1[I]
1390   PRINT #4:X2[I]
1400 NEXT I
1410 REM CAMBIO DE COORDENADAS A HEX.
1420 K2=FNT(2,0,0,0,0)
1430 K2=FNT(1,0,0,0,0)
1440 X3=X4=0
1450 L=1
1460   FOR I=1 TO 16
1470     IF X1[I]<0 THEN X1[I]=X1[I]+360
1480     K=FNT(L,X3,X4,X1[I],X2[I])
1490     L=0
1500     X3=X1[I]
1510     X4=X2[I]
1520 NEXT I
1530 K2=FNT(3,0,0,0,0)
1540 STOP
1550 END
1560 DEF FND(P,E1[*],E2[*],E3[*],N)
1570   DIM UC[100,4],VC[100,4],TC[4,4]
1580   MAT U=ZER(100,4)
1590   FOR I=1 TO P
1600     LET UC[I,1]=E1[I]
1610     LET UC[I,2]=E2[I]
1620     LET UC[I,3]=E3[I]
1630   NEXT I
1640   MAT T=ZER(4,4)
1650   IF N=2 THEN 1700
1660   IF N=3 THEN 1730
1670   LET TC[1,1]=TC[2,2]=TC[4,4]=1
1680   LET TC[3,3]=-1
1690   GOTO 1750
1700   LET TC[4,4]=TC[3,3]=TC[2,2]=1
1710   LET TC[1,1]=-1
1720   GOTO 1750
1730   LET TC[1,1]=TC[3,3]=TC[4,4]=1
```

```
1740 LET T[2,2]= -1
1750 MAT V=U*T
1760 FOR I=1 TO P
1770 LET E1[I]=V[I,1]
1780 LET E2[I]=V[I,2]
1790 LET E3[I]=V[I,3]
1800 NEXT I
1810 RETURN P
1820 FNEND
```

>

EXA2

```
1120 DIM A9%(2),B9%(2),O9%(80)
1130 DIM E1(16),E2(16),E3(16),X1(16),X2(16)
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH19,OUTPUT1,INPUT7,DIME1
1170 REM LECTURA DE LOS PUNTOS DEL FICHERO
1180 FOR I=1 TO 16
1190   READ #2:E1(I)
1200   READ #2:E2(I)
1210   READ #2:E3(I)
1220 NEXT I
1230 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1240 REM
1250 P=16
1260 K1=FNA(P,E1[*],E2[*],E3[*])
1270 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1280 FOR I=1 TO 16
1290   PRINT #3:E1(I)
1300   PRINT #3:E2(I)
1310   PRINT #3:E3(I)
1320 NEXT I
1330 REM CAMBIO DE COORDENADAS
1340 FOR I=1 TO 16
1350   X1(I)=128+E3(I)-E1(I)*.707
1360   X2(I)=128+E3(I)-E1(I)*.707
1370   REM
1380   PRIN* #4:X1(I)
1390   PRINT #4:X2(I)
1400 NEXT I
1410 REM CAMBIO DE COORDENADAS A HEX.
1420 K2=FNT(2,0,0,0,0)
1430 K2=FNT(1,0,0,0,0)
1440 X3=X4=0
1450 L=1
1460 FOR I=1 TO 16
1470   IF X1(I)>256 OR X1(I)<0 OR X2(I)>256 OR X2(I)<0 THEN 1490
1480   K=FNT(L,X3,X4,X1(I),X2(I))
1485   L=0
1490   X3=X1(I)
1500   X4=X2(I)
1510 NEXT I
1520 K2=FNT(3,0,0,0,0)
1530 STOP
1540 END
1550 DEF FNA(P,E1[*],E2[*],E3[*])
1560   DIM V(100,4),U(100,4),T(4,4)
1570   MAT U=ZER(100,4)
1580   MAT V=ZER(100,4)
1590   FOR I=1 TO P
1600     U(I,1)=E1(I)
1610     U(I,2)=E2(I)
1620     U(I,3)=E3(I)
1630     U(I,4)=1
1640   NEXT I
1650   MAT T=ZER(4,4)
1660   LET T(1,1)=.92582
1670   LET T(1,2)=.133631
1680   LET T(2,3)=.353553
1690   LET T(1,3)=-.353553
1700   LET T(2,2)=.935414
1710   LET T(3,1)=.377964
```



```
1720 LET T[3,2]=- .327327
1730 LET T[3,3]=.866025
1740 LET T[4,4]=1
1750 MAT V=U*T
1760 FOR I=1 TO P
1770     LET E1[I]=V[I,1]
1780     LET E2[I]=V[I,2]
1790     LET E3[I]=V[I,3]
1800 NEXT I
1815 RETURN P
1820 FNEND
```

EXA7

```
1120 DIM A9#[2],B9#[2],O9#Y80]
1130 DIM E1[16],E2[16],E3[16],X1[16],X2[16]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH20,OUTPUT1,INPUT8,ISO
1170 REM LECTURA DE LOS PUNTOS DEL FICHERO
1180 FOR I=1 TO 16
1190   READ #2;E1[I]
1200   READ #2;E2[I]
1210   READ #2;E3[I]
1220 NEXT I
1230 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1240 REM
1250 P=16
1260 K1=FNB(P,E1[*],E2[*],E3[*])
1270 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1280 FOR I=1 TO 16
1290   PRINT #3;E1[I]
1300   PRINT #3;E2[I]
1310   PRINT #3;E3[I]
1320 NEXT I
1330 REM CAMBIO DE COORDENADAS
1340 FOR I=1 TO 16
1350   X1[I]=128+E2[I]-E1[I]*.707
1360   X2[I]=128+E3[I]-E1[I]*.707
1370   REM
1380   PRINT #4;X1[I]
1390   PRINT #4;X2[I]
1400 NEXT I
1410 REM CAMBIO DE COORDENADAS A HEX.
1420 K2=FNT(2,0,0,0,0)
1430 K2=FNT(1,0,0,0,0)
1440 X3=X4=0
1450 L=1
1460 FOR I=1 TO 16
1470   IF X1[I]>256 OR X1[I]<0 OR X2[I]>256 OR X2[I]<0 THEN 1490
1480   K=FNT(L,X3,X4,X1[I],X2[I])
1485   L=0
1490   X3=X1[I]
1500   X4=X2[I]
1510 NEXT I
1520 K2=FNT(3,0,0,0,0)
1530 STOP
1540 END
1550 DEF FNB(P,E1[*],E2[*],E3[*])
1560   DIM VC[100,4],UC[100,4],TC[4,4]
1570   MAT U=ZER(100,4)
1580   MAT V=ZER(100,4)
1590   FOR I=1 TO P
1600     UC[I,1]=E1[I]
1610     UC[I,2]=E2[I]
1620     UC[I,3]=E3[I]
1630     UC[I,4]=1
1640   NEXT I
1650   MAT T=ZER(4,4)
1660   LET TC[1,1]=TC[3,1]=.707107
1670   LET TC[1,2]=.408248
1680   LET TC[3,2]=- .408248
1690   LET TC[1,3]=- .577353
1700   LET TC[3,3]=.577353
1710   LET TC[2,2]=.816497
1720   LET TC[2,3]=.577345
```

```
1730 LET T[4,4]=1
1750 MAT V=U*T
1760 FOR I=1 TO P
1770     LET E1[I]=V[I,1]
1780     LET E2[I]=V[I,2]
1790     LET E3[I]=V[I,3]
1800 NEXT I
1815 RETURN P
1820 FNEND
```

>

EXA28

```
1120 DIM A9#[2],B9#[2],O9#[80]
1130 DIM E1[16],E2[16],E3[16],X1[16],X2[16]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH21,INPUT2,TRAES,CAMB1
1160 INPUT "A=FACTOR DE ESTRECHAMIENTO X=",A
1170 INPUT "E=FACTOR DE ESTRECHAMIENTO Y=",E
1180 INPUT "J=FACTOR DE ESTRECHAMIENTO Z=",J
1190 INPUT "S=FACTOR DE ESTRECHAMIENTOTOTAL =",S
1200 REM LECTURA DE LOS PUNTOS DEL FICHERO
1210 FOR I=1 TO 16
1220   READ #2:E1[I]
1230   READ #2:E2[I]
1240   READ #2:E3[I]
1250 NEXT I
1260 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1270 REM
1280 P=16
1290 K2=FNB(P,E1[*],E2[*],E3[*],A,E,J,S)
1300 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1310 FOR I=1 TO 16
1320   PRINT #3:E1[I]
1330   PRINT #3:E2[I]
1340   PRINT #3:E3[I]
1350 NEXT I
1360 REM CAMBIO DE COORDENADAS
1370 FOR I=1 TO 16
1380   X1[I]=E2[I]-E1[I]*.707
1390   X2[I]=E3[I]-E1[I]*.707
1400   REM
1410   PRINT #4;X1[I]
1420   PRINT #4;X2[I]
1430 NEXT I
1440 REM CAMBIO DE COORDENADAS A HEX.
1450 K2=FNT(2,0,0,0)
1460 K2=FNT(1,0,0,0)
1470 X3=X4=0
1480 L=1
1490 FOR I=1 TO 16
1500   IF X1[I]>256 OR X1[I]<0 OR X2[I]>256 OR X2[I]<0 THEN 1520
1510   K=FNT(L,X3,X4,X1[I],X2[I])
1515   L=0
1520   X3=X1[I]
1530   X4=X2[I]
1540 NEXT I
1550 K=FNT(3,0,0,0)
1560 STOP
1570 END
1580 DEF FNB(P,E1[*],E2[*],E3[*],A,E,J,S)
1590   DIM V[100,4],U[100,4],T[4,4]
1600   MAT U=ZER(100,4)
1610   FOR I=1 TO P
1620     LET U[I,1]=E1[I]
1630     LET U[I,2]=E2[I]
1640     LET U[I,3]=E3[I]
1650     LET U[I,4]=1
1660   NEXT I
1670   MAT T=ZER(4,4)
1680   LET T[1,1]=A
1690   LET T[2,2]=E
1700   LET T[3,3]=J
1710   LET T[4,4]=1/S
```

```
1720 MAT V=U*T
1730 FOR I=1 TO P
1740   LET E1[I]=V[I,1]/V[I,4]
1750   LET E2[I]=V[I,2]/V[I,4]
1760   LET E3[I]=V[I,3]/V[I,4]
1770 NEXT I
1780 RETURN P
1790 FNEND
```

>

EXAS

```
1120 DIM A9#[ 2 ], B9#[ 2 ], O9#[ 80 ]
1130 DIM E1[ 16 ], E2[ 16 ], E3[ 16 ], X1[ 16 ], X2[ 16 ]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH22, INPUT2, TRASRO, CAMB2
1160 INPUT "N1=COSENO DEL EJE DE ROTACION X="; N1
1170 INPUT "N2=COSENO DEL EJE DE ROTACION Y="; N2
1180 INPUT "N3=COSENO DEL EJE DE ROTACION Z="; N3
1190 INPUT "T1=ROTACION DEL ANGULO EN GRADOS="; T1
1200 REM LECTURA DE LOS PUNTOS DEL FICHERO
1210 FOR I=1 TO 16
1220   READ #2; E1[ I ]
1230   READ #2; E2[ I ]
1240   READ #2; E3[ I ]
1250 NEXT I
1260 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1270 REM
1280 P=16
1290 K4=FNC(P, E1[ * ], E2[ * ], E3[ * ], N1, N2, N3)
1300 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1310 FOR I=1 TO 16
1320   PRINT #3; E1[ I ]
1330   PRINT #3; E2[ I ]
1340   PRINT #3; E3[ I ]
1350 NEXT I
1360 REM CAMBIO DE COORDENADAS
1370 FOR I=1 TO 16
1380   X1[ I ]=120+E2[ I ]-E1[ I ]*.707
1390   X2[ I ]=120+E3[ I ]-E1[ I ]*.707
1400   REM
1410   PRINT #4; X1[ I ]
1420   PRINT #4; X2[ I ]
1430 NEXT I
1440 REM CAMBIO DE COORDENADAS A HEX.
1450 K2=FNT(2, 0, 0, 0, 0)
1460 K2=FNT(1, 0, 0, 0, 0)
1470 X3=X4=0
1480 L=1
1490 FOR I=1 TO 16
1500   IF X1[ I ]>256 OR X1[ I ]<0 OR X2[ I ]>256 OR X2[ I ]<0 THEN 1520
1510   K=FNT(L, X3, X4, X1[ I ], X2[ I ])
1515   L=0
1520   X3=X1[ I ]
1530   X4=X2[ I ]
1540 NEXT I
1550 K2=FNT(3, 0, 0, 0, 0)
1560 STOP
1570 END
1580 DEF FNC(P, E1[ * ], E2[ * ], E3[ * ], N1, N2, N3)
1590   DIM UC[ 100, 4 ], VC[ 100, 4 ], TC[ 4, 4 ]
1600   MAT U=ZER(100, 4)
1610   FOR I=1 TO P
1620     LET UC[ I, 1 ]=E1[ I ]
1630     LET UC[ I, 2 ]=E2[ I ]
1640     LET UC[ I, 3 ]=E3[ I ]
1650   NEXT I
1660   MAT T=ZER(4, 4)
1670   LET T2=T1/57.2958
1680   LET TC[ 4, 4 ]=1
1690   LET TC[ 1, 1 ]=N1*N1+(1-N1*N1)*COS(T2)
1700   LET TC[ 1, 2 ]=N1*N2*(1-COS(T2))+N3*SIN(T2)
```

```

1710 LET T[1,3]=N1*N3*(1-COS(T2))-N2*SIN(T2)
1720 LET T[2,1]=N1*N2*(1-COS(T2))-N3*SIN(T2)
1730 LET T[2,2]=N2*N2+(1-N2*N2)*COS(T2)
1740 LET T[2,3]=N2*N3*(1-COS(T2))+N1*SIN(T2)
1750 LET T[3,1]=N1*N3*(1-COS(T2))+N2*SIN(T2)
1760 LET T[3,2]=N2*N3*(1-COS(T2))-N1*SIN(T2)
1770 LET T[3,3]=N3*N3+(1-N3*N3)*COS(T2)
1780 MAT V=U*T
1790 FOR I=1 TO P
1800 LET E1[I]=V[I,1]
1810 LET E2[I]=V[I,2]
1820 LET E3[I]=V[I,3]
1830 NEXT I
1840 RETURN P
1850 FNEND

```

EXA10

```
1120 DIM A9$(2),B9$(2),O9$(80)
1130 DIM E1(16),E2(16),E3(16),X1(16),X2(16)
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH23,INPUT3,ESTRERO,CAMB3
1120 INPUT "N1=COSENO DEL EJE DE ROTACION X=";N1
1170 INPUT "N2=COSENO DEL EJE DE ROTACION Y=";N2
1180 INPUT "N3=COSENO DEL EJE DE ROTACION Z=";N3
1190 INPUT "T1=ROTACION DEL ANGULO EN GRADOS=";T1
1200 REM LECTURA DE LOS PUNTOS DEL FICHERO
1210 FOR I=1 TO 16
1220   READ #2;E1(I)
1230   READ #2;E2(I)
1240   READ #2;E3(I)
1250 NEXT I
1260 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1270 REM
1280 P=16
1290 K4=FNC(P,E1[*],E2[*],E3[*],N1,N2,N3)
1300 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1310 FOR I=1 TO 16
1320   PRINT #3;E1(I)
1330   PRINT #3;E2(I)
1340   PRINT #3;E3(I)
1350 NEXT I
1360 REM CAMBIO DE COORDENADAS
1370 FOR I=1 TO 16
1380   X1(I)=120+E2(I)-E1(I)*.707
1390   X2(I)=120+E3(I)-E1(I)*.707
1400   REM
1410   PRINT #4;X1(I)
1420   PRINT #4;X2(I)
1430 NEXT I
1440 REM CAMBIO DE COORDENADAS A HEX.
1450 K2=FNT(2,0,0,0,0)
1460 K2=FNT(1,0,0,0,0)
1470 X3=X4=0
1480 L=1
1490 FOR I=1 TO 16
1500   IF X1(I)>256 OR X1(I)<0 OR X2(I)>256 OR X2(I)<0 THEN 1520
1510   K=FNT(L,X3,X4,X1(I),X2(I))
1515   L=0
1520   X3=X1(I)
1530   X4=X2(I)
1540 NEXT I
1550 K2=FNT(3,0,0,0,0)
1560 STOP
1570 END
1580 DEF FNC(P,E1[*],E2[*],E3[*],N1,N2,N3)
1590   DIM U(100,4),V(100,4),T(4,4)
1600   MAT U=ZER(100,4)
1610   FOR I=1 TO P
1620     LET U(I,1)=E1(I)
1630     LET U(I,2)=E2(I)
1640     LET U(I,3)=E3(I)
1650   NEXT I
1660   MAT T=ZER(4,4)
1670   LET T2=T1/57.2958
1680   LET T(4,4)=1
1690   LET T(1,1)=N1*N1+(1-N1*N1)*COS(T2)
1700   LET T(1,2)=N1*N2*(1-COS(T2))+N3*SIN(T2)
```



```

1710 LET T[1,3]=N1*N3*(1-COS(T2))-N2*SIN(T2)
1720 LET T[2,1]=N1*N2*(1-COS(T2))-N3*SIN(T2)
1730 LET T[2,2]=N2*N2+(1-N2*N2)*COS(T2)
1740 LET T[2,3]=N2*N3*(1-COS(T2))+N1*SIN(T2)
1750 LET T[3,1]=N1*N3*(1-COS(T2))+N2*SIN(T2)
1760 LET T[3,2]=N2*N3*(1-COS(T2))/N1*SIN(T2)
1770 LET T[3,3]=N3*N65.7LKq3*N3)*COS(T2)
1780 MAT V=U*T
1790 FOR I=1 TO P
1800     LET E1[I]=V[I,1]
1810     LET E2[I]=V[I,2]
1820     LET E3[I]=V[I,3]
1830 NEXT I
1840 RETURN P
1850 FNEND

```

EXA11

```
1120 DIM A9#[2],B9#[2],O9#[80]
1130 DIM E1[16],E2[16],E3[16],X1[16],X2[16]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH24,TRAES,ROTAR,CAMB4
1160 INPUT "N1=COSENO DEL EJE DE ROTACION X=";N1
1170 INPUT "N2=COSENO DEL EJE DE ROTACION Y=";N2
1180 INPUT "N3=COSENO DEL EJE DE ROTACION Z=";N3
1190 INPUT "T1=ROTACION DEL ANGULO EN GRADOS=";T1
1200 REM LECTURA DE LOS PUNTOS DEL FICHERO
1210 FOR I=1 TO 16
1220   READ #2;E1[I]
1230   READ #2;E2[I]
1240   READ #2;E3[I]
1250 NEXT I
1260 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1270 REM
1280 P=16
1290 K4=FNC(P,E1[*],E2[*],E3[*],N1,N2,N3)
1300 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1310 FOR I=1 TO 16
1320   PRINT #3;E1[I]
1330   PRINT #3;E2[I]
1340   PRINT #3;E3[I]
1350 NEXT I
1360 REM CAMBIO DE COORDENADAS
1370 FOR I=1 TO 16
1380   X1[I]=120+E2[I]-E1[I]*.707
1390   X2[I]=120+E3[I]-E1[I]*.707
1400   REM
1410   PRINT #4;X1[I]
1420   PRINT #4;X2[I]
1430 NEXT I
1440 REM CAMBIO DE COORDENADAS A HEX.
1450 K2=FNT(2,0,0,0,0)
1460 K2=FNT(1,0,0,0,0)
1470 X3=X4=0
1480 L=1
1490 FOR I=1 TO 16
1500   IF X1[I]>256 OR X1[I]<0 OR X2[I]>256 OR X2[I]<0 THEN 1520
1510   K=FNT(L,X3,X4,X1[I],X2[I])
1515   L=0
1520   X3=X1[I]
1530   X4=X2[I]
1540 NEXT I
1550 K2=FNT(3,0,0,0,0)
1560 STOP
1570 END
1580 DEF FNC(P,E1[*],E2[*],E3[*],N1,N2,N3)
1590   DIM UC[100,4],VC[100,4],T[4,4]
1600   MAT U=ZER(100,4)
1610   FOR I=1 TO P
1620     LET UC[I,1]=E1[I]
1630     LET UC[I,2]=E2[I]
1640     LET UC[I,3]=E3[I]
1650   NEXT I
1660   MAT T=ZER(4,4)
1670   LET T2=T1/57.2958
1680   LET T[4,4]=1
1690   LET T[1,1]=N1*N1+(1-N1*N1)*COS(T2)
1700   LET T[1,2]=N1*N2*(1-COS(T2))+N3*SIN(T2)
1710   LET T[1,3]=N1*N3*(1-COS(T2))-N2*SIN(T2)
1720   LET T[2,1]=N1*N2*(1-COS(T2))-N3*SIN(T2)
```

```
1730 LET T[2,2]=N2*N2+(1-N2*N2)*COS(T2)
1740 LET T[2,3]=N2*N3*(1-COS(T2))+N1*SIN(T2)
1750 LET T[3,1]=N1*N3*(1-COS(T2))+N2*SIN(T2)
1760 LET T[3,2]=N2*N3*(1-COS(T2))-N1*SIN(T2)
1770 LET T[3,3]=N3*N3+(1-N3*N3)*COS(T2)
1780 MAT V=U*T
1790 FOR I=1 TO P
1800 LET E1[I]=V[I,1]
1810 LET E2[I]=V[I,2]
1820 LET E3[I]=V[I,3]
1830 NEXT I
1840 RETURN P
1850 FNEND
```

```

EXA12
1120 DIM A9#[2],B9#[2],O9#[80]
1130 DIM E1[16],E2[16],E3[16],X1[16],X2[16]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPHOS+TRASP+RE+...
1160 ...
1170 REM LECTURA DE LOS PUNTOS DEL FICHERO
1180 FOR I=1 TO 16
1190   READ #2;E1[I]
1200   READ #2;E2[I]
1210   READ #2;E3[I]
1220 NEXT I
1230 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1240 REM
1250 P=16
1260 K4=FND(P,E1[*],E2[*],E3[*],N)
1270 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1280 FOR I=1 TO 16
1290   PRINT #3;E1[I]
1300   PRINT #3;E2[I]
1310   PRINT #3;E3[I]
1320 NEXT I
1330 REM CAMBIO DE COORDENADAS
1340 FOR I=1 TO 16
1350   X1[I]=180+E2[I]-E1[I]*.707
1360   X2[I]=180+E3[I]-E1[I]*.707
1370   REM
1380   PRINT #4;X1[I]
1390   PRINT #4;X2[I]
1400 NEXT I
1410 REM CAMBIO DE COORDENADAS A HEX.
1420 K2=FNT(2,0,0,0,0)
1430 K2=FNT(1,0,0,0,0)
1440 X3=X4=0
1450 L=1
1460 FOR I=1 TO 16
1470   IF X1[I]>256 OR X1[I]<0 OR X2[I]>256 OR X2[I]<0 THEN 1490
1480   K=FNT(L,X3,X4,X1[I],X2[I])
1485   L=0
1490   X3=X1[I]
1500   X4=X2[I]
1510 NEXT I
1520 K2=FNT(3,0,0,0,0)
1530 STOP
1540 END
1550 DEF FND(P,E1[*],E2[*],E3[*],N)
1560   DIM U[100,4],V[100,4],T[4,4]
1570   MAT U=ZER(100,4)
1580   FOR I=1 TO P
1590     LET U[I,1]=E1[I]
1600     LET U[I,2]=E2[I]
1610     LET U[I,3]=E3[I]
1620   NEXT I
1630   MAT T=ZER(4,4)
1640   IF N=2 THEN 1690
1650   IF N=3 THEN 1720
1660   LET T[1,1]=T[2,2]=T[4,4]=1
1670   LET T[3,3]=-1
1680   GOTO 1740
1690   LET T[4,4]=T[3,3]=T[2,2]=1
1700   LET T[1,1]=-1
1710   GOTO 1740

```

```
1720 LET TC[1,1]=TC[3,3]=TC[4,4]=1
1730 LET TC[2,2]=-1
1740 MAT V=U*T
1750 FOR I=1 TO P
1760     LET E1[I]=V[I,1]
1770     LET E2[I]=V[I,2]
1780     LET E3[I]=V[I,3]
1790 NEXT I
1800 RETURN P
1810 FNEND
```

EXA13

```
1120 DIM A9#[2],B9#[2],O9#[80]
1130 DIM E1[16],E2[16],E3[16],X1[16],X2[16]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH26,TRAES,REFLE2,CAMB6
1160 INPUT "N=CODIGO DE REFLEXION N=",N
1170 REM LECTURA DE LOS PUNTOS DEL FICHERO
1180 FOR I=1 TO 16
1190   READ #2;E1[I]
1200   READ #2;E2[I]
1210   READ #2;E3[I]
1220 NEXT I
1230 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1240 REM
1250 P=16
1260 K4=FND(P,E1[*],E2[*],E3[*],N)
1270 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1280 FOR I=1 TO 16
1290   PRINT #3;E1[I]
1300   PRINT #3;E2[I]
1310   PRINT #3;E3[I]
1320 NEXT I
1330 REM CAMBIO DE COORDENADAS
1340 FOR I=1 TO 16
1350   X1[I]=180+E2[I]-E1[I]*.707
1360   X2[I]=180+E3[I]-E1[I]*.707
1370   REM
1380   PRINT #4;X1[I]
1390   PRINT #4;X2[I]
1400 NEXT I
1410 REM CAMBIO DE COORDENADAS A HEX.
1420 K2=FNT(2,0,0,0,0)
1430 K2=FNT(1,0,0,0,0)
1440 X3=X4=0
1450 L=1
1460 FOR I=1 TO 16
1470   IF X1[I]>256 OR X1[I]<0 OR X2[I]>256 OR X2[I]<0 THEN 1490
1480   K=FNT(L,X3,X4,X1[I],X2[I])
1485   L=0
1490   X3=X1[I]
1500   X4=X2[I]
1510 NEXT I
1520 K2=FNT(3,0,0,0,0)
1530 STOP
1540 END
1550 DEF FND(P,E1[*],E2[*],E3[*],N)
1560   DIM UC[100,4],VC[100,4],TC[4,4]
1570   MAT U=ZER(100,4)
1580   FOR I=1 TO P
1590     LET UC[I,1]=E1[I]
1600     LET UC[I,2]=E2[I]
1610     LET UC[I,3]=E3[I]
1620   NEXT I
1630   MAT T=ZER(4,4)
1640   IF N=2 THEN 1690
1650   IF N=3 THEN 1720
1660   LET TC[1,1]=TC[2,2]=TC[4,4]=1
1670   LET TC[3,3]=-1
1680   GOTO 1740
1690   LET TC[4,4]=TC[3,3]=TC[2,2]=1
```

```
1700 LET T[1,1]=--1
1710 GOTO 1740
1720 LET T[1,1]=T[3,3]=T[4,4]=1
1730 LET T[2,2]=--1
1740 MAT V=U*T
1750 FOR I=1 TO P
1760 LET E1[I]=V[I,1]
1770 LET E2[I]=V[I,2]
1780 LET E3[I]=V[I,3]
1790 NEXT I
1800 RETURN P
1810 FNEND
```

>

EXA14

```
1120 DIM A9#[2],B9#[2],O9#[80]
1130 DIM E1[16],E2[16],E3[16],X1[16],X2[16]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH27,TRAES,REFLE3,CAMB7
1160 INPUT "N=CODIGO DE REFLEXION N=",N
1170 REM LECTURA DE LOS PUNTOS DEL FICHERO
1180 FOR I=1 TO 16
1190   READ #2;E1[I]
1200   READ #2;E2[I]
1210   READ #2;E3[I]
1220 NEXT I
1230 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1240 REM
1250 P=16
1260 K4=FND(P,E1[*],E2[*],E3[*],N)
1270 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1280 FOR I=1 TO 16
1290   PRINT #3;E1[I]
1300   PRINT #3;E2[I]
1310   PRINT #3;E3[I]
1320 NEXT I
1330 REM CAMBIO DE COORDENADAS
1340 FOR I=1 TO 16
1350   X1[I]=180+E2[I]-E1[I]*.707
1360   X2[I]=180+E3[I]-E1[I]*.707
1370   REM
1380   PRINT #4;X1[I]
1390   PRINT #4;X2[I]
1400 NEXT I
1410 REM CAMBIO DE COORDENADAS A HEX.
1420 K2=FNT(2,0,0,0,0)
1430 K2=FNT(1,0,0,0,0)
1440 X3=X4=0
1450 L=1
1460 FOR I=1 TO 16
1470   IF X1[I]>256 OR X1[I]<0 OR X2[I]>256 OR X2[I]<0 THEN 1490
1480   K=FNT(L,X3,X4,X1[I],X2[I])
1485   L=0
1490   X3=X1[I]
1500   X4=X2[I]
1510 NEXT I
1520 K2=FNT(3,0,0,0,0)
1530 STOP
1540 END
1550 DEF FND(P,E1[*],E2[*],E3[*],N)
1560   DIM UC[100,4],VC[100,4],TC[4,4]
1570   MAT U=ZER(100,4)
1580   FOR I=1 TO P
1590     LET UC[I,1]=E1[I]
1600     LET UC[I,2]=E2[I]
1610     LET UC[I,3]=E3[I]
1620   NEXT I
1630   MAT T=ZER(4,4)
1640   IF N=2 THEN 1690
1650   IF N=3 THEN 1720
1660   LET TC[1,1]=TC[2,2]=TC[4,4]=1
1670   LET TC[3,3]=-1
1680   GOTO 1740
1690   LET TC[4,4]=TC[3,3]=TC[2,2]=1
```



```
1700 LET TC 1, 1 ]=-1
1710 GOTO 1740
1720 LET TC 1, 1 ]=TC 3, 3 ]=TC 4, 4 ]=1
1730 LET TC 2, 2 ]=-1
1740 MAT V=U*T
1750 FOR I=1 TO P
1760     LET E1[I ]=V[I, 1 ]
1770     LET E2[I ]=V[I, 2 ]
1780     LET E3[I ]=V[I, 3 ]
1790 NEXT I
1800 RETURN P
1810 FNEND
```

EXA15

```
1120 DIM A9#[2],B9#[2],O9#[80]
1130 DIM E1[16],E2[16],E3[16],X1[16],X2[16]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH61,INPUT3,ISO16,CAMB51
1170 REM LECTURA DE LOS PUNTOS DEL FICHERO
1180 FOR I=1 TO 16
1190   READ #2;E1[I]
1200   READ #2;E2[I]
1210   READ #2;E3[I]
1220 NEXT I
1230 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1240 REM
1250 P=16
1260 K1=FNB(P,E1[*],E2[*],E3[*])
1270 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1280 FOR I=1 TO 16
1290   PRINT #3;E1[I]
1300   PRINT #3;E2[I]
1310   PRINT #3;E3[I]
1320 NEXT I
1330 REM CAMBIO DE COORDENADAS
1340 FOR I=1 TO 16
1350   X1[I]=128+E2[I]-E1[I]*.707
1360   X2[I]=128+E3[I]-E1[I]*.707
1370   REM
1380   PRINT #4;X1[I]
1390   PRINT #4;X2[I]
1400 NEXT I
1410 REM CAMBIO DE COORDENADAS A HEX.
1420 K2=FNT(2,0,0,0,0)
1430 K2=FNT(1,0,0,0,0)
1440 X3=X4=0
1450 L=1
1460 FOR I=1 TO 16
1470   IF X1[I]>256 OR X1[I]<0 OR X2[I]>256 OR X2[I]<0 THEN 1490
1480   K=FNT(L,X3,X4,X1[I],X2[I])
1485   L=0
1490   X3=X1[I]
1500   X4=X2[I]
1510 NEXT I
1520 K2=FNT(3,0,0,0,0)
1530 STOP
1540 END
1550 DEF FNB(P,E1[*],E2[*],E3[*])
1560   DIM VC[100,4],UC[100,4],TC[4,4]
1570   MAT U=ZER(100,4)
1580   MAT V=ZER(100,4)
1590   FOR I=1 TO P
1600     UC[I,1]=E1[I]
1610     UC[I,2]=E2[I]
1620     UC[I,3]=E3[I]
1630     UC[I,4]=1
1640   NEXT I
1650   MAT T=ZER(4,4)
1660   LET TC[1,1]=TC[3,1]=.707107
1670   LET TC[1,2]=.408248
1680   LET TC[3,2]=- .408248
1690   LET TC[1,3]=- .577353
1700   LET TC[3,3]=.577353
1710   LET TC[2,2]=.816497
1720   LET TC[2,3]=.577345
1730   LET TC[4,4]=1
```

```
1750 MAT V=U*T
1760 FOR I=1 TO P
1770     LET E1[I]=V[I,1]
1780     LET E2[I]=V[I,2]
1790     LET E3[I]=V[I,3]
1800 NEXT I
1815 RETURN P
1820 FNEND
```

EXA16

```
1120 DIM A9#[2],B9#[2],O9#[80]
1130 DIM E1[16],E2[16],E3[16],X1[16],X2[16]
1140 REM NOMBRE DE LA BASE DE DATOS
1150 FILES GRAPH65,INPUT3,DIME22,CAMB32
1170 REM LECTURA DE LOS PUNTOS DEL FICHERO
1180 FOR I=1 TO 16
1190   READ #2;E1[I]
1200   READ #2;E2[I]
1210   READ #2;E3[I]
1220 NEXT I
1230 REM CALCULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1240 REM
1250 P=16
1260 K1=FNA(P,E1[*],E2[*],E3[*])
1270 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1280 FOR I=1 TO 16
1290   PRINT #3;E1[I]
1300   PRINT #3;E2[I]
1310   PRINT #3;E3[I]
1320 NEXT I
1330 REM CAMBIO DE COORDENADAS
1340 FOR I=1 TO 16
1350   X1[I]=128+E2[I]-E1[I]*.707
1360   X2[I]=128+E3[I]-E1[I]*.707
1370   REM
1380   PRINT #4;X1[I]
1390   PRINT #4;X2[I]
1400 NEXT I
1410 REM CAMBIO DE COORDENADAS A HEX.
1420 K2=FNT(2,0,0,0,0)
1430 K2=FNT(1,0,0,0,0)
1440 X3=X4=0
1450 L=1
1460 FOR I=1 TO 16
1470   IF X1[I]>256 OR X1[I]<0 OR X2[I]>256 OR X2[I]<0 THEN 1490
1480   K=FNT(L,X3,X4,X1[I],X2[I])
1485   L=0
1490   X3=X1[I]
1500   X4=X2[I]
1510 NEXT I
1520 K2=FNT(3,0,0,0,0)
1530 STOP
1540 END
1550 DEF FNA(P,E1[*],E2[*],E3[*])
1560   DIM VC[100,4],UC[100,4],TC[4,4]
1570   MAT U=ZER(100,4)
1580   MAT V=ZER(100,4)
1590   FOR I=1 TO P
1600     UC[I,1]=E1[I]
1610     UC[I,2]=E2[I]
1620     UC[I,3]=E3[I]
1630     UC[I,4]=1
1640   NEXT I
1650   MAT T=ZER(4,4)
1660   LET TC[1,1]=.92582
1670   LET TC[1,2]=.133631
1680   LET TC[2,3]=.353553
1690   LET TC[1,3]=- .353553
1700   LET TC[2,2]=.935414
1710   LET TC[3,1]=.377964
```

```
1720 LET T[3,2]=- .327327
1730 LET T[3,3]=.866025
1740 LET T[4,4]=1
1750 MAT V=U*T
1760 FOR I=1 TO P
1770     LET E1[I]=V[I,1]
1780     LET E2[I]=V[I,2]
1790     LET E3[I]=V[I,3]
1800 NEXT I
1815 RETURN P
1820 FNEND
```

EXA17

```
1120 DIM A9#[ 2 ],B9#[ 2 ],O9#[ 80 ]
1125 DIM E1[ 16 ],E2[ 16 ],E3[ 16 ],X1[ 16 ],X2[ 16 ]
1130 REM NOMBRE DE LA BASE DE DATOS
1140 FILES GRAPH141,OUTPUT1,INPUT22,TRAS1
1141 INPUT "L = FACTOR DE TRASLACION X = ",L
1142 INPUT "M = FACTOR DE TRASLACION Y = ",M
1143 INPUT "N = FACTOR DE TRASLACION Z = ",N
1220 REM LECTURA DE LOS PUNTOS DEL FICHERO
1225 L=1
1230 FOR I=1 TO 16
1240   READ #2;E1[ I ]
1250   READ #2;E2[ I ]
1260   READ #2;E3[ I ]
1270 NEXT I
1280 REM CALULO DE LOS NUEVOS PUNTOS MEDIANTE LA FUNCION
1290 REM
1300 P=16
1310 K1=FNA(P,E1[* ],E2[* ],E3[* ],L,M,N)
1320 REM ESCRITURA DE LOS NUEVOS PUNTOS EN EL FICHERO
1330 FOR I=1 TO 16
1340   PRINT #3;E1[ I ]
1350   PRINT #3;E2[ I ]
1360   PRINT #3;E3[ I ]
1370 NEXT I
1380 REM CAMBIO DE COORDENADAS
1390 FOR I=1 TO 16
1400   X1[ I ]=128+E1[ I ]/(E3[ I ]+1)*25
1410   X2[ I ]=128-E2[ I ]/(E3[ I ]+1)*25
1420   REM
1430   PRINT #4;X1[ I ]
1440   PRINT #4;X2[ I ]
1450 NEXT I
1460 REM CAMBIO DE COORDENADAS A HEX.
1470 K1=FNT(2,0,0,0,0)
1480 K1=FNT(1,0,0,0,0)
1490 X3=X4=0
1500 L=1
1510 FOR I=1 TO 16
1520   IF X1[ I ]>256 OR X1[ I ]<0 OR X2[ I ]>256 OR X2[ I ]<0 THEN 1550
1530   K=FNT(L,X3,X4,X1[ I ],X2[ I ])
1540   L=0
1550   X3=X1[ I ]
1560   X4=X2[ I ]
1570   L=L+1
1580   K=FNT(3,0,0,0,0)
1700 STOP
1710 END
```

## LISTADO DE FICHEROS

1	21.21
2	30
3	30
4	21.21
5	60
6	30
7	21.21
8	60
9	60
10	21.21
11	30
12	60
13	21.21
14	30
15	30
16	-8.75
17	30
18	30
19	-8.75
20	30
21	60
22	21.21
23	30
24	60
25	21.21
26	60
27	60
28	-8.75
29	60
30	60
31	-8.75
32	30
33	60
34	-8.75
35	30
36	30
37	-8.75
38	60
39	30
40	-8.75
41	60
42	60
43	-8.75
44	60
45	30
46	21.21
47	60
48	30



1	15.0045
2	15.0045
3	45.0045
4	15.0045
5	45.0045
6	45.0045
7	15.0045
8	45.0045
9	15.0045
10	15.0045
11	36.1862
12	36.1862
13	36.1862
14	66.1862
15	15.0045
16	45.0045
17	45.0045
18	45.0045
19	66.1862
20	66.1862
21	36.1862
22	66.1862
23	36.1862
24	36.1862
25	66.1862
26	36.1862
27	66.1862
28	66.1862
29	66.1862
30	36.1862
31	45.0045
32	15.0045

1	15.2975
2	15.2975
3	45.2975
4	15.2975
5	45.2975
6	45.2975
7	15.2975
8	45.2975
9	15.2975
10	15.2975
11	56.4792
12	56.4792
13	56.4792
14	66.4792
15	15.2975
16	45.2975
17	45.2975
18	45.2975
19	66.4792
20	66.4792
21	56.4792
22	66.4792
23	56.4792
24	56.4792
25	66.4792
26	56.4792
27	66.4792
28	66.4792
29	66.4792
30	56.4792
31	45.2975
32	15.2975
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	
51	
52	
53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	

30	22.21
31	31
31	31
32	22.21
33	31
34	31
35	22.21
36	31
37	31
38	22.21
39	31
40	31
41	31
42	22.21
43	31
44	31
45	22.21
46	31
47	31
48	-7.75
49	31
50	31
51	-7.75
52	31
53	31
54	31
55	-7.75
56	31
57	31
58	-7.75
59	31
60	31
61	-7.75
62	31
63	31
64	-7.75
65	31
66	31
67	-7.75
68	31
69	31
70	-7.75
71	31
72	31
73	-7.75
74	31
75	31
76	-7.75
77	31
78	31
79	-7.75
80	31
81	31
82	-7.75
83	31
84	31
85	-7.75
86	31
87	31
88	-7.75
89	31
90	31
91	-7.75
92	31
93	31
94	-7.75
95	31
96	31
97	-7.75
98	31
99	31
100	-7.75
101	31
102	31
103	-7.75
104	31
105	31
106	-7.75
107	31
108	31
109	-7.75
110	31
111	31
112	-7.75
113	31
114	31
115	-7.75
116	31
117	31
118	-7.75
119	31
120	31
121	-7.75
122	31
123	31
124	-7.75
125	31
126	31
127	-7.75
128	31
129	31
130	-7.75
131	31
132	31
133	-7.75
134	31
135	31
136	-7.75
137	31
138	31
139	-7.75
140	31
141	31
142	-7.75
143	31
144	31
145	-7.75
146	31
147	31
148	-7.75
149	31
150	31
151	-7.75
152	31
153	31
154	-7.75
155	31
156	31
157	-7.75
158	31
159	31
160	-7.75
161	31
162	31
163	-7.75
164	31
165	31
166	-7.75
167	31
168	31
169	-7.75
170	31
171	31
172	-7.75
173	31
174	31
175	-7.75
176	31
177	31
178	-7.75
179	31
180	31
181	-7.75
182	31
183	31
184	-7.75
185	31
186	31
187	-7.75
188	31
189	31
190	-7.75
191	31
192	31
193	-7.75
194	31
195	31
196	-7.75
197	31
198	31
199	-7.75
200	31

1	22.21
2	31
3	31
4	22.21
5	61
6	31
7	22.21
8	61
9	61
10	22.21
11	31
12	61
13	22.21
14	31
15	31
16	-7.75
17	31
18	31
19	-7.75
20	31
21	61
22	22.21
23	31
24	61
25	22.21
26	61
27	61
28	-7.75
29	61
30	61
31	-7.75
32	31
33	61
34	-7.75
35	31
36	31
37	-7.75
38	61
39	31
40	-7.75
41	61
42	61
43	-7.75
44	61
45	31
46	22.21
47	61
48	31

1	10.605
2	9.9
3	30
4	10.605
5	19.8
6	30
7	10.605
8	19.8
9	60
10	10.605
11	9.9
12	30
13	10.605
14	9.9
15	30
16	-4.375
17	9.9
18	30
19	-4.375
20	9.9
21	60
22	10.605
23	9.9
24	60
25	10.605
26	19.8
27	60
28	-4.375
29	19.8
30	30
31	-4.375
32	9.9
33	60
34	-4.375
35	9.9
36	30
37	-4.375
38	19.8
39	30
40	-4.375
41	19.8
42	60
43	-4.375
44	19.8
45	30
46	10.605
47	19.8
48	30



1	21.2101
2	50.
3	50
4	21.2101
5	60.
6	50
7	21.2101
8	60.
9	60
10	21.2101
11	50.
12	60
13	21.2101
14	50.
15	50
16	-8.74995
17	50.
18	50
19	-8.74995
20	50.
21	60
22	21.2101
23	50.
24	60
25	21.2101
26	60.
27	60
28	-8.74986
29	60.
30	60
31	-8.74993
32	50.
33	60
34	-8.74995
35	50.
36	50
37	-8.74986
38	60.
39	50
40	-8.74986
41	60.
42	60
43	-8.74986
44	60.
45	50
46	21.2101
47	60.
48	50

1	143.004
2	143.004
3	173.004
4	143.004
5	173.004
6	173.004
7	143.004
8	173.004
9	143.004
10	143.004
11	164.186
12	164.186
13	164.186
14	194.186
15	143.004
16	173.004
17	173.004
18	173.004
19	194.186
20	194.186
21	164.186
22	194.186
23	164.186
24	164.186
25	194.186
26	164.186
27	194.186
28	194.186
29	194.186
30	164.186
31	173.004
32	143.004
33	143.643
34	-11.2746
35	43.526
36	63.6429
37	-24.7593
38	103.526
39	101.216
40	-7.18593
41	98.0413
42	101.216
43	-24.7593
44	103.526
45	101.216
46	35.1605
47	121.076
48	87.7496



1	21.21
2	-30
3	30
4	21.21
5	-60
6	30
7	21.21
8	-60
9	60
10	21.21
11	-30
12	60
13	21.21
14	-30
15	30
16	-8.75
17	-30
18	30
19	-8.75
20	-30
21	60
22	21.21
23	-30
24	60
25	21.21
26	-60
27	60
28	-8.75
29	-60
30	60
31	-8.75
32	-30
33	60
34	-8.75
35	-30
36	30
37	-8.75
38	-60
39	30
40	-8.75
41	-60
42	60
43	-8.75
44	-60
45	30
46	21.21
47	-60
48	30

1	135.005
2	195.005
3	105.005
4	195.005
5	105.005
6	225.005
7	135.005
8	225.005
9	135.005
10	195.005
11	156.186
12	216.186
13	156.186
14	246.186
15	135.005
16	225.005
17	105.005
18	225.005
19	126.186
20	246.186
21	156.186
22	246.186
23	156.186
24	216.186
25	126.186
26	216.186
27	126.186
28	246.186
29	126.186
30	216.186
31	105.005
32	195.005

1	50.9756
2	21.6769
3	29.0885
4	50.9756
5	49.1393
6	39.6951
7	42.3145
8	39.3195
9	65.6758
10	42.3145
11	11.2571
12	55.6692
13	50.9756
14	21.6769
15	29.0885
16	3.238
17	17.0733
18	39.6809
19	14.5769
20	7.25553
21	65.6617
22	42.3145
23	11.2571
24	55.6692
25	42.3145
26	39.3195
27	65.6758
28	14.5769
29	39.3195
30	76.2685
31	14.5769
32	7.25553
33	65.6617
34	3.238
35	17.0733
36	39.6809
37	3.238
38	45.1358
39	50.2875
40	14.5769
41	39.3195
42	76.2685
43	3.238
44	45.1358
45	50.2875
46	30.9756
47	49.1393
48	39.6951

1	127.177
2	135.189
3	155.24
4	145.795
5	137.403
6	163.759
7	109.341
8	153.153
9	127.177
10	135.189
11	142.784
12	165.392
13	124.948
14	183.356
15	109.341
16	153.153
17	137.403
18	163.759
19	153.01
20	193.962
21	124.948
22	183.356
23	142.784
24	165.392
25	170.846
26	175.998
27	153.01
28	193.962
29	170.846
30	175.998
31	155.24
32	145.795

1	36.2109
2	20.9064
3	22.3953
4	36.2109
5	45.4013
6	39.7156
7	57.4242
8	33.1539
9	57.0362
10	57.4242
11	6.65597
12	39.7159
13	36.2109
14	20.9064
15	22.3953
16	15.026
17	6.6753
18	39.6926
19	36.2392
20	-3.57214
21	57.0134
22	57.4042
23	6.65597
24	39.7159
25	57.4242
26	33.1539
27	57.0362
28	36.2392
29	20.9226
30	74.3537
31	36.2392
32	-3.57214
33	57.0134
34	15.026
35	6.6753
36	39.6926
37	15.026
38	33.1702
39	57.0131
40	36.2392
41	20.9226
42	74.3537
43	15.026
44	33.1702
45	57.0131
46	36.2109
47	45.4013
48	39.7156

1	123.305
2	124.794
3	147.6
4	142.115
5	120.555
6	144.437
7	96.0601
8	127.117
9	123.305
10	124.794
11	126.052
12	107.069
13	98.0067
14	109.392
15	96.0601
16	127.117
17	120.555
18	144.437
19	123.302
20	176.713
21	98.0067
22	109.392
23	126.052
24	107.069
25	150.547
26	174.39
27	123.302
28	176.713
29	150.547
30	174.39
31	147.6
32	142.115

1	11.105
2	10.23
3	31
4	11.105
5	20.13
6	31
7	11.105
8	20.13
9	61
10	11.105
11	10.23
12	61
13	11.105
14	10.23
15	31
16	-3.875
17	10.23
18	31
19	-3.875
20	10.23
21	61
22	11.105
23	10.23
24	61
25	11.105
26	20.13
27	61
28	-3.875
29	20.13
30	61
31	-3.875
32	10.23
33	61
34	-3.875
35	10.23
36	31
37	-3.875
38	20.13
39	31
40	-3.875
41	20.13
42	61
43	-3.875
44	20.13
45	31
46	11.105
47	20.13
48	31

1	2.37877
2	23.1488
3	12.2788
4	23.1488
5	12.2788
6	53.1488
7	2.37877
8	53.1488
9	2.37877
10	23.1488
11	12.9696
12	53.7396
13	12.9696
14	53.7396
15	2.37877
16	53.1488
17	12.2788
18	53.1488
19	12.8696
20	53.7396
21	12.9696
22	53.7396
23	12.9696
24	53.7396
25	12.8696
26	53.7396
27	12.8696
28	53.7396
29	12.8696
30	53.7396
31	12.2788
32	23.1488



1	22.21
2	1.52588E-05
3	43.8406
4	22.21
5	21.2132
6	65.0538
7	22.21
8	3.05176E-05
9	86.267
10	22.21
11	-21.2132
12	65.0538
13	22.21
14	1.52588E-05
15	43.8406
16	-7.75
17	1.52588E-05
18	43.8406
19	-7.75
20	-21.2132
21	65.0538
22	22.21
23	-21.2132
24	65.0538
25	22.21
26	3.05176E-05
27	86.267
28	-7.75
29	3.05176E-05
30	86.267
31	-7.75
32	-21.2132
33	65.0538
34	-7.75
35	1.52588E-05
36	43.8406
37	-7.75
38	21.2132
39	65.0538
40	-7.75
41	3.05176E-05
42	86.267
43	-7.75
44	21.2132
45	65.0538
46	22.21
47	21.2132
48	65.0538

1	104.298
2	148.138
3	125.511
4	169.351
5	104.298
6	190.565
7	83.0844
8	169.351
9	104.298
10	148.138
11	125.479
12	169.32
13	104.266
14	190.533
15	83.0844
16	169.351
17	104.298
18	190.565
19	125.479
20	211.746
21	104.266
22	190.533
23	125.479
24	169.32
25	146.692
26	190.533
27	125.479
28	211.746
29	146.692
30	190.533
31	125.511
32	169.351

1	10.605
2	9.89998
3	30
4	10.605
5	19.8
6	30
7	10.605
8	19.8
9	60
10	10.605
11	9.89998
12	60
13	10.605
14	9.89998
15	30
16	-4.37496
17	9.90001
18	30
19	-4.37496
20	9.90001
21	60
22	10.605
23	9.89998
24	60
25	10.605
26	19.8
27	60
28	-4.37496
29	19.8
30	60
31	-4.37496
32	9.90001
33	60
34	-4.37496
35	9.90001
36	30
37	-4.37496
38	19.8
39	30
40	-4.37496
41	19.8
42	60
43	-4.37496
44	19.8
45	30
46	10.605
47	19.8
48	30

1	122.402
2	142.502
3	132.302
4	142.502
5	132.302
6	172.502
7	122.402
8	172.502
9	122.402
10	142.502
11	132.993
12	153.093
13	132.993
14	183.093
15	122.402
16	172.502
17	132.302
18	172.502
19	142.893
20	183.093
21	132.993
22	183.093
23	132.993
24	153.093
25	142.893
26	153.093
27	142.893
28	183.093
29	142.893
30	153.093
31	132.302
32	142.502

1	11.105
2	10.23
3	31
4	11.105
5	20.13
6	31
7	11.105
8	20.13
9	61
10	11.105
11	10.23
12	61
13	11.105
14	10.23
15	31
16	-3.87496
17	10.23
18	31
19	-3.87496
20	10.23
21	61
22	11.105
23	10.23
24	61
25	11.105
26	20.13
27	61
28	-3.87495
29	20.13
30	61
31	-3.87496
32	10.23
33	61
34	-3.87496
35	10.23
36	-3.87496
37	10.23
38	31
39	-3.87495
40	20.13
41	61
42	-3.87495
43	20.13
44	31
45	11.105
46	20.13
47	31
48	31

1	122.379
2	143.149
3	132.279
4	143.149
5	132.279
6	173.149
7	122.379
8	173.149
9	622.379
10	143.149
11	132.97
12	133.74
13	132.97
14	133.74
15	122.379
16	173.149
17	132.279
18	173.149
19	142.67
20	133.74
21	132.97
22	133.74
23	132.97
24	133.74
25	142.67
26	133.74
27	142.67
28	133.74
29	142.67
30	133.74
31	132.279
32	143.149

1	11.105
2	-10.23
3	31
4	11.105
5	-20.13
6	31
7	11.105
8	-20.13
9	61
10	11.105
11	-10.23
12	61
13	11.105
14	-10.23
15	31
16	-3.675
17	-10.23
18	31
19	-3.675
20	-10.23
21	61
22	11.105
23	-10.23
24	61
25	11.105
26	-20.13
27	61
28	-3.675
29	-20.13
30	61
31	-3.675
32	-10.23
33	61
34	-3.675
35	-10.23
36	31
37	-3.675
38	-20.13
39	31
40	-3.675
41	-20.13
42	61
43	-3.675
44	-20.13
45	31
46	11.105
47	-20.13
48	31

1	161.919
2	203.149
3	152.019
4	203.149
5	152.019
6	233.149
7	161.919
8	233.149
9	161.919
10	203.149
11	172.51
12	213.74
13	172.51
14	243.74
15	161.919
16	233.149
17	152.019
18	233.149
19	162.61
20	243.74
21	172.51
22	243.74
23	172.51
24	213.74
25	162.61
26	213.74
27	162.61
28	243.74
29	162.61
30	213.74
31	152.019
32	203.149



1	-11.105
2	10.23
3	31
4	-11.105
5	20.13
6	31
7	-11.105
8	20.13
9	61
10	-11.105
11	10.23
12	61
13	-11.105
14	10.23
15	31
16	3.875
17	10.23
18	31
19	3.875
20	10.23
21	61
22	-11.105
23	10.23
24	61
25	-11.105
26	20.13
27	61
28	3.875
29	20.13
30	61
31	3.875
32	10.23
33	61
34	3.875
35	10.23
36	31
37	3.875
38	20.13
39	31
40	3.875
41	20.13
42	61
43	3.875
44	20.13
45	31
46	-11.105
47	20.13
48	31

1	196.081
2	218.851
3	207.981
4	216.851
5	207.981
6	248.851
7	198.081
8	246.851
9	198.081
10	218.851
11	187.49
12	208.26
13	187.49
14	238.26
15	198.081
16	248.851
17	207.981
18	248.851
19	197.39
20	238.26
21	187.49
22	238.26
23	187.49
24	208.26
25	197.39
26	208.26
27	197.39
28	238.26
29	197.39
30	208.26
31	207.981
32	218.851

1	11.105
2	-10.23
3	31
4	11.105
5	-20.13
6	31
7	11.105
8	-20.13
9	61
10	11.105
11	-10.23
12	61
13	11.105
14	-10.23
15	31
16	-3.875
17	-10.23
18	31
19	-3.875
20	-10.23
21	61
22	11.105
23	-10.23
24	61
25	11.105
26	-20.13
27	61
28	-3.875
29	-20.13
30	61
31	-3.875
32	-10.23
33	61
34	-3.875
35	-10.23
36	31
37	-3.875
38	-20.13
39	31
40	-3.875
41	-20.13
42	61
43	-3.875
44	-20.13
45	31
46	11.105
47	-20.13
48	31

1	161.919
2	203.149
3	152.019
4	203.149
5	152.019
6	233.149
7	161.919
8	233.149
9	161.919
10	203.149
11	172.51
12	213.74
13	172.51
14	243.74
15	161.919
16	233.149
17	152.019
18	233.149
19	162.61
20	243.74
21	172.51
22	243.74
23	172.51
24	213.74
25	162.61
26	213.74
27	162.61
28	243.74
29	162.61
30	213.74
31	152.019
32	203.149

1	81.4746
2	47.0394
3	50.3894
4	81.4746
5	102.153
6	89.3602
7	129.204
8	74.0962
9	128.332
10	129.204
11	19.4827
12	89.3607
13	81.4746
14	47.0394
15	50.3894
16	33.8086
17	19.3194
18	89.3088
19	81.5383
20	-8.03732
21	128.28
22	129.204
23	19.4827
24	89.3607
25	129.204
26	74.0962
27	128.332
28	81.5383
29	47.0762
30	167.251
31	81.5383
32	-8.03732
33	128.28
34	33.8086
35	19.3194
36	89.3088
37	33.8086
38	74.633
39	128.28
40	81.5383
41	47.0762
42	167.251
43	33.8086
44	74.633
45	128.28
46	81.4746
47	102.153
48	89.3602

1	117.437
2	120.787
3	172.55
4	159.758
5	111.249
6	164.984
7	56.1352
8	126.013
9	117.437
10	120.787
11	123.617
12	193.406
13	62.3151
14	198.633
15	56.1352
16	126.013
17	111.249
18	164.984
19	117.429
20	237.603
21	62.3151
22	198.633
23	123.617
24	193.406
25	178.73
26	232.377
27	117.429
28	237.603
29	178.73
30	232.377
31	172.55
32	159.758

1	44.6046
2	30.5506
3	41.0874
4	44.6046
5	70.7607
6	57.1609
7	60.9329
8	56.6201
9	94.5732
10	60.9329
11	16.2102
12	79.2997
13	44.6046
14	30.5506
15	41.0874
16	4.66271
17	24.5856
18	57.1405
19	20.9908
20	10.4451
21	94.5528
22	60.9329
23	16.2102
24	79.2997
25	60.9329
26	56.6201
27	94.5732
28	20.9908
29	50.855
30	109.626
31	20.9908
32	10.4451
33	94.5528
34	4.66271
35	24.5856
36	57.1405
37	4.66271
38	64.9955
39	72.414
40	20.9908
41	50.855
42	109.626
43	4.66271
44	64.9955
45	72.414
46	44.6046
47	70.7607
48	57.1609

1	126.815
2	138.352
3	167.225
4	153.625
5	141.341
6	179.494
7	101.131
8	164.22
9	126.815
10	138.352
11	149.289
12	181.844
13	123.605
14	207.712
15	101.131
16	164.22
17	141.341
18	179.494
19	164.015
20	222.986
21	123.605
22	207.712
23	149.289
24	181.844
25	189.699
26	197.117
27	164.015
28	222.986
29	189.699
30	197.117
31	167.225
32	153.625



1	136.676
2	115.891
3	136.676
4	104.172
5	132.478
6	115.702
7	132.478
8	121.75
9	136.676
10	115.891
11	124.973
12	115.891
13	126.438
14	121.75
15	132.478
16	121.75
17	132.478
18	115.702
19	126.438
20	115.702
21	126.438
22	121.75
23	124.973
24	115.891
25	124.973
26	104.172
27	126.438
28	115.702
29	124.973
30	104.172
31	136.676
32	104.172

1	22.21
2	31
3	31
4	22.21
5	61
6	31
7	22.21
8	61
9	61
10	22.21
11	31
12	61
13	22.21
14	31
15	31
16	-7.75
17	31
18	31
19	-7.75
20	31
21	61
22	22.21
23	31
24	61
25	22.21
26	61
27	61
28	-7.75
29	61
30	61
31	-7.75
32	31
33	61
34	-7.75
35	31
36	31
37	-7.75
38	61
39	31
40	-7.75
41	61
42	61
43	-7.75
44	61
45	31
46	22.21
47	61
48	31

DIRECTORY OF IFB:PIRAFE

NAME	.EXT	BLKS	LENGTH	ATTR	NAME	.EXT	BLKS	LENGTH	ATTR
IBIS	.DIR	26	3200	IF	IBIS	.MAP	3	256	IF
IBIS	.TO	24	2944	IF	IBIS	.LAB	2	128	IF
IBIS	.BIN	94	11740	SIF	IBIS	.CLI	22	2407	SIF
COPY		69	6469		CREDIT		156	19470	
DIR		55	6215		DELETE		39	4824	
TRI		10	1145		MENU		53	6569	
GRAPI1.1		3	204		GRAPI2.1		3	204	
GRAPI2.2		3	204		GRAPI3.1		3	170	
GRAPI3.2		5	432		GRAPI5.1		3	140	
GRAPI5.2		3	204		GRAPI5.3		3	214	
GRAPI6.1		3	204		GRAPI7.1		3	203	
GRAPI9.11		3	204		GRAPI9.12		3	155	
GRAPI9.13		3	140		GRAPI9.14		3	170	
GRAPI9.15		3	204		GRAPI9.16		3	224	
GRAPI9.17		3	214		GRAPI9.18		3	219	
GRAPI9.19		3	209		GRAPI9.21		3	209	
GRAPI9.22		3	205		GRAPI9.23		3	175	
GRAPI9.24		3	214		GRAPI9.25		3	214	
GRAPI9.26		3	209		GRAPI9.27		3	145	
GRAPI9.28		3	170		GRAPI9.29		3	204	
GRAPI9.31		3	224		GRAPI9.32		3	229	
GRAPI9.33		3	214		GRAPI9.34		3	160	
GRAPI9.35		3	140		GRAPI9.36		3	209	
GRAPI9.37		3	155		GRAPI9.38		3	209	
GRAPI9.39		3	204		GRE10 .11		3	150	
GRE10 .12		3	150		GRE10 .13		3	145	
GRE10 .14		3	140		GRE10 .15		3	150	
GRE10 .16		3	150		GRE10 .17		3	170	
GRE10 .18		3	170		GRE10 .19		3	170	
GRE10 .21		3	150		GRE10 .22		3	135	
GRE10 .23		3	151		GRE10 .24		3	204	
GRE10 .25		3	170		GRE10 .26		3	135	
GRE10 .27		3	135		GRE10 .28		3	170	
GRE10 .29		3	170		GRE10 .31		3	150	
GRE10 .32		3	150		GRE10 .33		3	150	
GRE10 .34		3	150		GRE10 .35		3	145	
GRE10 .36		3	150		GRE10 .37		3	160	
GRAPI4.33		3	214		GRAPI4.34		3	160	
GRAPI4.35		3	140		GRAPI4.36		3	209	
GRAPI4.37		3	209		GRAPI4.38		3	209	
GRAPI4.39		3	204		GRAPI8.1		3	170	
GRAPI2.1		3	144		GRAPI3.1		3	150	
GRAPI4.1		3	170		GRAPI5.1		3	150	
GRAPI5.2		5	432		GRAPI6.1		3	150	
GRAPI6.2		5	402		GRA144.1		3	165	
GRA144.2		3	130		GRA144.3		4	298	
GRA144.4		2	121		GRAPI4.11		3	214	
GRAPI4.12		3	160		GRAPI4.13		3	140	
GRAPI4.14		3	160		GRAPI4.15		3	204	
GRAPI4.16		3	219		GRAPI4.17		3	214	
GRAPI4.18		3	224		GRAPI4.19		3	204	
GRAPI4.21		3	209		GRAPI4.22		3	205	
GRAPI4.23		3	175		GRAPI4.24		3	214	
GRAPI4.25		3	214		GRAPI4.26		3	209	
GRAPI4.27		3	145		GRAPI4.28		3	170	
GRAPI4.29		3	204		GRAPI4.31		3	224	

GRAPI4.32	3	229
GRE10 .39	3	170
GRE11 .12	3	150
GRE11 .14	3	140
GRE11 .16	3	150
GRE11 .18	3	170
GRE11 .21	3	150
GRE11 .23	3	150
GRE11 .25	3	170
GRE11 .27	3	135
GRE11 .29	3	170
GRE11 .31	3	149
GRE11 .34	3	150
GRE11 .36	3	150
GRE11 .38	3	170

GRE10 .38	3	170
GRE11 .11	3	150
GRE11 .13	3	145
GRE11 .15	3	150
GRE11 .17	3	170
GRE11 .19	3	170
GRE11 .22	3	135
GRE11 .24	3	204
GRE11 .26	3	135
GRE11 .28	3	170
GRE11 .32	3	150
GRE11 .33	3	150
GRE11 .35	3	145
GRE11 .37	3	160
GRE11 .39	3	170

947

947/2002 BLOCKS USED

DIRECTORY OF IPO:PIRA

NAME	.EXT	BLKS	LENGTH	ATTR	NAME	.EXT	BLKS	LENGTH	ATTR
ISIS	.DIR	26	5200	IF	ISIS	.MAP	5	256	
ISIS	.TO	24	2944	IF	ISIS	.LAB	2	128	
ISIS	.BIN	94	11740	SIF	ISIS	.CLI	20	2407	
DIR		55	6815		COPY		69	8489	
CREDIT		156	19470		DELETE		39	4824	
TRI		10	1145		MENU		53	6589	
GRI144.1		3	134		GRI144.2		2	82	
GRI144.3		3	230		GRI144.4		2	81	
GRAFI1.1		3	159		GRAFI2.2		3	160	
GRAFI2.1		3	160		GRAFI3.1		3	135	
GRAFI9.11		3	160		GRAFI9.12		3	150	
GRAFI9.13		3	135		GRAFI9.14		3	135	
GRAFI9.15		3	145		GRAFI9.16		3	150	
GRAFI9.17		3	160		GRAFI9.18		3	160	
GRAFI9.19		3	160		GRAFI9.21		3	160	
GRAFI9.22		3	155		GRAFI9.23		3	155	
GRAFI9.24		3	150		GRAFI9.25		3	160	
GRAFI9.26		3	155		GRAFI9.27		3	135	
GRAFI9.28		3	150		GRAFI9.29		3	160	
GRAFI9.31		3	160		GRAFI9.32		3	150	
GRAFI9.33		3	160		GRAFI9.34		3	135	
GRAFI9.35		3	135		GRAI10.11		2	125	
GRAFI9.36		3	134		GRAFI9.37		3	160	
GRAFI9.38		3	155		GRAFI9.39		3	160	
GRAI10.12		2	115		GRAI10.13		2	110	
GRAI10.14		2	106		GRAI10.15		2	115	
GRAI10.16		2	116		GRAI10.17		3	135	
GRAI10.18		3	135		GRAI10.19		3	135	
GRAI10.21		2	115		GRAI10.22		2	110	
GRAI10.23		2	110		GRAI10.24		3	135	
GRAI10.25		3	135		GRAI10.26		3	131	
GRAI10.27		2	111		GRAI10.28		3	136	
GRAI10.29		3	135		GRAI10.31		2	125	
GRAI10.32		2	115		GRAI10.33		2	115	
GRAI10.34		2	115		GRAI10.35		2	109	
GRAI10.36		2	116		GRAI10.37		2	116	
GRAI10.38		3	135		GRAI10.39		3	135	
GRAI11.11		2	125		GRAI11.12		2	115	
GRAI11.13		2	110		GRAI11.14		2	106	
GRAI11.15		2	115		GRAI11.16		2	116	
GRAI11.17		3	135		GRAI11.18		3	135	
GRAI11.19		3	135		GRAI11.21		2	115	
GRAI11.22		2	110		GRAI11.23		2	110	
GRAI11.24		3	135		GRAI11.25		3	135	
GRAI11.26		3	131		GRAI11.27		2	110	
GRAI11.28		3	135		GRAI11.29		3	135	
GRAI11.31		2	125		GRAI11.32		2	115	
GRAI11.33		2	115		GRAI11.34		2	115	
GRAI11.35		2	110		GRAI11.36		2	115	
GRAI11.37		2	115		GRAI11.38		3	135	
GRAI11.39		3	135		GRAFI4.11		3	160	
GRAFI4.12		3	150		GRAFI4.13		3	135	
GRAFI4.14		3	135		GRAFI4.15		3	146	
GRAFI4.16		3	150		GRAFI4.17		3	160	
GRAFI4.18		3	160		GRAFI4.19		3	160	
GRAFI4.21		3	160		GRAFI4.22		3	155	
GRAFI4.23		3	157		GRAFI4.34		3	135	
GRAFI4.25		3	161		GRAFI4.26		3	155	

GRAFI4.27	3	136	GRAFI4.28	3	150
GRAFI4.29	3	160	GRAFI4.31	3	160
GRAFI4.32	3	150	GRAFI4.33	3	159
GRAFI4.35	3	135	GRAFI4.36	3	135
GRAFI4.37	3	160	GRAFI4.38	3	155
GRAFI4.39	3	160	GRAFI3.2	4	316
GRAFI6.1	3	160	GRAFI7.1	3	169
GRAFI5.1	3	135	GRAFI5.2	3	145
GRAFI3.3	3	160	GRAFI8.1	3	135
GRAI12.1	2	110	GRAI13.1	2	114
GRAI14.1	3	135	GRAI15.1	2	110
GRAI16.1	2	114	GRAI16.2	4	316
GRAI15.2	4	317	GRAFI4.24	3	150

904

904/2002 BLOCKS USED

DIRECTORY OF :F0:POLIPE

NAME	.EXT	BLKS	LENGTH	ATTR	NAME	.EXT	BLKS	LENGTH	ATTR
ISIS	.DIR	26	3200	IF	ISIS	.MAP	3	256	IF
ISIS	.TO	24	2944	IF	ISIS	.LAB	2	128	IF
ISIS	.BIN	94	11740	SIF	ISIS	.CLI	20	2407	SIF
COPY		69	8489		DIR		55	6815	
DELETE		39	4824		CREDIT		156	19470	
FRI		10	1145		MENU		53	6589	
GAR144.1		3	204		GRAFT4.35		3	214	
GRAFT4.36		4	276		GRAFT4.37		4	264	
GAR144.4		3	165		GRAFT1.1		3	229	
GRAFT2.1		3	214		GRAFT2.2		3	214	
GRAFT3.1		3	214		GRAFT3.2		5	506	
GRAFT5.1		3	214		GRAFT5.2		4	265	
GRAFT5.3		4	264		GRAFT6.1		3	224	
GRAFT7.1		3	224		GRAFT4.11		3	219	
GRAFT4.12		3	224		GRAFT4.13		3	214	
GRAFT4.16		4	298		GRAFT4.17		4	264	
GRAFT4.18		3	249		GRAFT4.19		3	214	
GRAFT4.21		3	234		GRAFT4.22		3	230	
GRAFT4.23		3	249		GRAFT4.24		3	244	
GRAFT4.25		4	264		GRAFT4.26		3	239	
GRAFT4.27		3	244		GRAFT4.28		3	213	
GRAFT4.29		3	214		GRAFT4.31		3	254	
GRAFT4.32		4	276		GRAFT4.33		4	264	
GRAFT4.34		3	234		GRAT10.11		3	214	
GRAT10.12		3	214		GRAT10.13		3	214	
GRAT10.14		3	214		GRAT10.15		3	214	
GRAT10.16		3	214		GRAT11.11		3	214	
GRAT10.17		3	213		GRAT10.18		3	214	
GRAT10.19		3	214		GRAT10.21		3	214	
GRAT10.22		3	165		GRAT10.23		3	214	
GRAT10.24		3	214		GRAT10.25		3	214	
GRAT10.26		3	165		GRAT10.27		3	214	
GRAT10.28		3	214		GRAT10.29		3	214	
GRAT10.31		3	214		GRAT10.32		3	214	
GRAT10.33		3	214		GRAT10.34		3	214	
GRAT10.35		3	214		GRAT10.36		3	214	
GRAT10.37		3	214		GRAT10.38		3	214	
GRAT10.39		3	214		GRAT11.12		3	214	
GRAT11.13		3	214		GRAT11.14		3	214	
GRAT11.15		3	214		GRAT11.16		3	214	
GRAT11.17		3	214		GRAT11.18		3	214	
GRAT11.19		3	214		GRAT11.21		3	214	
GRAT11.22		3	165		GRAT11.23		3	214	
GRAT11.24		3	214		GRAT11.25		3	214	
GRAT11.26		3	165		GRAT11.27		3	214	
GRAT11.28		3	214		GRAT11.29		3	214	
GRAT11.31		3	214		GRAT11.32		3	214	
GRAT11.33		3	214		GRAT11.34		3	214	
GRAT11.35		3	214		GRAT11.36		3	214	
GRAT11.37		3	214		GRAT11.38		3	214	
GRAT11.39		3	214		GRAFT8.1		3	215	
GRAT12.1		3	214		GRAT13.1		3	214	
GRAT14.1		3	223		GRAT15.1		3	223	
GRAT16.1		3	213		GRAT15.2		5	506	
GRAT16.2		5	486		GRAFT4.14		4	264	
GRAFT4.15		4	264		GAR144.2		3	165	
GAR144.3		4	318		GRAFT9.11		3	224	
GRAFT9.12		3	234		GRAFT9.13		3	214	

GRAFT9.14	4	260	GRAFT9.15	4	264
GRAFT9.16	4	269	GRAFT9.17	4	264
GRAFT9.18	3	254	GRAFT9.19	3	224
GRAFT9.21	3	234	GRAFT9.22	3	230
GRAFT9.23	3	249	GRAFT9.24	3	249
GRAFT9.25	4	264	GRAFT9.26	3	239
GRAFT9.27	3	244	GRAFT9.28	3	214
GRAFT9.29	3	214	GRAFT9.31	3	254
GRAFT9.32	4	276	GRAFT9.33	4	264
GRAFT9.34	3	234	GRAFT9.35	3	214
GRAFT9.36	4	276	GRAFT9.37	4	264
GRAFT9.38	3	234	GRAFT9.39	3	214
GRAFT4.38	3	234	GRAFT4.79	3	214

968

968/2002 BLOCKS USED



DIRECTORY OF :FQ:EXA

NAME	.EXT	BLKS	LENGTH	ATTR	NAME	.EXT	BLKS	LENGTH	ATTR
ISIS	.DIR	26	3200	IF	ISIS	.MAP	3	256	IF
ISIS	.T0	24	2944	IF	ISIS	.LAB	2	128	IF
ISIS	.BIN	94	11740	SIF	ISIS	.CLI	20	2407	SIF
DIR		53	6813		DELETE		39	4824	
CREDIT		156	19470		COPY		69	8489	
MENU		53	6589		TRI		10	1145	
GRAPH1.3		3	213		GRAPH1.41		3	214	
GRAPH1.42		3	214		GRAPH1.43		3	214	
GRAPH1.51		3	155		GRAPH2.51		3	155	
GRAPH1.61		3	214		GRAPH1.62		3	214	
GRAPH1.63		3	214		GRAPH1.64		3	214	
			214		GRAPH1.6c		3	214	
								214	
GRAPH2.01		3	214		GRAPH1.67		3	214	
GRAPH1.621		3	214		GRAPH1.622		3	214	
GRAPH1.491		4	298		GRAPH1.492		4	296	
GRAPH1.493		3	160		GRAPH1.494		2	126	
GRAPH1.661		3	214		GRAPH1.662		3	170	
GRAPH1.663		3	214		GRAPH1.664		3	182	
GRAPH1.665		3	214		GRAPH1.666		3	170	
GRAPH1.667		3	214		GRAPH1.668		3	182	
GRAPH1.669		3	214		GRAPH1.670		3	170	
GRAPH1.671		3	214		GRAPH1.71		3	214	
GRAPH1.72		3	214		GRAPH1.73		3	214	
GRAPH1.49		4	310		GRAPH1.91		3	213	
GRAPH1.672		3	183		GRAPH1.673		3	214	
GRAPH1.674		3	176		PRUEBA		5	465	
GRAPH1.53		5	391		GRAPH2.1		3	155	
GRAPH2.21		3	214		GRAPH1.675		3	214	
GRAPH1.676		3	183		GRAPH1.677		3	214	
GRAPH1.678		3	175		GRAPH1.679		3	214	
GRAPH2.52		3	159		GRAPH2.53		3	154	
GRAPH6.11		3	155		GRAPH6.12		5	392	
GRAPH6.51		3	155		GRAPH6.52		4	318	
GRU14.31		3	155		GRU14.32		3	155	
GRU14.33		3	155		GRU14.34		3	155	
GRU14.35		3	155		GRU14.36		3	155	
GRU14.37		3	155		GRU14.38		3	155	
GRU14.39		3	155		GRU13.21		3	155	
GRU12.11		3	214		GRU12.12		3	215	
GRU12.13		3	214		GRU12.14		3	215	
GRU12.15		3	214		GRU12.16		3	214	
GRU12.17		3	214		GRU12.18		3	214	
GRU12.19		3	214		GRU12.21		3	214	
GRU12.22		3	170		GRU12.23		3	214	
GRU12.24		3	182		GRU12.25		3	214	
GRU12.26		3	170		GRU12.27		3	214	
GRU12.28		3	182		GRU12.29		3	214	
GRU12.31		3	214		GRU12.32		3	182	
GRU12.34		3	175		GRU12.35		3	214	
GRU12.36		3	182		GRU12.37		3	214	
GRU12.38		3	174		GRU12.39		3	213	
GRU13.11		3	155		GRU13.12		3	155	
GRU13.13		3	155		GRU13.14		3	155	
GRU13.15		3	155		GRU13.16		3	155	
GRU13.17		3	155		GRU13.18		3	155	
GRU13.19		3	155		GRU13.22		2	125	
GRU13.23		3	155		GRU13.24		3	155	
GRU13.25		3	155		GRU13.26		2	125	

GRU13 .27	3	155	GRU13 .28	3	155
GRU13 .29	3	155	GRU13 .31	3	155
GRU13 .32	3	155	GRU13 .33	3	156
GRU13 .34	3	155	GRU13 .35	3	155
GRU13 .36	3	155	GRU13 .37	3	155
GRU13 .38	3	155	GRU13 .39	3	155
GRU14 .11	3	155	GRU14 .12	3	155
GRU14 .13	3	155	GRU14 .14	3	155
GRU14 .15	3	155	GRU14 .16	3	155
GRU14 .17	3	155	GRU14 .18	3	155
GRU14 .19	3	155	GRU14 .21	3	155
GRU14 .22	2	125	GRU14 .23	3	155
GRU14 .24	3	155	GRU14 .25	3	155
GRU14 .26	2	125	GRU14 .27	3	155
GRU14 .28	3	155	GRU14 .29	3	155

964

964/2002 BLOCKS USED

DIRECTORY OF :F0:POLI

NAME	.EXT	BLKS	LENGTH	ATTR	NAME	.EXT	BLKS	LENGTH	ATTR
IBIS	.DIR	26	3200	IF	IBIS	.MAP	3	256	IF
IBIS	.T0	24	2944	IF	IBIS	.LAB	2	128	IF
IBIS	.BIN	94	11740	SIF	IBIS	.CLI	20	2407	SIF
DIR		55	6815		COPY		69	9489	
CREDIT		156	19470		DELETE		39	4834	
TRI		10	1145		MENU		53	6589	
GRAF1	.1	3	181		GRF144.2		2	94	
GRF144.3		3	225		GRF144.4		2	93	
GRAF3	.1	4	370		GRAF3	.2	4	338	
GRAF4	.11	3	155		GRAF4	.12	3	155	
GRAF4	.13	3	160		GRAF4	.14	3	140	
GRAF4	.15	3	150		GRAF4	.16	3	140	
GRAF4	.17	3	150		GRAF4	.18	3	140	
GRAF4	.19	3	160		GRAF5	.1	3	182	
GRAF5	.2	3	182		GRAF5	.3	3	182	
GRAF6	.1	3	182		GRAF7	.1	3	181	
GRAF4	.21	3	160		GRAF4	.22	3	145	
GRAF4	.23	3	140		GRAF4	.24	3	130	
GRAF4	.25	3	150		GRAF4	.26	3	135	
GRAF4	.27	3	140		GRAF4	.28	3	135	
GRAF4	.29	3	159		GRAF4	.31	3	135	
GRAF4	.32	3	140		GRAF4	.33	3	150	
GRAF4	.34	2	125		GRAF4	.35	3	160	
GRAF4	.36	3	140		GRAF4	.37	3	150	
GRAF4	.38	2	125		GRAF4	.39	3	160	
GRAF2	.1	3	182		GRAF2	.2	3	182	
GRAF8	.1	2	120		GRAF12.1		2	120	
GRAF13.1		2	120		GRAF14.1		2	119	
GRAF15.1		2	120		GRAF16.1		3	130	
GRAF15.2		4	338		GRAF16.2		4	338	
GRF144.1		3	129		GRAF9	.11	3	170	
GRAF9	.12	3	170		GRAF9	.13	3	182	
GRAF9	.14	3	160		GRAF9	.15	3	182	
GRAF9	.16	3	171		GRAF9	.17	3	181	
GRAF9	.18	3	160		GRAF9	.19	3	182	
GRAF9	.21	3	182		GRAF9	.22	3	165	
GRAF9	.23	3	161		GRAF9	.24	3	150	
GRAF9	.25	3	182		GRAF9	.26	3	165	
GRAF9	.27	3	160		GRAF9	.28	3	150	
GRAF9	.29	3	182		GRAF9	.31	3	150	
GRAF9	.32	3	160		GRAF9	.33	3	182	
GRAF9	.34	3	139		GRAF9	.35	3	182	
GRO10	.27	2	120		GRAF9	.36	3	159	
GRAF9	.37	3	182		GRAF9	.38	3	140	
GRAF9	.39	3	182		GRO10	.11	2	120	
GRO10	.12	2	120		GRO10	.13	2	120	
GRO10	.14	2	120		GRO10	.15	2	120	
GRO10	.16	2	120		GRO10	.17	2	120	
GRO10	.19	2	120		GRO10	.21	3	130	
GRO10	.22	2	115		GRO10	.23	2	120	
GRO10	.24	2	120		GRO10	.26	2	119	
GRO10	.25	2	119		GRO10	.28	2	120	
GRO10	.29	2	120		GRO10	.31	2	120	
GRO10	.32	2	120		GRO10	.33	2	120	
GRO10	.34	2	120		GRO10	.35	2	120	
GRO10	.36	2	120		GRO10	.37	2	120	
GRO10	.38	2	120		GRO10	.39	2	120	
GRO11	.11	2	120		GRO11	.12	2	120	

GRO11 .13	2	121	GRO11 .14	2	120
GRO11 .15	2	120	GRO11 .16	2	120
GRO11 .17	2	121	GRO11 .18	2	121
GRO11 .19	2	121	GRO11 .21	3	131
GRO11 .22	2	115	GRO11 .23	2	120
GRO11 .24	2	120	GRO11 .25	2	120
GRO11 .26	2	115	GRO11 .27	2	120
GRO11 .28	2	121	GRO11 .29	2	121
GRO11 .31	2	120	GRO11 .32	2	120
GRO11 .33	2	120	GRO11 .34	2	120
GRO11 .35	2	121	GRO11 .36	2	120
GRO11 .37	2	120	GRO11 .38	2	120
GRO11 .39	2	121	GRO10 .18	2	120

884

884/2802 BLOCKS USED

1 U0000U0000U0F0FL0F0FL1E0FL2D0FL2D1E1L2D2DL1E2DL0F2DL0F1E1L0F0FL1919L2424  
L2433L2442  
2 L1937L0F2DL1E2DL2D2DL3737L4242L3342L2442L2433L2424L3324L4224L4233L4242  
L2433L4224  
3 L3719L2D0FX

1 U0000U0000U0F0FL0F0FL1E0FL2D0FL2D1E1L2D2DL1E2DL0F2DL0F1E1L0F0FL1919L2424  
L2433L2442  
2 L1937L0F2DL1E2DL2D2DL3737L4242L3342L2442L2433L2424L3324L4224L4233L4242  
L4233L4224  
3 L3719L2D0FX

1 U0000U0000U0216L0216L0C16L0C05L0C34L0234L0225L0216L0C21L0C30L0C3FL0234  
L0C34L163F  
2 L0C3FL0C30L0C21L1621L1630L163FL1630L1621L0C16X

1 U0000U0000U06695L8695L9499LA39DL9FA0L988AL8CB6L7EB2L82A4L8695L90A0L9BAA  
L9789L9307  
2 L89BDL7EB2L8CB6L988ALA6C5LB0CFLA2CBL93C7L9789L9BAAAL9AEL88B2LB4C1LB0CF  
LB4C1L88B2  
3 LADABLA39DX

1 U0000U0000UC387LC387LD287LE187LE178LE169LD269LC369LC378LC387LCD91LD89C  
LD88DL87E  
2 LCD73LC369LD269LE169LEB73LF67ELE77ELD87ELD88DL89CLE79CLF69CLF68DLF67E  
LF68DLF69C  
3 LEB91LE187X

1 U0000U0000U7F87L7F87L8D8CL9891L929AL89A3L789EL6D99L7690L7F87L8696L8EA5  
L89AEL7CB7  
2 L75A6L6D99L789EL89A3L91B2L99C1L8ABCL7CB7L85AEL8EA5L9CAALAAFLA1B8L99C1  
LA1B8LAAAF  
3 LACA0L9891X

1 U0000U0000U7B7CL7B7CL8785L938EL868FL7890L6C87L607FL6D7DL7B7CL7C8CL7E9D  
L709EL629F  
2 L618FL607FL6C87L7890L79A0L7BB0L6FA8L629FL709EL7E9DL8AA5L96AEL8BAFL7BB0  
L38AFL96AE  
3 L959EL938EX



0584LD-99  
1 004210000 08870L58700  
2 004210000 08870L58700

1 00000000000715AL715AL7882L7297L6940L609CL6893L715AL7999L70A2L67ABL609C  
L6340L71AF  
2 L67ABL70A2L7999L629DL7AA6L71AF1.7AA6L629DL788EX

LISTADO DEL PROGRAMA TRIDIMENSIONAL.MENU



PL/M-80 COMPILER

ISIS-II PL/M-80 V3.1 COMPILATION OF MODULE TRIDIMENSIONAL  
OBJECT MODULE PLACED IN TRI.OBJ  
COMPILER INVOKED BY: PLM80 TRI.SRC WORKFILES (:F0:,:F0:)

\$ PAGELength(41)

PL/M-80 COMPILER

```

$ EJECT
/*****
      /*PROGRAMA DE FUNCIONAMIENTO DEL PLOTTER*/
*****/
/*****
TRIDIMENSIONAL:DO;
1      1      IOSET:  PROCEDURE (CONFIG) EXTERNAL;
2      2      DECLARE CONFIG BYTE;
3      2      END IOSET;
4      2      PO:     PROCEDURE (CRTR) EXTERNAL;
5      2      DECLARE CRTR BYTE;
6      2      END PO;
7      2      EXIT:   PROCEDURE EXTERNAL;
8      2      END EXIT;
9      2      CO:     PROCEDURE (CHAR) EXTERNAL;
10     2      DECLARE CHAR BYTE;
11     2      END CO;
12     2      ERROR:  PROCEDURE (ERRNUM) EXTERNAL;
13     2      DECLARE ERRNUM ADDRESS;
14     2      END;
15     2      CI:     PROCEDURE BYTE EXTERNAL;
16     2      END CI;
17     2      WRITE:  PROCEDURE (AFTN,BUFFER,COUNT,STATUS) EXTERNAL;
18     2      DECLARE (AFTN,BUFFER,COUNT,STATUS) ADDRESS;
19     2      END WRITE;
20     2      OPEN:   PROCEDURE (AFTN,FILE,ACCESS,MODE,STATUS) EXTERNAL;
21     2      DECLARE (AFTN,FILE,ACCESS,MODE,STATUS) ADDRESS;
22     2      END OPEN;
23     2      READ:   PROCEDURE (AFTN,BUFFER,COUNT,ACTUAL,STATUS) EXTERNAL;
24     2      DECLARE (AFTN,BUFFER,COUNT,ACTUAL,STATUS) ADDRESS;
25     2      END READ;
26     2      HEXASC: PROCEDURE (Z) BYTE; /*CONVERSION ASCII-HEXADECIMAL*/
27     2      DECLARE Z BYTE;
28     2      FLAG=0;
29     2      IF (Z>='0') AND (Z<='9') THEN GOTO L3;
30     2      IF (Z>='A') AND (Z<='F') THEN GOTO L4;
31     2      IF Z=0DH THEN DO;
32     2      FLAG=5;
33     2      GOTO L5;
34     2
35     2
36     3
37     3
38     3      END;
39     2      IF Z='X' THEN DO;
40     2
41     3      FLAG=2;
42     3      GOTO L5;
43     3      END;
44     2      IF Z='L' THEN DO;
45     2
46     3      FLAG=3;
47     3      GOTO L5;

```

PL/M-80 COMPILER

```

48 3      END;
49 2      IF Z='U' THEN DO;
51 3      FLAG=4;
52 3      GOTO L5;
53 3      END;
54 2      FLAG=1;
55 2      GOTO L5;
56 2      L5:  Z=Z AND (0FH);
57 2      GOTO L5;
58 2      Z=Z AND (0FH);
59 2      Z=Z+09H;
60 2      L5:  RETURN Z;
61 2      END HEXASC;
62 1      ACCFIL: PROCEDURE (H) BYTE; /*COMPROBACION DE FINAL DE FICHERO*/
63 2          DECLARE H BYTE;
64 2          IF H>127 THEN DO;
66 3          CALL READ(AFT#IN,.BUFFER,128,.ACTUAL,.STATUS);
67 3          IF STATUS<>0 THEN GOTO P2;
69 3          H=0;
70 3          END;
71 2          RETURN H;
72 2          END ACCFIL;
73 1      COGEB: PROCEDURE; /*TOMA DE DATOS*/
74 2      AHI:  I=ACCFIL(I);
75 2          A=BUFFER(I);
76 2          A=HEXASC(A);
77 2          I=I+1;
78 2          IF FLAG=5 THEN GOTO AHI;
80 2          IF (FLAG=1) AND((FLAG=1) OR (FLAG=0)) THEN GOTO AHI;
82 2          END COGEB;

```

PL/M-80 COMPILER

```

83 1      DATA#IN: PROCEDURE ; /*ELECCION DEL FICHERO*/
84 2          NUM=0;
85 2          DO WHILE TECLA<>0DH;
86 3          D1:  FILE(NUM)=CI AND 07FH;
87 3              IF FILE(NUM)=07FH THEN DO;
89 4                  CALL CO(08);
90 4                  CALL CO(20H);
91 4                  CALL CO(08);
92 4                  NUM=NUM-1;
93 4                  GOTO D1;
94 4              END;
95 3              CALL CO(FILE(NUM));
96 3              TECLA=FILE(NUM);
97 3              NUM=NUM+1;
98 3              END;
99 2              NUM=NUM-1;
100 2          DO WHILE NUM<=19;
101 3              FILE(NUM)=20H;
102 3              NUM=NUM+1;
103 3              END;
104 2              CALL CO(0DH);
105 2              CALL CO(0AH);
106 2              RETURN;
107 2          END DATA#IN;
108 1      PUNTOS: PROCEDURE;
109 2          FLAG1=1;

```

```

110 2          CALL COGEB;
111 2          FLAG1=0;
112 2          IF FLAG=2 THEN GOTO L13;
114 2          IF FLAG=3 THEN NLAPIZ=00;
116 2          IF FLAG=4 THEN NLAPIZ=01H;
118 2          CALL COGEB;
119 2          X1=A;
120 2          CALL COGEB;
121 2          X2=A;
122 2          X=SHL(X1,4)+X2;
123 2          CALL COGEB;
124 2          Y1=A;
PL/M-B0 COMPILER

125 2          CALL COGEB;
126 2          Y2=A;
127 2          Y=SHL(Y1,4)+Y2;
128 2          END PUNTOS;
129 1          DECLARE (AFT$IN,STATUS,ACTUAL,AFT$CO)ADDRESS;
130 1          DECLARE (A,B,J,I,FLAG,OLAPIZ,NLAPIZ,X,Y,X1,X2,Y1,Y2,FLAG1,NUM) BYTE;
131 1          DECLARE (TECLA) BYTE;
132 1          DECLARE FILE(10) BYTE;
133 1          DECLARE BUFFER(128) BYTE;
/*COMIENZO DEL PROGRAMA*/
134 1          CALL WRITE(0,('NOMBRE DEL FICHERO?',0DH,0AH),21,.STATUS);
135 1          IF STATUS<>0 THEN GOTO P2;
137 1          CALL DATA$IN;
138 1          CALL IOSET(095H);
139 1          LB:  CALL OPEN(.AFT$IN,.FILE,1,0,.STATUS);
140 1          IF STATUS<>0 THEN GOTO P2;
142 1          CALL READ(AFT$IN,.BUFFER,128,.ACTUAL,.STATUS);
143 1          IF STATUS<>0 THEN GOTO P2;
145 1          I=0;OLAPIZ=0FFH;
147 1          ALLI: CALL PUNTOS;
148 1          IF OLAPIZ<>NLAPIZ THEN DO;
150 2          CALL PO(NLAPIZ);
151 2          DO J=1 TO 40;
152 3              CALL TIME(250);
153 3          END;
154 2          END;
155 1          ELSE DO;
156 2          CALL PO(NLAPIZ);
157 2          END;
158 1          OLAPIZ=NLAPIZ;
159 1          CALL PO(X);
160 1          CALL PO(Y);
161 1          DO J=1 TO 15;
162 2              CALL TIME(250);
163 2          END;
164 1          CALL PO(0);
165 1          GOTO ALLI;
166 1          P2:  CALL ERROR(STATUS);
PL/M-B0 COMPILER

167 1          L13:  CALL EXIT;
168 1          END TRIDIMENSIONAL;

```

MODULE INFORMATION:

```
CODE AREA SIZE      = 0343H      835D
VARIABLE AREA SIZE  = 00A4H      1A4D
MAXIMUM STACK SIZE  = 000EH      14D
155 LINES READ
0 PROGRAM ERROR(S)
```

END OF PL/M-80 COMPILATION

ISIS-II PLM-60 V3.1 COMPILATION OF MODULE MENU  
 OBJECT MODULE PLACED IN MENU.OBJ  
 COMPILER INVOKED BY: PLM60 MENU.SRC WORKFILES (:F0:,:F0:)

```

1      MENU:DO:

2      1      CO:          PROCEDURE (CHAR) EXTERNAL;
3      2      DECLARE CHAR BYTE;
4      2      END CO;
5      1      CI:          PROCEDURE BYTE EXTERNAL;
6      2      END CI;
7      1      EXIT:        PROCEDURE EXTERNAL;
8      2      END EXIT;
9      1      WRITE:       PROCEDURE (AFTN,BUFFER,COUNT,STATUS)EXTERNAL;
10     2      DECLARE (AFTN,BUFFER,COUNT,STATUS)ADDRESS;
11     2      END WRITE;
12     1      DECLARE (ENT,P,W,CON)BYTE;
13     1      DECLARE (COUNT,STATUS)ADDRESS;
14     1      DMM:

15     1      CALL WRITE(0,.(1BH,45H,0AH,0AH,0AH),5,.STATUS);
16     1      CALL WRITE(0,.( '          MENU' ),32,.STATUS);
17     1      CALL WRITE(0,.(0DH,0AH, '          1. EXAEDRO' ),32,.STATUS);
18     1      CALL WRITE(0,.(0DH,0AH,0AH,0AH, '          2. PIRAMIDE' ),34,.STATUS);
19     1      CALL WRITE(0,.(0DH,0AH,0AH,0AH, '          3. PRISMA TRUNCADO' ),41,.STATUS);
20     1      CALL WRITE(0,.(0DH,0AH,0AH,0AH, '          4. PIRAMIDE DE BASE EXAGONAL' ),51,.STATUS);
21     1      CALL WRITE(0,.(0DH,0AH,0AH,0AH, '          5. PRISMA EXAGONAL' ),41,.STATUS);

22     1      DATO: ENT=CI AND 07FH;
23     1      IF ENT=31H THEN GOTO DATO;
24     1      IF ENT=35H THEN GOTO DATO;
25     1      IF ENT=39H THEN GOTO DATO;
26     1      IF ENT=3DH THEN GOTO DATO;
27     1      IF ENT=3FH THEN GOTO DATO;
28     1      IF ENT=43H THEN GOTO DATO;
29     1      IF ENT=47H THEN GOTO DATO;
30     1      IF ENT=4BH THEN GOTO DATO;
31     1      IF ENT=4FH THEN GOTO DATO;
32     1      IF ENT=53H THEN GOTO DATO;

35     1      DMM:

36     1      CALL WRITE(0,.(1BH,45H,0AH,0AH,0AH),5,.STATUS);
37     1      CALL WRITE(0,.( '          1.FIGURA BASICA' ),20,.STATUS);
38     1      CALL WRITE(0,.(0DH,0AH,0AH, '          9. TRASLACION - ESTRECHAMIENTO' ),48,.STATUS);
39     1      CALL WRITE(0,.( '          2. TRASLACION' ),21,.STATUS);
40     1      CALL WRITE(0,.( '          A. TRASLACION - ROTACION' ),44,.STATUS);
41     1      CALL WRITE(0,.(0DH,0AH,0AH, '          3. ESTRECHAMIENTO' ),24,.STATUS);
42     1      CALL WRITE(0,.( '          B. ESTRECHAMIENTO - ROTACION' ),45,.STATUS);
43     1      CALL WRITE(0,.(0DH,0AH,0AH, '          4. ROTACION' ),18,.STATUS);
44     1      CALL WRITE(0,.( '          C. TRAS ESTR - ROTA' ),48,.STATUS);
45     1      CALL WRITE(0,.(0DH,0AH, '          5. REFLEXION' ),18,.STATUS);
46     1      CALL WRITE(0,.( '          D. TRAS - ESTR - REFLE' ),48,.STATUS);
47     1      CALL WRITE(0,.(0DH,0AH, '          6. DINETRICO' ),18,.STATUS);
48     1      CALL WRITE(0,.( '          E. ESTR - DIMETRICO' ),45,.STATUS);
49     1      CALL WRITE(0,.(0DH,0AH, '          7. ISOMETRICO' ),19,.STATUS);
50     1      CALL WRITE(0,.( '          F. ESTR - ISOMETRICO' ),44,.STATUS);

```

CALL WRITE(0,.(0DH,0AH,' 0. CONICHA',15,.STATUS)).

51 1 CALL WRITE(0,.( ' 3. VALIDA'-(0DH,0AH,3FH)-37,.STATUS));

52 1 L1: P-CI AND 07FH;  
 53 1 IF P=31H THEN GOTO L1;  
 55 1 IF P=47H THEN GOTO L1;  
 57 1 IF P=31H THEN GOTO EX1;  
 59 1 IF P=32H THEN GOTO EX2;  
 61 1 IF P=33H THEN GOTO EX3;  
 63 1 IF P=34H THEN GOTO EX4;  
 65 1 IF P=35H THEN GOTO EX5;  
 67 1 IF P=36H THEN GOTO EX6;  
 69 1 IF P=37H THEN GOTO EX7;  
 71 1 IF P=38H THEN GOTO EX8;  
 73 1 IF P=39H THEN GOTO EX9;  
 75 1 IF P=41H THEN GOTO EX10;  
 77 1 IF P=42H THEN GOTO EX11;  
 79 1 IF P=43H THEN GOTO EX12;  
 81 1 IF P=44H THEN GOTO EX13;  
 83 1 IF P=45H THEN GOTO EX14;  
 85 1 IF P=46H THEN GOTO EX15;  
 87 1 IF P=47H THEN GOTO L12;

89 1 EX1: CALL WRITE(0,.(1BH,45H,0AH,0AH),4,.STATUS);  
 90 1 CALL WRITE(0,.( ' GRAPH1.3',0DH,0AH),16,.STATUS);  
 91 1 GOTO AB;

92 1 EX2: CALL WRITE(0,.(1BH,45H,0AH,0AH),4,.STATUS);  
 93 1 CALL WRITE(0,.( ' GRAPH1.41',14,.STATUS)-  
 94 1 CALL WRITE(0,.(0DH,0AH,0AH,' GRAPH1.42',17,.STATUS);  
 95 1 CALL WRITE(0,.(0DH,0AH,0AH,' GRAPH1.43',0DH,0AH),19,.STATUS);  
 96 1 GOTO AB;

97 1 EX3: CALL WRITE(0,.(1BH,45H,0AH,0AH),4,.STATUS);  
 98 1 CALL WRITE(0,.( ' GRAPH1.51',14,.STATUS);  
 99 1 CALL WRITE(0,.(0DH,0AH,0AH,' GRAPH1.53',17,.STATUS);  
 100 1 CALL WRITE(0,.(0DH,0AH,0AH,' PRUEBA',0DH,0AH),16,.STATUS);  
 101 1 GOTO AB;

102 1 EX4: CALL WRITE(0,.(1BH,45H,0AH,0AH),4,.STATUS);  
 103 1 CALL WRITE(0,.( ' GRAPH1.61',15,.STATUS);  
 104 1 CALL WRITE(0,.( ' GRAPH1.661',20,.STATUS);  
 105 1 CALL WRITE(0,.( ' GRAPH1.671',20,.STATUS);  
 106 1 CALL WRITE(0,.(0DH,0AH,' GRAPH1.62',17,.STATUS);  
 107 1 CALL WRITE(0,.( ' GRAPH1.662',20,.STATUS);  
 108 1 CALL WRITE(0,.( ' GRAPH1.672',20,.STATUS);  
 109 1 CALL WRITE(0,.(0DH,0AH,' GRAPH1.63',17,.STATUS);  
 110 1 CALL WRITE(0,.( ' GRAPH1.663',20,.STATUS);  
 111 1 CALL WRITE(0,.( ' GRAPH1.673',20,.STATUS);  
 112 1 CALL WRITE(0,.(0DH,0AH,' GRAPH1.64',17,.STATUS);  
 113 1 CALL WRITE(0,.( ' GRAPH1.664',20,.STATUS);  
 114 1 CALL WRITE(0,.( ' GRAPH1.674',20,.STATUS);

```

115 1 CALL WRITE(0,.(0DH,0AH,' GRAPH1.65'),17,.STATUS);
116 1 CALL WRITE(0,.( ' GRAPH1.66'),20,.STATUS);
117 1 CALL WRITE(0,.( ' GRAPH1.67'),20,.STATUS);
118 1 CALL WRITE(0,.(0DH,0AH,' GRAPH1.66'),17,.STATUS);
119 1 CALL WRITE(0,.( ' GRAPH1.666'),20,.STATUS);
PLM-60 COMPILER

120 1 CALL WRITE(0,.( ' GRAPH1.676'),20,.STATUS);
121 1 CALL WRITE(0,.(0DH,0AH,' GRAPH1.67'),17,.STATUS);
122 1 CALL WRITE(0,.( ' GRAPH1.667'),20,.STATUS);
123 1 CALL WRITE(0,.( ' GRAPH1.677'),20,.STATUS);
124 1 CALL WRITE(0,.(0DH,0AH,' GRAPH1.68'),17,.STATUS);
125 1 CALL WRITE(0,.( ' GRAPH1.668'),20,.STATUS);
126 1 CALL WRITE(0,.( ' GRAPH1.678'),20,.STATUS);
127 1 CALL WRITE(0,.(0DH,0AH,' GRAPH1.69'),17,.STATUS);
128 1 CALL WRITE(0,.( ' GRAPH1.669'),20,.STATUS);
129 1 CALL WRITE(0,.( ' GRAPH1.679',0DH,0AH),22,.STATUS);
130 1 GOTO AB;

131 1 EX5: CALL WRITE(0,.(1BH,45H,0AH,0AH),4,.STATUS);
132 1 CALL WRITE(0,.( ' GRAPH1.71'),14,.STATUS);
133 1 CALL WRITE(0,.(0DH,0AH,0AH,' GRAPH1.72'),17,.STATUS);
134 1 CALL WRITE(0,.(0DH,0AH,0AH,' GRAPH1.73',0DH,0AH),19,.STATUS);
135 1 GOTO AB;

136 1 EX6: CALL WRITE(0,.(1BH,45H,0AH,0AH),4,.STATUS);
137 1 CALL WRITE(0,.( ' GRAPH1.91',0DH,0AH),17,.STATUS);
138 1 GOTO AB;

139 1 EX7: CALL WRITE(0,.(1BH,45H,0AH,0AH),4,.STATUS);
140 1 CALL WRITE(0,.( ' GRAPH2.01',0DH,0AH),17,.STATUS);
141 1 GOTO AB;

142 1 EX8: CALL WRITE(0,.(1BH,45H,0AH,0AH),4,.STATUS);
143 1 CALL WRITE(0,.( ' GRAPH2.1',0DH,0AH),16,.STATUS);
144 1 GOTO AB;

145 1 EX9: CALL WRITE(0,.(1BH,45H,0AH,0AH),4,.STATUS);
146 1 CALL WRITE(0,.( ' GRU12.11'),14,.STATUS);
147 1 CALL WRITE(0,.( ' GRU12.21'),20,.STATUS);
148 1 CALL WRITE(0,.( ' GRU12.31'),20,.STATUS);
149 1 CALL WRITE(0,.(0DH,0AH,' GRU12.12'),16,.STATUS);
150 1 CALL WRITE(0,.( ' GRU12.22'),20,.STATUS);
151 1 CALL WRITE(0,.( ' GRU12.32'),20,.STATUS);
152 1 CALL WRITE(0,.(0DH,0AH,' GRU12.13'),16,.STATUS);
153 1 CALL WRITE(0,.( ' GRU12.23'),20,.STATUS);
154 1 CALL WRITE(0,.( ' GRU12.33'),20,.STATUS);
155 1 CALL WRITE(0,.(0DH,0AH,' GRU12.14'),16,.STATUS);
156 1 CALL WRITE(0,.( ' GRU12.24'),20,.STATUS);
157 1 CALL WRITE(0,.( ' GRU12.34'),20,.STATUS);
158 1 CALL WRITE(0,.(0DH,0AH,' GRU12.15'),16,.STATUS);
159 1 CALL WRITE(0,.( ' GRU12.25'),20,.STATUS);
160 1 CALL WRITE(0,.( ' GRU12.35'),20,.STATUS);

```

```

161 1 CALL WRITE(0,.(0DH,0AH,' GRU12.16'),16,.STATUS);
162 1 CALL WRITE(0,.(? GRU12.26'),20,.STATUS);
163 1 CALL WRITE(0,.(? GRU12.36'),20,.STATUS);
164 1 CALL WRITE(0,.(0DH,0AH,' GRU12.17'),16,.STATUS);
165 1 CALL WRITE(0,.(? GRU12.27'),20,.STATUS);
166 1 CALL WRITE(0,.(? GRU12.37'),20,.STATUS);
167 1 CALL WRITE(0,.(0DH,0AH,' GRU12.18'),16,.STATUS);
168 1 CALL WRITE(0,.(? GRU12.28'),20,.STATUS);
169 1 CALL WRITE(0,.(? GRU12.38'),20,.STATUS);
170 1 CALL WRITE(0,.(0DH,0AH,' GRU12.19'),16,.STATUS);
PL/M-80 COMPILER

171 1 CALL WRITE(0,.(? GRU12.29'),20,.STATUS);
172 1 CALL WRITE(0,.(? GRU12.39',0DH,0AH),22,.STATUS);
173 1 GOTO AB;

174 1 EX10: CALL WRITE(0,.(1BH,45H,0AH,0AH),4,.STATUS);
175 1 CALL WRITE(0,.(? GRU13.11'),14,.STATUS);
176 1 CALL WRITE(0,.(? GRU13.21'),20,.STATUS);
177 1 CALL WRITE(0,.(? GRU13.31'),20,.STATUS);
178 1 CALL WRITE(0,.(? GRU13.41'),20,.STATUS);
179 1 CALL WRITE(0,.(? GRU13.22'),20,.STATUS);
180 1 CALL WRITE(0,.(? GRU13.32'),20,.STATUS);
181 1 CALL WRITE(0,.(0DH,0AH,' GRU13.13'),16,.STATUS);
182 1 CALL WRITE(0,.(? GRU13.23'),20,.STATUS);
183 1 CALL WRITE(0,.(? GRU13.33'),20,.STATUS);
184 1 CALL WRITE(0,.(0DH,0AH,' GRU13.14'),16,.STATUS);
185 1 CALL WRITE(0,.(? GRU13.24'),20,.STATUS);
186 1 CALL WRITE(0,.(? GRU13.34'),20,.STATUS);
187 1 CALL WRITE(0,.(0DH,0AH,' GRU13.15'),16,.STATUS);
188 1 CALL WRITE(0,.(? GRU13.25'),20,.STATUS);
189 1 CALL WRITE(0,.(? GRU13.35'),20,.STATUS);
190 1 CALL WRITE(0,.(0DH,0AH,' GRU13.16'),16,.STATUS);
191 1 CALL WRITE(0,.(? GRU13.26'),20,.STATUS);
192 1 CALL WRITE(0,.(? GRU13.36'),20,.STATUS);
193 1 CALL WRITE(0,.(0DH,0AH,' GRU13.17'),16,.STATUS);
194 1 CALL WRITE(0,.(? GRU13.27'),20,.STATUS);
195 1 CALL WRITE(0,.(? GRU13.37'),20,.STATUS);
196 1 CALL WRITE(0,.(0DH,0AH,' GRU13.18'),16,.STATUS);
197 1 CALL WRITE(0,.(? GRU13.28'),20,.STATUS);
198 1 CALL WRITE(0,.(? GRU13.38'),20,.STATUS);
199 1 CALL WRITE(0,.(0DH,0AH,' GRU13.19'),16,.STATUS);
200 1 CALL WRITE(0,.(? GRU13.29'),20,.STATUS);
201 1 CALL WRITE(0,.(? GRU13.39',0DH,0AH),22,.STATUS);
202 1 GOTO AB;

203 1 EX11: CALL WRITE(0,.(1BH,45H,0AH,0AH),4,.STATUS);
204 1 CALL WRITE(0,.(? GRU14.11'),14,.STATUS);
205 1 CALL WRITE(0,.(? GRU14.21'),20,.STATUS);
206 1 CALL WRITE(0,.(? GRU14.31'),20,.STATUS);
207 1 CALL WRITE(0,.(0DH,0AH,' GRU14.12'),16,.STATUS);
208 1 CALL WRITE(0,.(? GRU14.22'),20,.STATUS);
209 1 CALL WRITE(0,.(? GRU14.32'),20,.STATUS);

```





```
ZE = 000EH      8D  
MAXIMUM STACK SIZE = 0006H    6D  
252 LINES READ  
0 PROGRAM ERRORS
```

END OF FLIM-80 COMPILATION

## BIBLIOGRAFIA

- DAVID F. ROGERS , J. ALAN ADAMS  
" MATEMATICAL ELEMENTS FOR COMPUTER GRAFICS "
  
- LOUIS HOHENSTEIN  
" COMPUTER PERIFHERAL FOR MINICOMPUTERS  
, MICROPROCESADOR, AND PERSONAL COMPUTER "
  
- JOSE MONPIN POBLET  
" INTERCONEXION DE PERIFERICOS A MICROPROCESADORES "