FACULTAD DE CIENCIAS

DEPARTAMENTO DE ELECTRICIDAD Y ELECTRONICA

CONCEPCION, DISEÑO Y REALIZACION DE UN MICROORDENADOR PARA OPERAR CON SEÑALES ANALOGICAS

ALBERTO PRIETO ESPINOSA

UNIVERSIDAD DE GRANADA

UNIVERSIDAD DE GRANADA

FACULTAD DE CIENCIAS

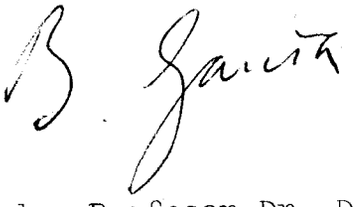
DEPARTAMENTO DE ELECTRICIDAD Y ELECTRONICA

CONCEPCION, DISEÑO Y REALIZACION DE UN MICROORDENADOR PARA OPERAR CON SEÑALES ANALOGICAS.

Memoria presentada por
Alberto PRIETO ESPINOSA
para aspirar al Grado de
Doctor en Ciencias,
Sección de Físicas.

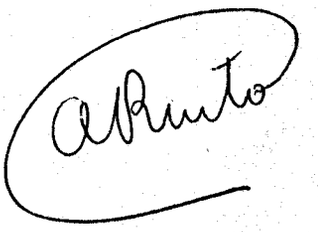
CONCEPCION, DISEÑO Y REALIZACION DE UN MICROORDENADOR PARA OPERAR CON SEÑALES ANALOGICAS

Visado en Granada
en abril de 1976.



Firmado: Profesor Dr. D.
Bernardo GARCIA OLMEDO
Catedratico de Electricidad
y Magnetismo y Director del
Departamento de Electrici-
dad y Electrónica de la
Facultad de Ciencias de la
Universidad de Granada.

Trabajo presentado para el
Grado de Doctor en Ciencias,
Sección de Físicas,
Granada, abril de 1976.



Fdo: Alberto PRIETO ESPINOSA
Licenciado en Ciencias,
Sección de Físicas.

A Mari-Pepa y
Beatriz.

En primer lugar deseo expresar mi profunda gratitud al Prof. Dr. D. Bernardo GARCIA OLMEDO que, no sólo ha dirigido este trabajo, sino que también ha colaborado y ayudado constantemente hasta en los pequeños detalles de su concepción y realización. Al Prof. M. Guy BOULAYE debo sus valiosas sugerencias, orientaciones, críticas y ayuda material, asi como a todo el "Equipe de Recherche en Architecture d'Ordinateurs" y al "Departement d'Informatique" de la Universidad de Rennes, muy expecialmente a MM. Charles Wagner y Gérard Michel.

A los profesores Drs. D. Juan de Dios LOPEZ GONZALEZ y D. Gerardo PARDO SANCHEZ les agradezco su constante interés, ayuda y estímulo por la consecución de mi doctorado.

Deseo también mostrar mi gratitud a las personas que han intervenido en mi iniciación en esta rama de la Ciencia, mis profesores de la Facultad de Ciencias Físicas de la Universidad Complutense de Madrid; profesores GARCIA SANTESMASES, RODRIGUEZ VIDAL, MELLADO, MORENO, ALIQUE, RUBIO, VAQUERO, MIRA, CARTUJO,...

Al profesor Dr. D. José MIRO NICOLAU debo mi iniciación como investigador en Sistemas Digitales, en unos laboriosos e inolvidables años en el Centro de Investigaciones Técnicas de Guipuzcoa y E.T.S. de Ingenieros Industriales de San Sebastián.

A D. Placido REYES le agradezco su contribución a mi formación práctica: en todo momento supo comunicarme su gran experiencia e intuición de Laboratorio.

Agradezco al Dr. D. Antonio LLORIS su ayuda en la redacción de esta memoria. A los compañeros del Centro de Cálculo y del Departamento de Electricidad y Electrónica les debo su gran colaboración, destacando la de D. Francisco JURADO y Buenaventura CLARES. También expreso mi sincero agradecimiento a las Srtas. Mari Carmen GARCIA y María Angustias VILLOSLADA por su eficiencia y diligencia en la realización material de esta memoria.

Por último, y no por ello con menor importancia, agradezco a la Comisión de Investigación de la Presidencia de Gobierno su ayuda material, sin la cual no hubiese podido realizar este trabajo.
Al Ministerio de Asuntos Extranjeros de Francia y a la Agregaduría Científica en España les agradezco sus ayudas que me permitieron permanecer el curso 1974-75 en la Universidad de Rennes y ampliar mi formación en mini y microordenadores en París.

Me gustaría que todos los aquí citados se sintiesen copartícipes de esta investigación.

PROLOGO

Al iniciar esta investigación nos propusimos, siguiendo con nuestra línea de investigación en sistemas híbridos, construir un ordenador digital para operar con señales analógicas. Se pretendía, utilizando técnicas de microprogramación, grabar en una memoria ROM (Read Only Memory) una serie de algoritmos que controlasen la realización por una unidad aritmético-lógica, de operaciones básicas con funciones prèviamente digitalizadas y memorizadas en una memoria RAM (Random Access Memory). Desde un teclado se posibilitaría la ejecución de instrucciones (producto, integral,... de funciones) paso a paso, tal y como se realiza en un calculador convencional. Pretendíamos posteriormente, en una segunda etapa, partiendo del calculador diseñado y haciendo uso de un segundo nivel de microprogramación y grabación de memorias ROM, realizar un calculador programable con programa almacenable en memoria, es decir, un ordenador.

Debido a la comercialización en gran escala, y fácil adquisición de "microprocesadores", decidimos utilizar uno de ellos para nuestro dispositivo. El microprocesador utilizado es un circuito integrado de tecnología MOS que contiene en tan sólo una pastilla de unos 2 cm^2 y de 18 terminales, un procesador; es decir, una unidad de control, una unidad aritmetico-lógica, un registro contador de programa, ... Este circuito, al igual que todo microprocesador, está tan perfeccionado que se utiliza con un lenguaje de programación en el que existen no sólo instrucciones aritméticas y de transferencia de información, sino también instrucciones de bifurcación y llamadas a rutinas.

La filosofía de arquitectura de diseño viene en gran parte impuesta por las características del microprocesador. Utilizando este circuito en lugar de realizar una unidad aritmética y de control clásica, además de poder conseguir nuestro objetivo de una forma más eficaz, podiamos plantearnos la realización en una sóla etapa del ordenador programable, e introduciriamos en nuestro laboratorio una nueva técnica de diseño de sistemas digitales.

Como es lógico el lenguaje básico del microprocesador no era apto para operar directamente con señales analógicas; esto nos llevó a definir un "lenguaje intérprete" orientado a operar con funciones (LPF) y a la construcción del traductor correspondiente.

También hubo que realizar un conjunto de programas para el control del funcionamiento del ordenador; por lo que tuvimos que construir un MONITOR y definir su lenguaje de utilización ("lenguaje de control"). Todo el Sistema Operativo (S.O.) (monitor, traductor y subrutinas asociadas), escrito en el lenguaje básico del microprocesador, está grabado en una memoria REPROM, similarmente a como se hace en las técnicas de microprogramación.

Para conseguir con pocos bits el acceso fácil a toda la memoria se ha utilizado ampliamente el concepto de paginación.

Se han seguido en lo posible las ideas expuestas en el pasado año de 1975, por Nicoud (referencia 90) sobre generalización y unificación de nomenclatura sobre "software" y lenguajes nemotécnicos en mini y microprocesadores.

Esta memoria pretende recoger los aspectos más importantes y conclusiones de nuestra investigación, que consideramos una aportación al desarrollo de la arquitectura de sistemas de tratamiento numérico de señales analógicas, utilizando algunas de las técnicas más avanzadas del momento.

Granada 31 de Marzo de 1976.

PROPOSITOS Y OBJETIVOS DE LA INVESTIGACION:

El presente trabajo tiene por objeto la concepción, diseño y realización de un prototipo de microordenador para operar con señales analógicas.

El desarrollo de la investigación se han centrado fundamentalmente en los siguientes aspectos:

a) La realización de un microordenador de propósito general.
b) Adaptación al mismo de dispositivos adecuados de entrada y salida para funciones analógicas con la resolución de los problemas de interfase.
c) Síntesis de un sistema operativo (S.O.) para el control del funcionamiento del dispositivo.
d) Definición de un lenguaje para operar con funciones (LPF) y construcción del traductor correspondiente.
e) Definición de un lenguaje ensamblador (LE) e implementación del "ensamblador cruzado" correspondiente.

El lenguaje operativo a definir debe poseer instrucciones tales como suma, integración,... de funciones, de tal forma que con un programa que se almacena en la memoria se puedan evaluar expresiones matemáticas en las que los datos sean funciones analógicas.

Se ha pretendido realizar un dispositivo con las siguientes posibilidades:

- orientar su uso a la realización de cálculos con funciones utilizando técnicas numéricas y ofreciendo las facilidades que da un lenguaje de programación.
  (10), (26), (37), (56), (63), (70), (72), (80), (93), (108), (110), (111).
- utilizarlo como un microordenador de propósito general, sin tener necesidad de operar con funciones.
- usarlo "en línea" con otros dispositivos (de control), por ejemplo (71), (102), (109) y de trabajar en "tiempo real".
- utilizarlo como simulador de dispositivos híbridos de uso especial; así por ejemplo, poder simular memorias incrementales (49), (87).
- operar como gran cantidad de sistemas electrónicos concretos que lleven implícito la realización digital de cálculos analógicos (filtros digitales (54),...).

El objeto de la investigación es la realización de un prototipo con la mayor modularidad posible buscando soluciones óptimas y originales tanto en la concepción desu arquitectura como en su diseño.

La organización de la Memoria se ha realizado de la forma siguiente:

El capítulo I constituye una introducción con la que se pretende centrar el trabajo de investigación dentro del cálculo electrónico, y hacer algunas consideraciones sobre las funciones analógicas a procesar.

También en este capítulo se exponen las características del dispositivo, se presenta la estructura básica adoptada para el ordenador teniendo en cuenta las especificaciones y funcionamiento del microprocesador, y se establece la filosofía de concepción y diseño de la totalidad del sistema.

El capítulo II contiene las definiciones y repertorios de instrucciones de los lenguajes del microordenador: lenguaje máquina (LM), lenguaje ensamblador (LE), lenguaje orientado a operar con funciones (LPF) y lenguaje de control (LC), así como la descripción del panel de control del microordenador y todo lo relacionado con su utilización.

El capítulo III estudia los algorítmos, organigramas y programas del Sistema Operativo, y se comentan los aspectos relacionados con la imbricación e implementación de programas en la memoria REPROM. La formulación de los algorítmos va precedida de algunas consideraciones sobre la aritmética utilizada.

El capítulo IV recoge los aspectos relacionados con la física del ordenador: diseño de los circuitos estructurales y de control. Todo ello se ilustra con consideraciones relativas a la realización de montaje.

El capítulo V contiene unas consideraciones autocríticas en base a las cuales se sugieren optimizaciones posibles del sistema. El capítulo finaliza enumerando las conclusiones y principales aportaciones.

En los apéndices se recoge información relativa a la notación y terminología utilizada, se presenta la totalidad de programas del Sistema Operativo tal y como se han grabado en la memoria REPROM y se exponen los esquemas del montaje del dispositivo.

Para facilitar la lectura de la Memoria deben tenerse presente las siguientes observaciones:

1) Las referencias bibliográficas se dan mediante un número entre paréntesis; en algunas de ellas se indica también la página o capítulo donde se encuentra la información referenciada; esto se hace utilizando las iniciales, p, ó c, respectivamente. Ejemplo: (27), (13,c2), (17,p325).

2) Cuando se hace alusión a cualquier parte del texto se indica el epígrafe o sección que la contiene con la abreviatura secc. seguida del número del capítulo y de la sección. Ejemplo: (secc. 3.2.1).

3) Se utiliza la notación IVERSON (APL) (7), (54), (79) para describir las operaciones y transferencias de información en el interior del ordenador. En el apéndice I se recoge un breve resumen de los símbolos IVERSON utilizados.

4) En las referencias (36), (39), (44), (51) y (4) (9) pueden verse las acepciones empleadas a lo largo de este Memoria sobre diversos términos informáticos relacionados con el "Software". Hay que tener presente que la Informática, como toda ciencia nueva no tiene una terminología clásica universalmente aceptada.

5) Para los esquemas lógicos de circuitos se ha seguido fundamentalmente la notación IEEE/ANSI (30).

6) En el apéndice II se recopilan las abreviaturas más utilizadas a lo largo de la Memoria.

INDICE

CAPITULO III. ALGORITMOS Y PROGRAMAS DEL SISTEMA OPERATIVO.-

CAPITULO IV. DISEÑO ELECTRONICO DEL DISPOSITIVO.-

Págs.

CAPITULO I
INTRODUCCION

1.1. INTRODUCCION:

Este primer capítulo se propone dar al lector de esta Memoria una visión general del trabajo realizado y servir de introducción a los demás capítulos dedicados a llevar a la práctica las ideas aquí expuestas sobre la arquitectura del sistema.

En primer lugar se pretende centrar el tema de la investigación haciendo unas breves consideraciones comparativas

entre el cálculo analógico, digital e híbrido. Posteriormente se indica el tipo de funciones susceptibles de ser procesadas y se analizan los errores que se dan en los procesos de muestreo y digitalización. A continuación se exponen las características básicas del microprocesador, tanto físicas como operativas, para pasar a describir la organización y configuración del ordenador. En cuanto al microprocesador utilizado sólo se presenta la información necesaria a fin de comprender su funcionamiento y sentar las bases para definir la estructura básica de la totalidad del dispositivo.

Finalmente se especifica y justifica la organización y filosofía dada al Sistema Operativo para terminar exponiendo la estructuración de memoria.

1.2 CALCULO ANALOGICO, DIGITAL E HIBRIDO.-

Una de las aplicaciones en la que la electrónica ha tenido un mayor desarrollo es sin duda la del cálculo matemático. El cálculo se puede hacer analógica o digitalmente; en el primer caso las variables cambian en el tiempo de forma continua y en el segundo sólo pueden tomar valores discretos. Los dispositivos pueden ser de "propósito general" cuando se han diseñado para resolver una gran variedad de problemas, o de "uso concreto o especial" en el caso de que su finalidad sea la resolución de un tipo de problemas concretos; si no se especifica lo contrario, los conceptos de calculador u ordenador se refieren a los sistemas de propósito general.

En 1925 el Dr. Bush del M.I.T. (119) diseñó el primer calculador eléctrico, que era de naturaleza analógica; como todos los de este tipo, sobre un panel se configuraban circuitos eléctricos que se regían por las mismas ecuaciones matemáticas que el proceso físico objeto de estudio. Analizando las variables del circuito (tensiones o corrientes) se resolvía por analogía el problema planteado. El primer calculador digital electrónico digno de mencionar, MARK-1, se construyó 19 años después (1944), y fué realizado por el Prof. Aiken de la Universidad de Harward.

Hasta el año 1950 (111) las técnicas de cálculo analógico se utilizaron mucho ya que sólo ellas permitían la evaluación de expresiones que llevaban consigo la realización de gran cantidad de operaciones. Su papel disminuyó con el perfeccionamiento de las técnicas numéricas y el desarrollo de los calculadores digitales; no obstante, el cálculo analógico sigue ofreciendo posibilidades complementarias.

CALCULO DIGITAL	CALCULO ANALOGICO
Tratamiento secuencial sobre variables discretas.	Tratamiento paralelo con variables continuas.
Todo problema se descompone en numerosas operaciones muy elementales ligadas a un ciclo base de aproximadamente 1 microsegundo.	Alta velocidad, operación en "tiempo real".
Precisión limitada unicamente por el número de bits que se utilizan para representar los datos.	Precisión baja (2 ó 3 cifras decimales exactas, menor que el 0,01 % del fondo de escala (9)) y dada por la calidad de los componentes del sistema.
Posibilidad de trabajar en coma flotante, eliminandose problemas de factores de escala.	Problemas de factor de escala.
Magnitud y complejidad del problema limitada sólo por el tiempo disponible de ordenador.	Magnitud y complejidad del problema limitada por la cantidad y tipo de componentes que constituyen el sistema y por la acumulación de errores.
Posibilidad de memorizar indefinidamente datos pudiendo acceder a ellos con gran facilidad.	Dificultades en el acceso y memorización indefinida de datos.
Gran versatilidad en cuanto al tipo de problemas que puede resolver (estadística, gestión, bancos de datos,...). Gran perfeccionamiento en técnicas de utilización (software muy desarrollado)	Aplicaciones muy limitadas (resolución de sistemas de ecuaciones diferenciales,...).
Posibilidad de tomar decisiones lógicas y representar datos numéricos, alfabéticos,...	Imposibilidad de tomar decisiones lógicas y representar datos alfabéticos,...

Tabla 1.1
Resumen comparativo entre ordenadores digitales y analógicos.

En la tabla 1.1 se puede ver un resumen comparativo entre los ordenadores analógicos y digitales. Las principales ventajas de los primeros radican en la facilidad de ejecutar diversos problemas concretos (resolución de sistemas de ecuaciones diferenciales, por ejemplo), la superior velocidad sobre el cálculo digital y la eficiencia en la implementación electrónica de operaciones, como la integración (45),...

El cálculo digital, por su parte, reune las características de que la precisión operativa no depende de la calidad de los componentes utilizados, la facilidad de memorizar datos numéricos o no numéricos indefinidamente y la gran variedad de problemas que se pueden resolver.

El cálculo híbrido pretende reunir las ventajas de los dos métodos mencionados:

- .combinar la velocidad de cálculo analógico con la precisión del digital.
- .utilizar subrutinas electrónicas analógicas con aumento de posibilidades.
- .poder trabajar con datos discretos y continuos.
- .hacer uso de las facilidades de memorización digital,...

En cuanto a los sistemas orientados a la resolución de un problema concreto (control de procesos, instrumentación; por ejemplo), el cálculo analógico juega usualmente un papel predominante, aunque poco a poco se van imponiendo las técnicas digitales (24), (71), según se aumenta la escala de integración y la velocidad operativa.

El dispositivo objeto de esta investigación puede considerarse como un sistema o microordenador híbrido, ya que está orientado a la realización de operaciones con funciones analógicas utilizando técnicas digitales; no obstante no tiene las posibilidades y ventajas de los sistemas híbridos en cuanto a la realización de cálculos en forma analógica; por ejemplo, no puede resolver analógicamente ecuaciones diferenciales.

Las entradas al dispositivo realizado pueden ser señales analógicas, digitalizadas por un conversor A/D; todo el procesamiento interno se hace por medio de algoritmos numéricos y la salida puede ser dada con un conversor D/A en forma analógica. Se pretende dar la posibilidad de efectuar un tratamiento numérico a funciones analógicas utilizando las facilidades que da un lenguaje de programación y un ordenador digital. El tratamiento de funciones se puede realizar también sin necesidad de utilizar los conversores de entrada / salida.

1.3 CONCEPTOS SOBRE LAS FUNCIONES A PROCESAR:

La mayor parte de las variables físicas son funciones continuas del tiempo. Esta funciones, por medio de transductores adecuados, pueden convertirse en señales eléctricas (corrientes o tensiones) (93), que en este trabajo se denominan "funciones analógicas". Así es fácil traducir a señales eléctricas: temperaturas, presiones, variables fisiológicas, ... Es indudable el interés que tiene el tratamiento matemá-

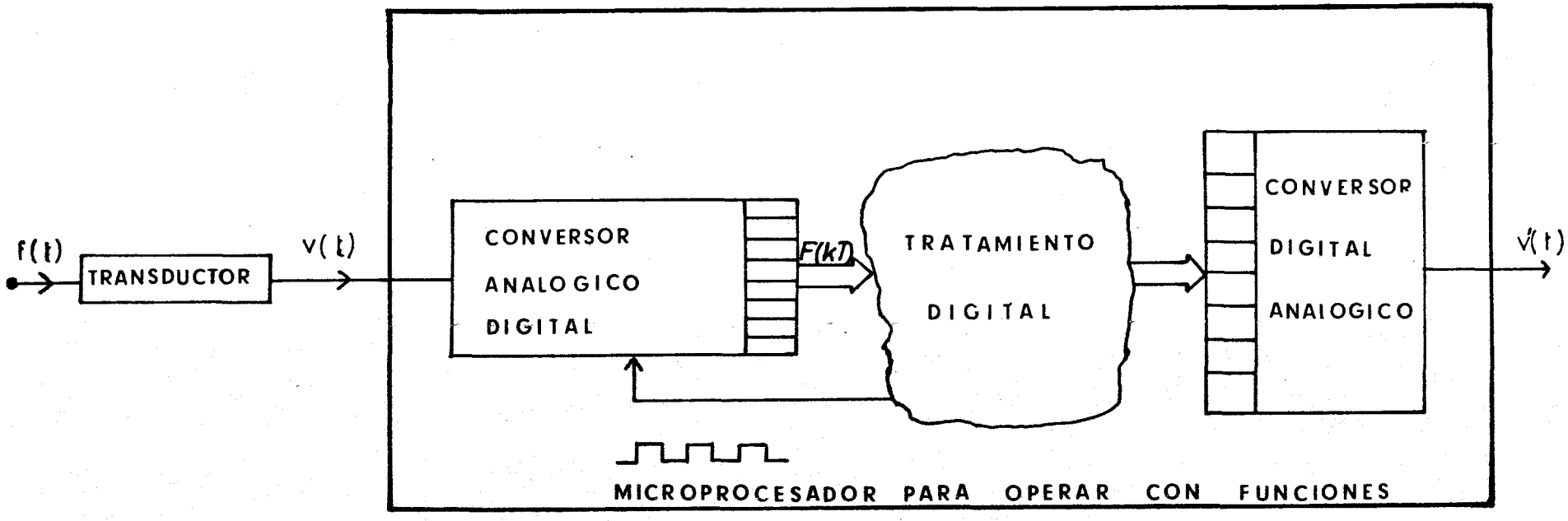


FIG. 1-1

EL MICROORDENADOR REALIZA UN TRATAMIENTO DIGITAL DE SEÑALES ANALOGICAS

tico de estas señales (10), (37), (46), (56), (63), (70), (108). El prototipo objeto del presente trabajo permite el procesamiento de este tipo de funciones.

Las funciones analógicas, según se indica en la figura 1.1, pueden ser aceptadas como entradas por un conversor A/D, que proporciona las salidas en forma digital y, por tanto, en un formato susceptible de ser comprendido por el ordenador. Los resultados, a su vez, están expresados en forma digital y, por medio de un conversor D/A, pueden ser dados a la salida en forma de una señal analógica.

Las funciones analógicas, como para todo conversor, deben ser amplificadas o atenuadas antes de introducirlas en el microordenador de tal forma que estén comprendidas entre unas determinadas cotas de amplitud (A,B).

Bajo el control del ordenador, el conversor A/D efectúa un muestreo en amplitud (figura 1.3) y cada observación es convertida en su correspondiente expresión digital con un fondo de escala de 8 bits.

Cuando el ordenador no trabaja en tiempo real solamente se lee un intervalo acotado de la función externa: (figura 1.2):

$$f(t) \quad \text{donde } a \leq t \leq b$$

El muestreo que efectua el conversor de entrada es periódico, de periodo T, realizándose una partición uniforme del intervalo considerado.

Así pues, en el microordenador, una función queda definida por un conjunto finito, y por tanto numerable, de valores digitales que se almacenan en posiciones consecutivas de memoria:

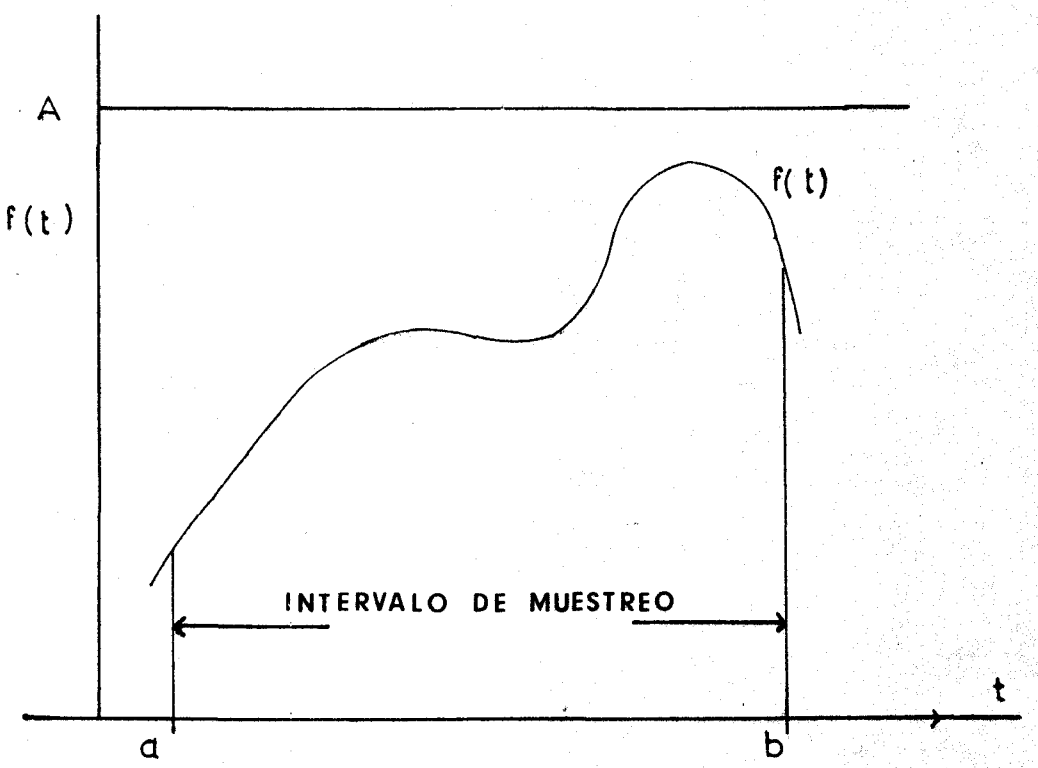


FIG. 1-2

ENTRADA AL CONVERSOR A/D

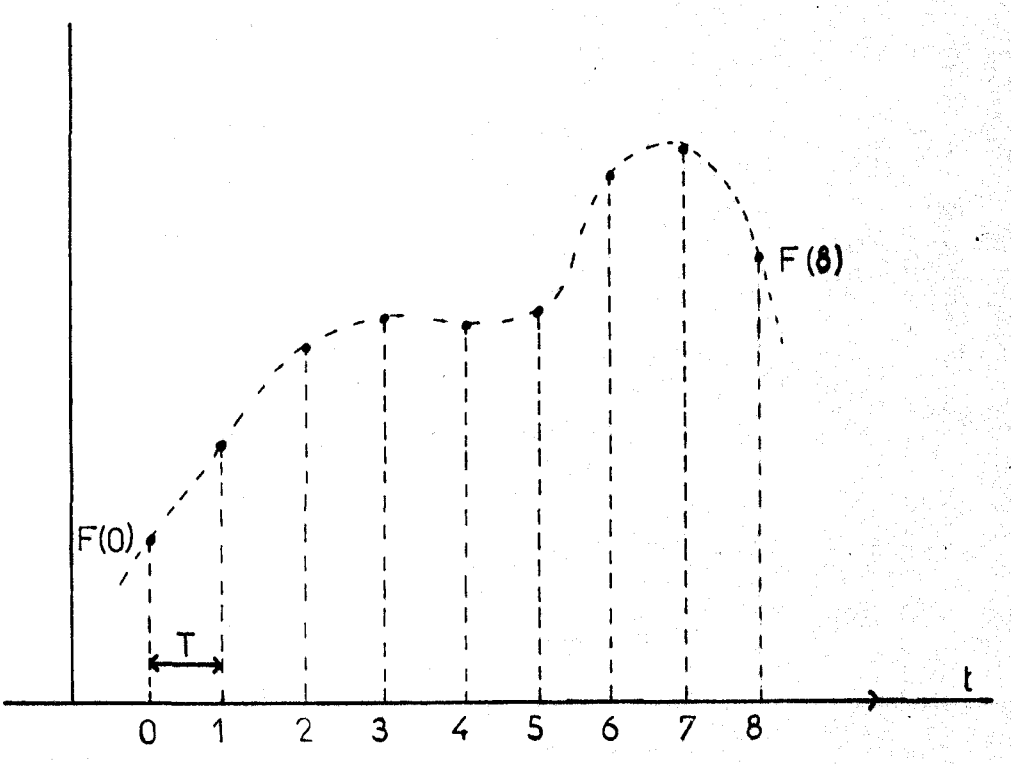


FIG. 1-3

MUESTREO QUE REALIZA EL CONVERSOR

$f(t) \quad a \leq t \leq b \longrightarrow F(kT)$ donde k toma los valores 0,1,2,...

abreviadamente el conjunto de los puntos se denomina F(kT) o simplemente F.

Las señales a procesar, en principio, son funciones reales de variable real, uniformes y acotadas sobre un intervalo (A,B). Con una programación adecuada se pueden tratar igualmente funciones complejas.

El programador o usuario del microordenador puede utilizar, para el control de la conversión, dos parámetros:

- la velocidad o periodo de conversión y
- la longitud del intervalo de tiempo a tomar de la función de entrada o, lo que es lo mismo, el número de puntos a memorizar.

Estos parámetros varían dentro de determinados límites.

1.4 ORGANIZACION Y CONFIGURACION DEL ORDENADOR:

En esta sección se pretende hacer una descripción del ordenador indicando la filosofía y criterios de diseño que se han seguido.

En primer lugar se presenta la estructura básica del microordenador, para lo cual previamente se efectua una descripción de la constitución y funcionamiento del microprocesador utilizado.

Posteriormente se exponen las soluciones aportadas para la realización del sistema operativo teniendo en cuenta el soporte físico en el qué se va a implementar, y se desarrollan los aspectos teóricos y prácticos de mayor interés.

Por último se indica como se ha efectuado la organización de memoria.

1.4.1 ORGANIZACION FISICA DEL MICROORDENADOR:

El microordenador está constituido (figura 1.7) por:

- un microprocesador, que contiene una unidad aritmético-lógica y una unidad de control. El lenguaje básico propio del microoprocesador constituye el lenguaje máquina del ordenador.
- dispositivos de entrada y salida.
- multiplexo de cuatro canales de información en uno. Cada canal es de 8 bits.
- consola o panel de control.
- circuitos combinacionales y secuenciales de control y
- memoria organizada en palabras de 8 bits (octetos). Una parte de ella (2k) es de tipo ROM y contiene el Sistema Operativo o Software del ordenador, y el resto (hasta 5k) es de tipo RAM de escritura/lectura.

El próximo epígrafe se dedica al microprocesador y el siguiente (1.4.1.2) a la estructura básica del ordenador.

1.4.1.1 MICROPROCESADOR:

Los microprocesadores y microordenadores han adquirido un gran desarrollo en la actualidad, dando una nueva orientación a la filosofía de diseño de sistemas; es muy amplia la bibliografía existente en la actualidad sobre este tema, pudiendose clasificar de la siguiente forma:

- información básica (definición de microprocesador, características, constitución, funcionamiento,...)
  (5), (34), (41), (63), (64), (65), (66), (78), (88), (96), (107), (114).
- análisis comparativo entre distintos microprocesadores.
  (116), (118), (120).
- arquitectura de microordenadores (diseño, interfase,...)
  (57), (58), (70), (71).
- lenguajes, ensambladores, compiladores,...
  (33), (35), (85), (90), (94), (95), (70).
- aplicaciones (10), (24), (92), (46), (71).

Se remite al lector interesado a dichas referencias ya que en la presente memoria sólo se considera el microprocesador elegido.

Puesto que la filosofía básica del sistema es independiente del microprocesador utilizado, en este caso se ha elegido el INTEL 8008, además de por su disponibilidad en el mercado, por las siguientes razones.

1º Es un microprocesador que opera con palabras de 8 bits, puede direccionar hasta 16 K de memoria, y el tiempo de ciclo es de 2 a 3 microsegundos. Estas características unidas a las de su lenguaje básico (secc. 2.2), lo hacen adecuado para la construcción de un prototipo como es el dispositivo objeto de esta investigación.

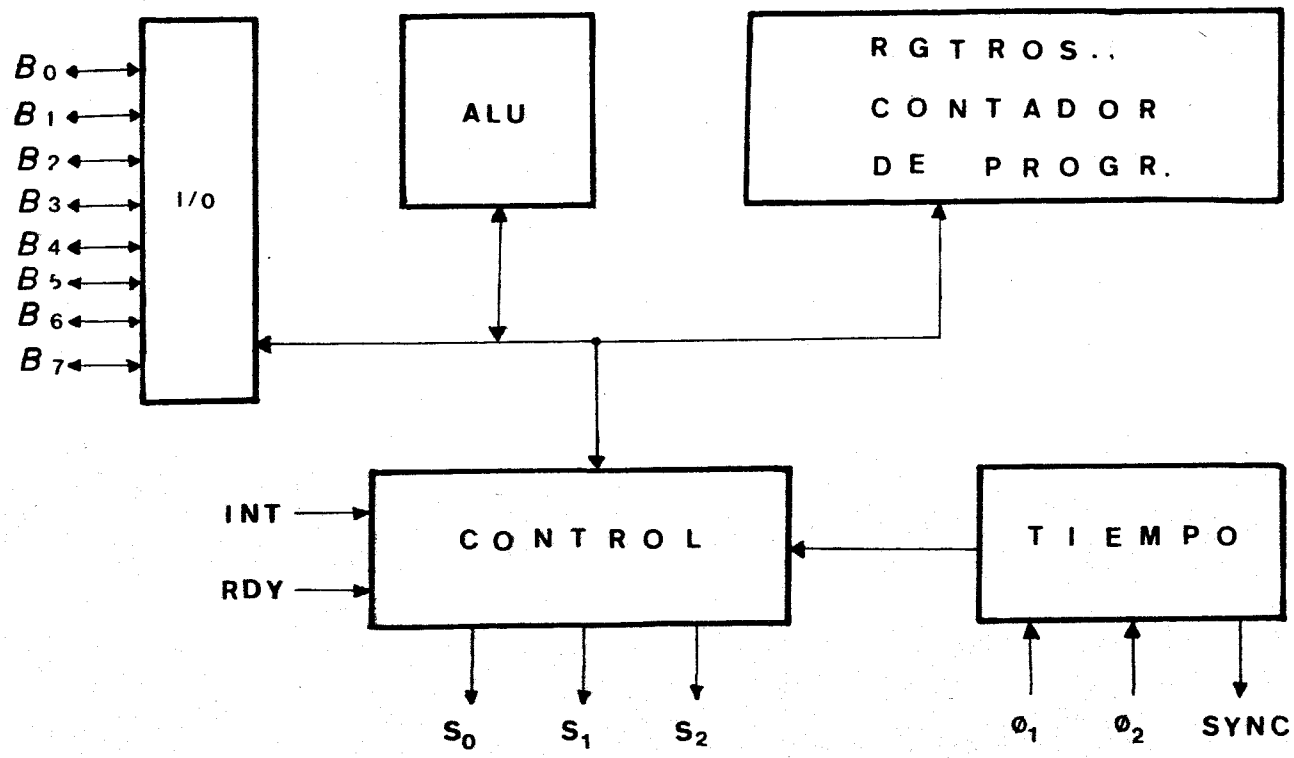


FIG. 1-4

DIAGRAMA DE BLOQUES BASICO DEL MICROPROCESADOR

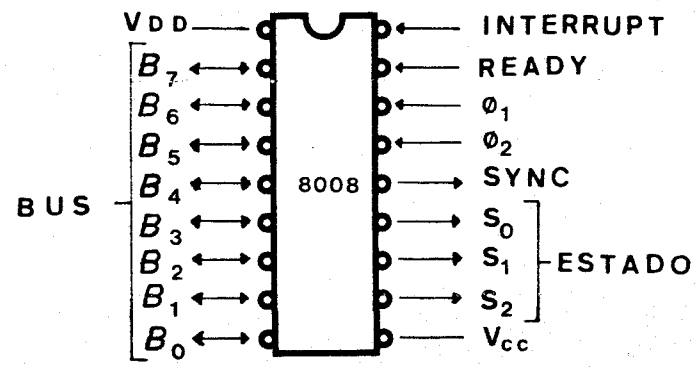


FIG. 1-5

TERMINALES DEL MICROPROCESADOI

2º A pesar de las constantes mejoras introducidas en los nuevos microprocesadores, en cuanto a velocidad, potencia de lenguaje, etc., el Intel 8008 sigue siendo característico y representativo de este tipo de circuitos.

3º El costo del circuito elegido estaba en consonancia con las posibilidades económicas para este proyecto aunque el diseño, algoritmos del S.O.,... se hubiese simplificado considerablemente utilizando otros microprocesadores más potentes (INTEL 8080, MOTOROLA 6800,...).

A continuación se describe brevemente el funcionamiento y constitución del microprocesador.

1.4.1.1.1 FUNCIONAMIENTO: (64), (65), (66).

En la figura 1.4 se muestra un esquema básico del microprocesador, constituido por una pastilla MOS de 18 terminales, figura 1.5.

El microprocesador intercambia con su exterior direcciones de memoria, instrucciones y datos a través de un bus bidireccional, B, de 8 bits. Las direcciones de memoria (14 bits) se dan multiplexadas en el tiempo.

El microprocesador necesita dos alimentaciones ($V_{DD} = -9V \pm 5\%$ y $V_{CC} = \pm 5V \pm 5\%$) y dos señales de reloj (Ø1 y Ø2). Aparte de las instrucciones que entran por el bus B, se dispone de dos entradas auxiliares (READY e INTERRUPT) y tres de salida (SO, S1, S2) que definen el estado del microprocesador.

El terminal READY permite hacer entrar el microprocesador en el estado de "ESPERA" o salir del mismo (tabla 1.4). A través del terminal INTERRUPT, y con circuitos externos adecuados, se pueden insertar instrucciones manualmente.

La ejecución de una instrucción lleva consigo la realización de uno, dos o tres "ciclos". Cada ciclo está constituido por varios "estados", caracterizados por las señales de salida SO, S1 y S2 (tabla 1.2). Los ciclos pueden ser de lectura de memoria (PCI o PCR), escritura de memoria (PCW) o entrada/salida del ordenador (PCC) (tabla 1.3).

En la tabla 1.5 se indican las transferencias de información que se realizan en el bus bidireccional dependiendo de los ciclos y estados.

SO	S1	S2	estado
0	1	0	T1
0	1	1	T1I
0	0	1	T2
0	0	0	espera
1	0	0	T3
1	1	0	STOP
1	1	1	T4
1	0	1	T5

Tabla 1.2

estados posibles del microprocesador.

	CH6	CH7	
PCI	0	0	lectura de memoria ins/dato.
PCR	0	1	lectura de memoria.
PCC	1	0	entrada/salida.
PCW	1	1	escritura en memoria.

tabla 1.3

ciclos posibles del microprocesador.

estados	forma de	
	entrar	salir
ESPERA	READY = 0	READY = 1
STOP	instruc.HLT	INTERRUPT = 1

tabla 1.4

método de entrada y salida en los estados de ESPERA y STOP.

	PCI	PCR	PCW	PCC
T1	B <—— CPL (salida)	B<——L (salida)	B<—— L (salida)	B <—— A (salida)
T2	B <—— CPH (salida)	B<——H (salida)	B<——H (salida)	B<—— b (salida)
T3	(RI o b)<—M(CP) (entrada)	b<— B (entrada)	B<——b (entrada)	b<—— B (entrada)
T4 T5	E J E C U C I O N			

B: bus de entrada/salida.

CP: contador de programas (CP13,..., CP0)

CPL: 8 bits menos significativos (CP7,..., CP0)

CPH: 6 bits más significativos (CP13,...,CP8)

M: memoria principal externa.

A: acumulador.

H,L: registros índices del microprocesador.

RI: registro de instrucciones del microprocesador.

b: registro auxiliar de datos del microprocesador.

(1): sólo en isntrucciones de entrada al ordenador.

tabla 1.5

transferencias de información en ciclos/estados

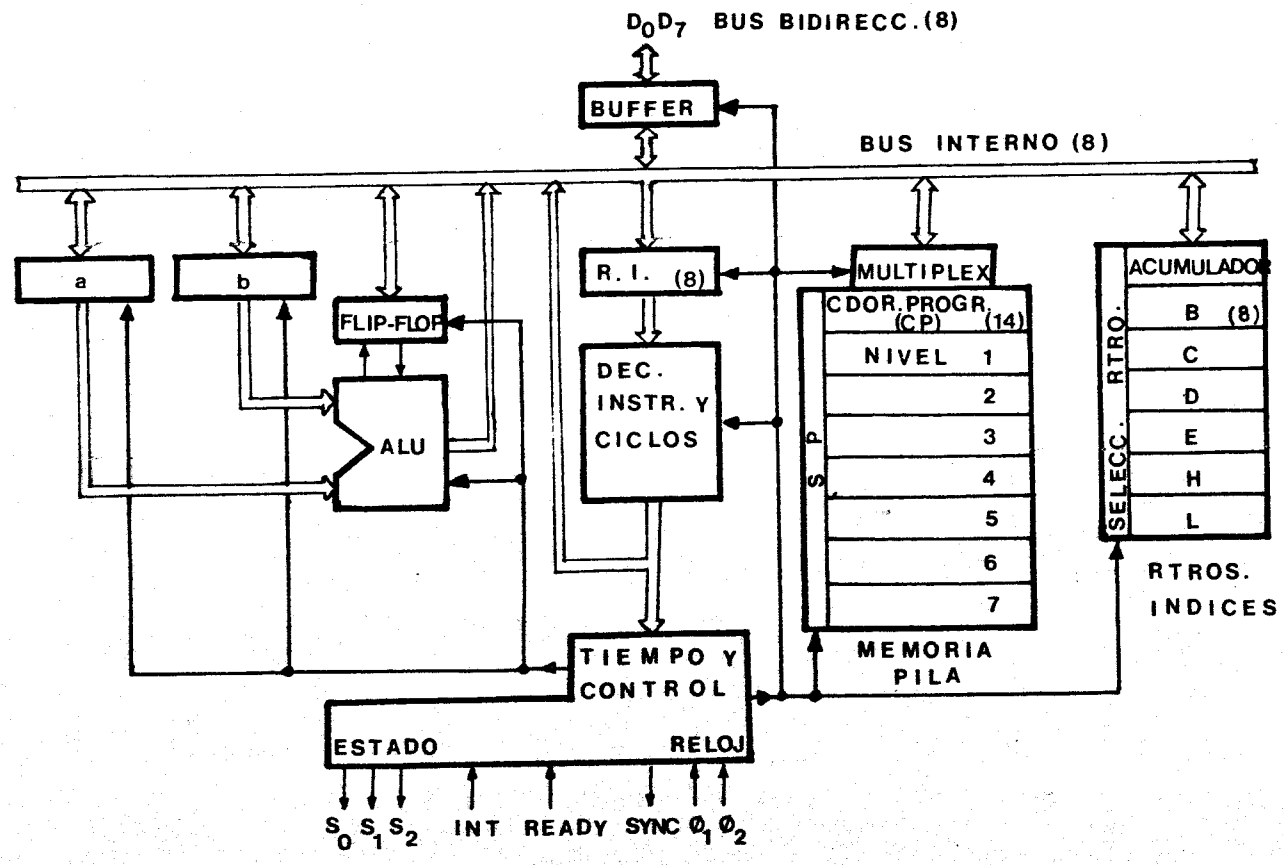


FIG. 1-6

DIAGRAMA BASICO DEL MICROPROCESADOR

1.4.1.1.2 CONSTITUCION.-

En la figura 1.6 puede verse un diagrama básico del microprocesador. Las partes fundamentales, que están conectadas entre sí por un bus interno de ocho bits, son las siguientes:

. unidad aritmético-lógica (ALU):

En ella se realizan las siguientes operaciones: suma, suma teniendo en cuenta el contenido de la báscula de rebose, resta, resta teniendo en cuenta el contenido de la báscula de rebose, AND, exclusive OR, OR, comparación, incremento, decremento y rotación a derecha e izquierda. Cuatro básculas banderas ("flagg") indican la ocurrencia de rebose, contenido nulo, paridad y signo (MSB), de los resultados de las operaciones realizadas. Esto permite realizar saltos condicionales dentro de un programa.

. registros intermedios de entrada/salida:

Estos registros ("buffer") son bidireccionales y comunican al microprocesador con el resto del dispositivo. Están controlados por el registro de instrucciones y los circuitos de control de estados.

. registro de instrucciones y control: las instrucciones leidas de la memoria principal externa son almacenados en el registro de instrucciones (RI), y decodificadas para generar las señales de control de memorias, unidad aritmético-lógica (ALU) y transición de estados.

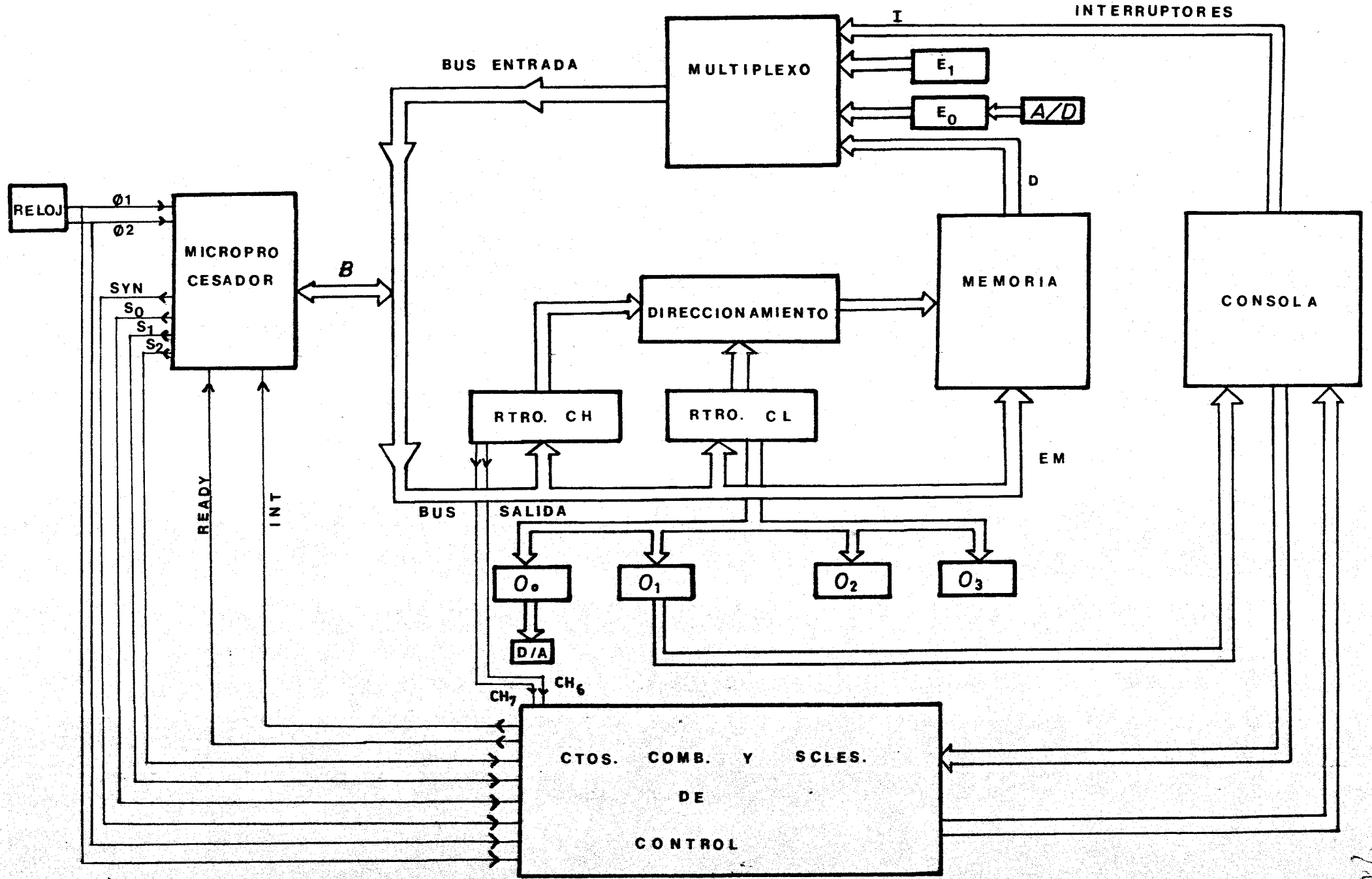


FIG. 1-7 - DIAGRAMA GENERAL DEL MICROORDENADOR

. Memorias: Existen dos memorias (de tipo dinámico):

1) Memoria pila de direcciones: contiene 8 registros de 14 bits. El primer registro es el contador de programa (CP), éste se incrementa en los estados T1. Los demás registros facilitan la conexión automática de rutinas por hardware. En los estados T1 y T2 de los ciclos PCI, según se ha indicado en la tabla 1.5, se obtiene en el bus externo el contenido del contador de programa.

2) Registros índices o memoria "scratch": contiene 7 registros índices de ocho bits que se denominan:

A(acumulador), B,C,D, H y L.

A estos registros se tiene acceso por medio del lenguaje LM.
El lenguaje LPF sólo permite utilizar A,B, y C.
El lenguaje LM permite el direccionamiento indirecto e indexado para lo cual se utilizan los registros índices H y L, que contienen conjuntamente la dirección de memoria completa: el H los 7 bits más significativos, y el L los 8 restantes.

1.4.1.2 ESTRUCTURA BASICA DEL MICROORDENADOR:

En la figura 1.7 puede verse un diagrama general del ordenador.

La información dada por el microprocesador en los estados T1 y T2 se memoriza en los registros CL y CH, respectivamente.

Los bits CH6 y CH7 indican (tabla 1.3) el ciclo en el que se encuentra el microprocesador y constituyen señales de entrada para los circuitos de control.

En los ciclos PCI, PCR y PCW los bits CH5 a CH0 y CL7 a CL0 contienen una dirección de memoria; esta dirección es decodificada

de acuerdo con los requerimientos de los circuitos integrados utilizados como memoria según se verá en el capítulo 4.

Cuando hay salida de información, en los ciclos PCC, el contenido de CL constituye el dato de salida, por lo que este registro está conectado a los distintos dispositivos de salida (Oo, O1, O2 y O3).

El bus de salida está también contectado a la entrada de memoria ya que en los ciclos PCW, estado T3, el microprocesador facilita el dato a memorizar.

Por medio de un multiplexo, se puede dirigir al bus de entrada la información procedente de cualquiera de los siguientes dispositivos:

- registros de entrada EO y E1.
- salida de memoria (D)
- interruptores del panel (I).

Los circuitos de control utilizan como entradas básicas las señales siguientes:

. relojes (Ø1 y Ø2)
. sincronismo (SYN)
. estado (SO, S1 y S2)
. ciclo (CH6 y CH7) y
. pulsadores e interruptores de consola.

las salidas principales son:

. control de carga de los registros CH y CL
. control de decodificación de direcciones de memoria
. control de lectura/escritura de memoria
. control de multiplexo
. control de dispositivos de entrada/salida (básculas RS).
. señales INTERRUPT y READY.
. interfase con conversores,...

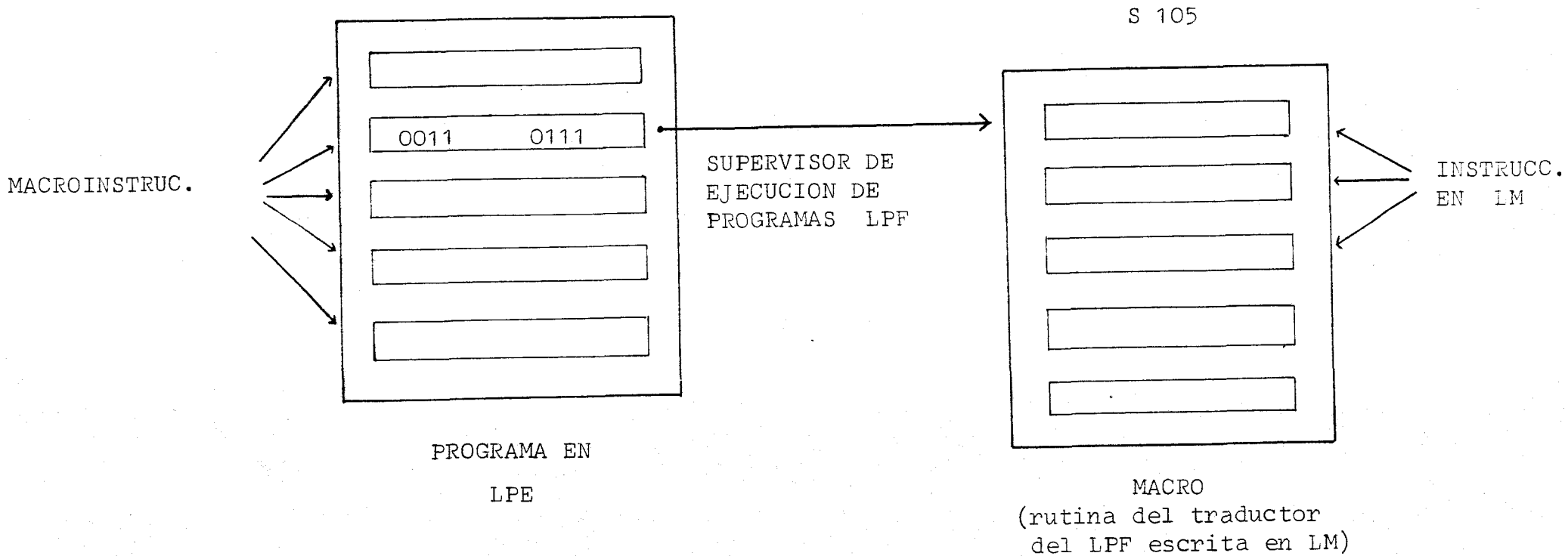


figura 1.8

El supervisor de ejecución de programas LPF hace corresponder a cada macroinstrucción una rutina del traductor escrita en lenguaje LM..

En el capítulo 4 se detallará todo lo referente a la constitución diseño y funcionamiento de estos circuitos.

1.4.2 SISTEMA OPERATIVO (S.O.)

1.4.2.1 CONSTITUCION Y GENERALIDADES:

Una de las partes fundamentales del presente trabajo consiste en la definición y construcción de un lenguaje orientado a operar con funciones (LPF) puesto que el microprocesador sólo entiende el lenguaje máquina de ordenador (LM). Es necesario, por tanto, realizar un traductor que traslade las instrucciones (macroinstrucciones) del lenguaje LPF a lenguaje LM.

El traductor se ha realizado de tal forma que a cada macroinstrucción le corresponde una rutina o macro escrita en LM (figura 1.8) La macro es la expresión en LM del algoritmo que realiza la operación que indica la instrucción; así, por ejemplo, a la macroinstrucción "integrar una función" (secc. 2.3.2) a la cual se ha asignado el código:

0011 0111

le corresponde la rutina nº S105 del traductor (secc. 3.5.2) que es la programación en LM del algoritmo de integración por la regla rectangular.

Además del traductor el S.O. contiene los programas que controlan el funcionamiento del microordenador y que constituyen el MONITOR del sistema. Los programas del monitor se utilizan mediante un "lenguaje de control" (LC), y controlan la realización de tareas tales como:

- carga de programas en memoria
- ejecución de programas escritos en LPF (programa SUPERVISOR)
- vaciados de memoria,....

Tanto el traductor del LPF como el monitor son "INTERPRETES" ya que cada instrucción de sus lenguajes da lugar a un conjunto de instrucciones en LM que se ejecutan inmediatamente. A diferencia de un compilador, un interprete no genera un programa objeto a partir de un programa fuente en unas fases previas a la fase de ejecución; en el caso presente la traducción y ejecución se realizan instrucción a instrucción en una sóla fase.

Existen además del monitor y del traductor del LPF subrutinas que son utilizadas por el propio S.O. y que el usuario no puede llamar directamente; estas subrutinas se clasifican en subrutinas del monitor, subrutinas auxiliares y subrutinas elementales.

En resumen, el S.O. o software del microordenador está constituido por:

. Programas del MONITOR
. TRADUCTOR del lenguaje orientado a operar con funciones,
. Subrutinas del monitor,
. Subrutinas auxiliares y
. Subrutinas elementales.

Todas las instrucciones de los lenguajes LM, y LC y LPF han de darse al microordenador en binario.

Las instrucciones que constituyen el S.O., al estar escritas en LM, son ejecutadas directamente por el microprocesador sin necesidad de ninguna manipulación previa y están permanentemente grabadas en la memoria del microordenador (en la memoria REPROM), no siendo necesario por lo tanto una fase de "carga del sistema". La filosofía de diseño e implementación del S.O. es completamente

similar a la utilizada en las técnicas de MICROPROGRAMACION (15), (16), (17), (18), (22), (23), (25), (28), (41), (47), (55), (59), (60), (61), (63), (67), (73), (76), (99), (100), (106), (122), (123), (124), (125), por lo que se ha hecho especial hincapié en el estudio de estas. La diferencia fundamental entre un programa del S.O. y un microprograma radica en que las instrucciones del primero son instrucciones máquina del microprocesador y son tratadas por la unidad de control de éste de una forma clásica(1); por el contrario, las del microprograma son microinstrucciones, que, en lo que estrictamente se denomina microprogramación, se ejecutan en tan sólo un ciclo base, ya que no necesitan un tratamiento o decodificación posterior: cada bit de la microinstrucción constituye directamente una señal de control.

En un sentido amplio de la palabra "microprogramación" puede decirse que el ordenador propuesto utiliza esta técnica, siendo los programas del sistema microprogramas. La mayor parte de los dispositivos de hoy día hacen uso de esta acepción amplia del concepto de microprogramación, aunque hay autores que opinan (17 p.35) que toda microinstrucción que lleva consigo un tratamiento posterior debe denominarse más estrictamente "mini-instrucción", y hablarse de "miniprogramación" y "miniprograma".

En la tabla 1.6 puede verse un resumen sobre la constitución del S.O.

---

(1) En la actualidad hay microprocesadores (serie 3000 de Intel, por ejemplo (64 p.6-53)) que utilizan internamente la técnica de la microprogramación.

sistema operativo

- **MONITOR**
  - lectura de memoria ("vaciados de memoria")
  - escritura en memoria
  - carga de programas fuentes
  - lectura de programas fuentes
  - control de traducción y ejecución de programas fuente (SUPERVISOR)
  - ejecución de programas máquina.
  - lectura del contenido de los registros del microprocesador.
- **MACROS DEL LPF**
  - rutinas operativas (producto de funciones, integral,...)
  - rutinas de bifurcación
  - rutinas de transferencia de información
  - otras (entrada/salida, STOP,...)
- **SUBRUTINAS DEL SISTEMA OPERATIVO**
  - subrutinas de mensajes
  - subrutinas de ensamblaje de instrucciones
  - subrutinas de obtención de direcc. absolutas a partir de direcciones simbólicas o relativas.
  - subrutinas de operaciones elementales.
  - subrutinas auxiliares.

Tabla 1.6
Resumen de programas que constituyen el sistema operativo.

1.4.2.2 PROGRAMAS DEL MONITOR:

El microprocesador posee una instrucción básica denominada "restart" (RST - 00AAA101) (A puede ser 0 ó 1) con la que se puede llamar a ocho subrutinas de ocho octetos cada una y que constituyen las 64 primeras palabras de la memoria del ordenador. Al ejecutarse la instrucción se transfiere el control del microprocesador a la dirección:

00 0000 00AA A000

es decir, con una sóla palabra se puede llamar a una de las ocho direcciones que se indican en la tabla 1.7

instrucc. 00AA A101	dirección absoluta	Prma. Mtor.
0000 0101	00 0000 0000 0000	PM0
0000 1101	00 0000 0000 1000	PM1
0001 0101	00 0000 0001 0000	PM2
0001 1101	00 0000 0001 1000	PM3
0010 0101	00 0000 0010 0000	PM4
0010 1101	00 0000 0010 1000	PM5
0011 0101	00 0000 0011 0000	PM6
0011 1101	00 0000 0011 1000	PM7

Tabla 1.7
instrucciones restart posibles
y direcciones a las que se transfiere el control.

La instrucción RESTART es especialmente útil para usarla en combinación con la posibilidad que ofrece la señal de control INTERRUPT, de insertar instrucciones manualmente por consola. Haciendo entrar el microprocesador en un ciclo de interrupción, con una sóla instrucción RESTART colocada manualmente en consola, se lanza uno de los ocho programas que comiencen en la dirección correspondiente indicada en la tabla 1.7.

El diseño del sistema se ha efectuado de tal forma que las instrucciones del lenguaje de control son instrucciones RESTART; esto es, cada una de las posibles instrucciones de este tipo da el control a uno de los programas del monitor.

En la última columna de la tabla 1.7 se indican los nombres de los 8 programas monitores; éstos se han grabado a partir de las direcciones que se indican en la correspondiente línea de la columna central. Para lanzar un programa monitor, por ejemplo PM7, basta con colocar en consola la instrucción indicada en la columna izquierda, 0011 1101.Como es lógico estas instrucciones son precisamente las del lenguaje LC.

En el capítulo 2 se define, de acuerdo con estas ideas, el lenguaje de control y en el capítulo 3 se explican los programas monitores.

1.4.2.3 INTERPRETE DEL LENGUAJE LPF:

A cada instrucción del lenguaje LPF se le ha hecho corresponder biunívocamente la dirección absoluta del comienzo de una macro. Logicamente, por estar el ordenador organizado en octetos, el código de una instrucción debe contener 8 bits, con lo que se pueden direccionar solamente 256 programas, o lo que es lo mismo puede haber 256 instrucciones como máximo.

El diseño del traductor se ha realizado de tal forma que el código instrucción es precisamente una dirección relativa del comienzo de la macro correspondiente; para ello se ha dividido la memoria ROM en 256 páginas de 8 bits cada una. El código de una instrucción es la dirección de comienzo de la página donde empieza el programa que ejecuta dicha instrucción.

El programa SUPERVISOR (PM7) de la ejecución de un programa fuente tiene entre otras la misión de obtener a partir del código instrucción, la dirección absoluta de comienzo de una página.

Debido a que un programa ocupa en general más de una página, 8 octetos, no todas las combinaciones posibles de 8 bits dan lugar a un código de instrucción. Como es lógico las direcciones de comienzo de programas no se han determinado después de definir los códigos de las instrucciones, sino al revés: para aprovechar al máximo la capacidad de memoria se ha realizado un estudio sobre la imbricación de programas que ha fijado la dirección de comienzo de cada macro, y esta dirección ha determinado el código instrucción correspondiente.

Es obvio que un programa no ocupará necesariamente un número de octetos múltiplo de 8; es decir, un número completo de páginas, por lo que habrá una pérdida de capacidad de memoria. Esta pérdida, se minimiza gracias a las facilidades que da el lenguaje LM de saltar de una dirección a otra de la memoria, aunque una instrucción de salto lleva consigo una pérdida de 3 octetos.

Estas ideas se llevan a la práctica, según se verá en los capitulos 2 y 3, en la implementación del SUPERVISOR y del TRADUCTOR.

1.4.3 ORGANIZACION DE MEMORIA

Los canales de información, el tratamiento de datos y la memorización están estructurados en palabras de 8 bits, Como

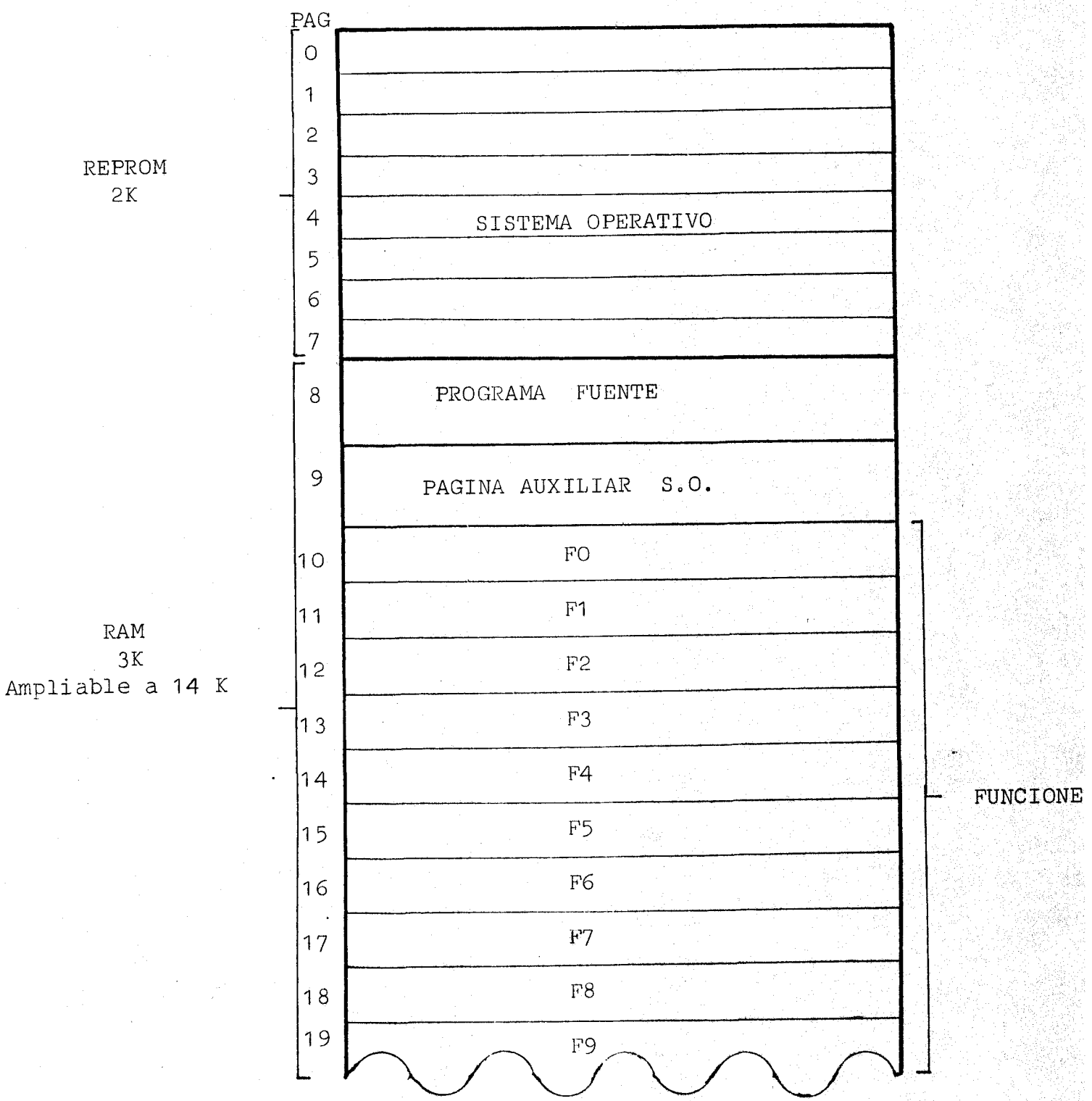


figura 1.9

contenido de la memoria trabajando con un programa LPF.

con un octeto se pueden direccionar 256 posiciones de memoria, esta última se ha configurado en páginas de 256 palabras; es decir, la memoria está constituida por 12 páginas de memoria RAM y 8 de memoria ROM; cada una de estas últimas 8 páginas a su vez, según se ha dicho en el epígrafe anterior, a efectos de direccionamiento de programas operativos, se dividen en 32 páginas de ocho octetos.

En la figura 1.9 puede verse el contenido de la memoria trabajando con un programa fuente. Utilizando el dispositivo como un microordenador de propósito general el programador puede disponer de la memoria según sus especificaciones.

La página 8, bajo el control de un programa del monitor, se ha destinado a la memorización del programa fuente; éste puede tener como máximo 256 octetos, número aceptable teniendo en cuenta la potencialidad del lenguaje y el tipo de problemas a resolver (secc. 2.3.3). Los direccionamientos dentro de un programa fuente (saltos, memorización de resultados o datos,...) se hacen con 8 bits (un octeto).

La página 9 es utilizada por el S.O., En ella se memoriza automáticamente el contenido del contador del programa fuente, se graban direcciones para anidamiento de subrutinas, se almacenan los exponentes de las funciones, etc. (secc. 3.5.1).

Las páginas 10 y sucesivas se reservan para memorizar las funciones y las constantes, bien sean datos o resultados.

La estructuración se ha realizado de forma que para hacer referencia a una función dentro de un programa fuente es necesario dar en binario la posición relativa de la página en que

empieza la función. Así, la función F3 se define como 0000 0011 y el intérprete le hace corresponder la página 13 de la memoria RAM (página 1101) este tipo de direccionamiento se denomina "simbólico". Según se verá más adelante (secc. 2.3.3) una función puede ocupar más de una página.

1.5 SINOPSIS:

En este capítulo de introducción se ha centrado el trabajo dentro del campo de la electrónica aplicada al cálculo matemático y se ha hecho referencia al tipo de funciones que se pueden procesar.

También se ha especificado las características técnicas de las que se deducen la potencialidad y limitaciones del sistema.

Finalmente se ha dado una descripción de las pates constitutivas del ordenador y expuesto la filosofía de concepción y diseño de la totalidad del sistema. Referente al S.O. se ha indicado como se lanzan y ejecutan los programas del monitor y traductor a partir de las instrucciones de los lenguajes LC y LPF.

CAPITULO II

DEFINICION Y DESCRIPCION DE LENGUAJES.
UTILIZACION DEL ORDENADOR.

2.1 INTRODUCCION:

En este capítulo se expone todo lo referente a la utilización del sistema.

En primer lugar se hace una descripción del lenguaje básico, y se definen los lenguajes operativo y de control. Para cada uno de ellos se indica:

1) los distintos formatos de las instrucciones.
2) una tabla con el repertorio de instrucciones.
3) algunas consideraciones específicas de las instrucciones más complejas.

La notación nemotécnica dada está unificada en lo posible, ya que se ha intentado llevar a la práctica las ideas expuestas por Nicoud (90).

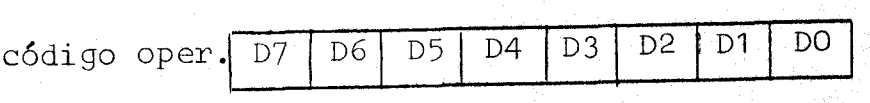
En segundo lugar se describe el funcionamiento y utilización del panel de control, y por último se indican los mensajes que puede dar el sistema operativo.

2.2 LENGUAJES MAQUINA (LM) Y ENSAMBLADOR (LE):

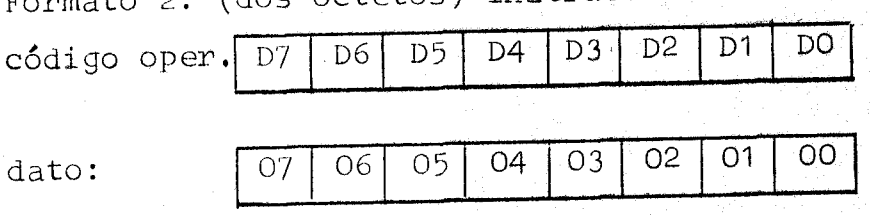
En esta sección se describen los lenguajes máquina y ensamblador del microprocesador. Estos lenguajes son de gran importancia porque además de poder ser utilizados por el programador en ellos van escritos todos los programas y rutinas del S.O..

2.2.1 FORMATO DE INSTRUCCIONES MAQUINA:

Formato 1: (un octeto), instucciones de direccionamiento implícito.



Formato 2: (dos octetos) instrucciones inmediatas:



Formato 3: (tres octetos) instrucciones de direccionamiento absoluto.

código operación:	D7	D6	D5	D4	D3	D2	D1	D0
bits menos significat.:	07	06	05	04	03	02	01	00
bits más significativo (pág)	-	-	05	04	03	02	01	00

m

2.2.2 REPERTORIO DE INSTRUCCIONES:

En la tabla 2.1 se presenta un resumen acerca del repertorio de instrucciones, en la que se incluye para cada instrucción su código de ensamblador, su código binario, su formato y la operación que se realiza (en notación Iverson).

Como se puede apreciar existen cuatro tipos fundamentales de instrucciones:

a) Instrucciones de transferencias de datos entre registros y memoria.

b) Instrucciones aritméticas y lógicas entre acumulador-registros, acumulador-memoria o acumulador-dato inmediato. Los resultados quedan almacenados siempre en el acumulador.

Tabla 2.1:

INSTRUCCIONES DE LOS LENGUAJES ENSAMBLADOR Y MAQUINA

I. TRANSFERENCIA DE DATOS:

ENSAMBLADOR	BINARIO	Fto.	OPERACION	N. IVERSON
LOAD d,s	11DD DSSS	1	transfs. registros	d <—— s
LOAD d,(HL)	11DD D111	1	lectura memoria	d <—— (H,L)
LOAD (HL),s	1111 1SSS	1	escritura en memoria	(H,L) <—— s
LOAD d,*n	OODD D110	2	escritura inmet.rtr.	d <—— n
LOAD (HL),*n	0011 1110	2	escritura inmet.mri.	(H;L) <—— n
INR d	OODD D000	1	incremento de rtro.	d <—— d+1(d≠A)
DCR d	OODD D001	1	decremento de rtro.	d <—— d-1(d≠A)

III. SALTOS:

JUMP m	01-- -100	3	salto incondicional	PC <—— m
JUMP FC,m	010P P000	3	salto condicional	F:0;(=)->(PC<-m)
JUMP FS,m	011P P000	3	salto condicional	F:1;(=)->(PC<-m)
CALL m	01-- -110	3	llamada rtna.inconn.	ST<—PC;PC<—m
CALL FC,m	010P P010	3	llamada rtna.condic.	F:0;(=)—>(PC<-m)
CALL FS,m	011P P010	3	llamada rtna.condic.	F:1;(=) —>(PC<-m)
RET	0C-- -111	1	retorno a programa	PC <—— ST
RET FC	000P P011	1	retorno condicional	F:0;(=)—>(PC<-ST)
RET FS	001P P011	1	retorno condicional	F:1;(=)->(PC<-ST)
RST L	00AA A101	1	llamada RESTART	ST<-PC;PC<-000AAA 00

IV. OTRAS INSTRUCCIONES:

INP e	0100 MMM1	1	entrada disposit. e	A <—— e
OUT s	01MM M1M1	1	salida disposti. s	s <—— A
HLT	0000 0000	1	entrada en STOP	
HLT	1111 1111	1	entrada en STOP	
NOP	1100 0000	1	no operar	

V. PSEUDOFUNCIONES:

DIR			puntos de direccionamiento simbolico.	
EQU			puntos de direccionamiento simbolico.	

Formato lenguaje Ensamblador:

1 5	6 10	11 20
etiqueta	operación	operandos

II. ARITMETICO-LOGICAS:

ADD A,s	1000 0SSS	1	adicción	A <—— A +s
ADD A,(HL)	1000 0111	1		A<—— A + (H,L)
ADD A,*n	0000 0100	2		A<—— A + n
ADDC A,s	1000 1SSS	1	adicción con "carry"	A<—— A + s +c
ADDC A,(HL)	1000 1111	1		A<—— A +(H,L)+c
ADDC A,*n	0000 1100	2		A<—— A + n + c
SUB A,s	1001 0SSS	1	resta	A<—— A - s
SUB A,(HL)	1001 0111	1		A<—— A - (H,L)
SUB A,*n	0001 0100	2		A<—— A - n
SUB A,s	1001 1SSS	1		A<—— A - s -c
SUBC A,(HL)	1001 1111	1	resta con "borrow"	A<—— A- (HL)-c
SUBC A,*n	0001 1100	2		A<—— A - n -c
AND A,s	1010 0SSS	1	intersección	A<—— A s
AND A,(HL)	1010 0111	1		A<—— A (HL)
AND A,*n	0010 0100	2		A<—— A n
XOR A,s	1010 1SSS	1	exclusive OR	A<—— A ⊕ s
XOR A,(HL)	1010 1111	1		A<—— A ⊕ (H,L)
XOR A,*n	0010 1100	2		A<—— A ⊕ n
OR A,s	1011 0SSS	1	unión	A<—— A Vs
OR A,(HL)	1011 0111	1		A<—— A V (H,L)
OR A,*n	0011 0100	2		A<—— A V n
COMP A,s	1011 1SSS	1	comparación	A : s
COMP A,(HL)	1011 1111	1		A : (H,L)
COMP A,*n	0011 1100	2		A : n
RL A	0000 0010	1	rotación a izquier.	A<—— A
RR A	0000 1010	1	rotación a derecha	A<—— A
RLC A	0001 0010	1	rotación con carry iz	(c,A)<—— (c,A)
RRC A	0001 1010	1	rotación con carry d.	(c,A)<—— (c,A)

d ó s { A 000; B 001; C 010; D 011; E 100; H 101; L 110 } DDD ó SSS

F { C 00; Z 01; S 10; P 11 } PP

e { E0 000; E1 001 } MMM

s { S0 11 000; S1 11 001; S2 11 010; S3 11 011 } MM MMM

—: indiferencias.
ST<—— PC apilar direcciones
PC: contador de progrmas.
ST: memoria pila.

c) Instrucciones de saltos: pueden realizarse bifurcaciones, llamadas a subrutinas y retornos a los programas de llamada, condicionales (de acuerdo con los contenidos de las " básculas banderas"), o no.

d) Otras instrucciones: instrucciones de E/S, entrada en STOP y no operativas.

2.2.3 ALGUNAS CONSIDERACIONES SOBRE LOS LENGUAJES

2.2.3.1 LENGUAJE MAQUINA:

Si en la instrucción LOAD d,s se hace d=s se obtiene una instrucción no operativa NOP, con la que el microprocesador no realiza ninguna operación efectiva. Esta instrucción es útil cuando se realiza un programa nuevo en el que posteriormente se piensan hacer modificaciones.

Todas las instrucciones de direccionamiento relativo o indexado utilizan como registros índices H y L.

Las básculas bandera de la unidad aritmético-lógica son afectadas por las siguientes operaciones:

básculas afectadas			
rebose	cero	signo	paridad
ADD-ADDC SUB-SUBC (1) COMP RRC-RR RLC-RL	INC-DEC ADD-ADDC SUB-SUBC AND-XOR-OR COMP	INC-DEC ADD-ADDC SUB-SUBC AND-XOR-OR COMP	INC-DEC ADD-ADDC SUB-SUBC AND-XOR-OR COMP

Tabla 2.2

(1) puesta a cero.

básculas afectadas por las distintas operaciones aritmético-lógicas.

2.2.3 LENGUAJE ENSAMBLADOR:

Es un conjunto de nemotécnicos, tanto para las funciones como para los operandos, y de convenciones que evitan tener que escribir un programa en lenguaje máquina (binario). Otra vantaja del lenguaje ensamblador sobre el máquina radica en el hecho de que en el primero se utiliza en todo momento un direccionamiento simbólico.

Se ha realizado un "ensamblador cruzado" (cross assembler) de dos fases escrito en lenguaje BASIC e implementado en un UNIVAC 1108. Este ensamblador traduce programas escritos en lenguaje ensamblador a lenguaje máquina, dando un direccionamiento binario-absoluto; Además posee una gran potencialidad en cuanto al análisis de errores lexicográficos y de direccionamiento.

En el Apéndice III pueden verse listados producidos por el ensamblador. La presentación del diseño, análisis, programación e implementación del ensamblador cruzado se efectuará en un ulterior trabajo (referencia 70 bis).

El formato general de una instrucción ensamblador es

1 5	6 10	11
etiqueta	operación	operandos

El campo etiqueta es opcional. Cuando se usa sirve para identificar simbólicamente la dirección de la instrucción en la que va.

Se han incorporado al lenguaje dos directivos (pseudo-funciones). Su finalidad es dar información al traductor, y por lo tanto no originan código alguno:

CAMPO ETIQUETA	CAM.OPERAC.	CAMPO OPERANDOS
etiqueta	EQU	punto, página
etiqueta	DIR	punto, página

En las dos pseudo-instrucciones el campo operando es una dirección de memoria escrita en decimal. En DIR la etiqueta es opcional.

La orden EQU hace corresponder al identificador en el campo etiqueta la dirección absoluta que aparece en el campo operandos. La DIR especifica la dirección absoluta base para las instrucciones que le sigan hasta la siguiente ocurrencia de una nueva orden DIR.

Se pueden insertar dentro de un programa comentarios, sin mas que escribir un asterisco (*), en la primera posición.

En el Apéndice III figuran todos los programas del S.O. escritos en lenguaje ensamblador y máquina. Se remite al lector interado a él, para una mejor compresión de la operatividad de ambos lenguajes.

2.3 LENGUAJE ORIENTADO A OPERAR CON FUNCIONES (LPF):

Con este lenguaje el programador tiene la posibilidad de efectuar un tratamiento numérico y lógico de señales analógicas.

2.3.1 FORMATOS DE INSTRUCCIONES:

Los cuatro formatos posibles son:

Formato 1: (un octeto) instrucciones de direccionamiento implícito.

código operación:

D7	D6	D5	D4	D3	D2	D1	D0

Formato 2: (dos octetos) instrucciones de direccionamiento simbólico de funciones.

código operación:

D7	D6	D5	D4	D3	D2	D1	D0

dirección simból.:

0	0	F5	F4	F3	F2	F1	F0

Fn

(de 0 a 54 en binario)

Formato 3: (dos octetos) instrucciones de direccionamiento inmediato-relativo.

código operación:

D7	D6	D5	D4	D3	D2	D1	D0

direcc. relativa:

07	06	05	04	03	02	01	00

n

Formato 4: (dos octetos) instrucciones inmediatas:

código operación:

D7	D6	D5	D4	D3	D2	D1	D0

dato:

07	06	05	04	03	02	01	00

2.3.2 REPERTORIO DE INSTRUCCIONES:

En la tabla 3.2 puede verse el repertorio de instrucciones del lenguaje LPF.

Las instrucciones se clasifican de la siguiente forma:

I. Instrucciones de transferencias de datos entre registros y memoria:
   a) operando con funciones.
   b) sin operar con funciones.

II. Instrucciones aritmético-lógicas.
   a) operando con funciones.
   b) operando sin funciones.

III. Instrucciones de salto.

IV. Otras instrucciones.

Las operaciones aritméticas y lógicas se pueden hacer de las siguientes formas:

a) A <—— A op B, entre el acumulador y el registro B

B) A <—— A op(n), entre el acumulador y el contenido de memoria dado por el segundo octeto.

C) A <—— A op n, entre el acumulador y el dato dado por el segundo octeto (operaciones inmediatas).

Tabla 3.2 INSTRUCCIONES DEL LENGUAJE LPF

I. TRANSFERENCIA DE DATOS:

NEMOTECNICO	BINARIO	FORMATO	Operación	notación Iverson
LOADF FO,Fn	0101 1000	2	transf. de fcion.	FO<— Fn
LOADF Fn,FO	0101 1010	2	transf. de fcion.	Fn<— FO
LOADF Fn,A	0110 0010	2	definic. fción. cte.	Fn<—(A,.....,A)
LOADF A,Fn(C)	0110 0101	2	obtenc. punto de fcion.	A<— Fn(C)
LOADF Fn(C),A	0110 0110	2	definic. punto fcion.	Fn(C)<— A
LOADF PNP,*n	0110 0111	4	definic. Nº pág./fcion.	PNP<— n
LOADF PNP,C	0110 1010	1	"	PNP<— C
LOADF PR,*n	0110 1001	4	definic. ralentización	PR<— n
LOADF PR,C	0110 1011	1	"	PR<— C
LOADF A,PNP	0110 1100	1	obtenc. nº pág./fcion.	A<— PNP
LOADF A,PR	0110 1101	1	obtenc. ralentización	A<— PR
LOADF XFn,C	0110 1110	2	definic. de exponente	XFn<— C
LOADF A,XFn	0110 1111	1	obtenc. de expoenente.	A<— XFn
LOAD d,s	1001 DDSS	1	transf. rtros. y m.	d<— s
LOAD d,*n	1000 DD00	4	definic. de variables.	d<— n

III. SALTOS:

JUMP AN,C	1000 0001	1		A:0;(<)-CP - C
JUMP AN,*n	1000 0101	4		A:0;(<)-CP - n
JUMP AP,C	1000 0010	1		A:0;(>)—>CP<—C
JUMP AP,*n	1000 0110	4	saltos condicionales	A:0;(>)—>CP<—n
JUMP AZ,C	1000 0011	1		A:0;(=)—>CP<—C
JUMP AZ,*n	1000 0111	4		A:0;(=)—>CP<—n
CALL C	1000 1001	1		(ST)<—PC, PC-C
CALL *n	1000 1101	4	rutinas	(ST)<—PC, PC-n
RET	1000 1010	1		CP<— (SP)
JUMP C	1000 1011	1	saltos incodicionales	CP<— C
JUMP *n	1000 1111	4		CP<— n

IV. OTRAS INSTRUCCIONES:

INPF Fn	0111 0000	2	entrada de función	Fn<— Eo
INPF FC	0111 0011	1	"	FC<— Eo
OUTF Fn	0111 0100	2	salida de función	SO<— Fn
OUTF FC	0111 0111	1	"	SO<— FC
INP e	0001 110E	1	entrada de disp.EO y E1	A<— Ee
OUT S	0001 111s	1	salida en disp. SO yS1	Ss<— A
STOP	1111 1111	1	fin de programa	STOP
ER	1111 1011	3	Intercalar rut. L.M.	

d o s	DD o SS
A	00
B	01
C	10
(C)	11

d/s

AN: acumulador negativo
AP: acumulador positivo
AZ: acumulador cero.

e { 0—conversor A/D
  { 1—consola/teletipo

s { 0—conversor D/A
  { 1—consola

II. ARITMETICO-LOGICAS:

ADDF FO,Fn	0010 0000	2	suma fción.	FO⟵ FO + Fn
SUBF FO,Fn	0010 1000	2	resta fción.	FO⟵ FO - Fn
MULF FO,Fn	0010 1111	2	multiplicación	FO⟵ FO * Fn
DIVF FO,Fn	0011 0011	2	divison.fcnes.	FO⟵ FO / Fn
INTF FO,Fn	0011 0111	2	integral indef.	FO⟵ $\int$ Fn dt
DRVF FO,Fn	0011 1100	2	derivada fción.	FO⟵ dFn/dt
RTPF Fn,Fn	0100 0110	2	rotación+en tip	(Fn(i),Fn(o),...,Fn(i))
RTNF Fn,Fn	0100 1001	2	rotación-en t.	(Fn(1),Fn(2),...,Fn(o))
TCPF Fn,Fn	0100 1100	2	traslac. + en t	(0,Fn(o),...,Fn(i))
TCNF Fn,Fn	0100 1110	2	traslac. - en t	(Fn(1),...,Fn(i),o)
MINF A,Fn	0101 0000	2	mímimo fción.	A⟵MIN(Fn(o),...,Fn(i))
MAXF A,Fn	0101 0100	2	máximo fción.	A⟵MAX(Fn(o),...,Fn(i))
DRCF Fn,Fn	0101 1110	2	desplaz. d.	XFn⟵XFn+C Fn⟵Fn/2C
DLCF Fn,Fn	0110 0000	2	desplto. izda.	XFn⟵XFn-C Fn⟵2CFn
ADD A,B		1		A⟵ A+B
ADD A,(n)	1010 00xx	3	Suma	A⟵ A+(n)
ADD A,*n		4		A⟵ A+n
SUB A,B		1		A⟵ A-B
SUB A,(n)	1010 01xx	3	Resta	A⟵ A-(n)
SUB A,*n		4		A⟵ A-n
MUL A,B		1		A⟵ A*B
MUL A,(n)	1010 10xx	3	Multipl.	A⟵ A*(n)
NUL A,*n		4		A⟵ A*n
DIV A,B		1		A⟵ A/B
DIV A,(n)	1010 11xx	3	división	A⟵ A/(n)
DIV A,*n		4		A⟵ A/n
AND A,B		1		A⟵ A B
AND A,(n)	1011 00xx	3	intersección	A⟵ A (n)
AND A,*n		4		A⟵ A n
XOR A,B		1		A⟵ A⊕B
XOR A,(n)	1011 01xx	3	unión exclusiva	A⟵ A ⊕ (n)
XOR A,*n		4		A⟵ A ⊕ n
OR A,B		1		A⟵ AVB
OR A,(n)	1011 10xx	3	unión	A⟵ AV(n)
OR A,*n		4		A⟵ AVn
DL A	1010 0011	1	desplazamiento	A⟵ A
DR A	1010 0111	1	desplto. dcha.	A⟵ A
RL A	1010 1011	1	rotación izda.	A⟵ A
RR A	1010 1111	1	rotción. dcha.	A⟵ A

xx {
- A,B — 00
- A,(n) — 01
- A,*n — 10

El utilizador del lenguaje dispone de tres registros y 256 posiciones de memoria para almacenar el programa y datos. Los registros se denominan A, B y C. C suele utilizarse como registro de indexación.

Los saltos y llamadas a subrutinas pueden realizarse a una dirección indexada (dada por el registro C) o inmediata.

Todas las instrucciones cuyo código termina en F se refieren a funciones. Fn quiere decir la función que indica el segundo octeto: n.

2.3.3 ALGUNAS CONSIDERACIONES:

. entrada y salida de funciones:

Los dispositivos de entrada y salida de funciones son EO y SO, respectivamente. Hay dos formas de leer una función.

a) punto a punto, por medio de las instrucciones INP 0 y OUT 0. Con estas instrucciones se lee o escribe sólo un punto de la función, con lo que se puede trabajar en tiempo real (ver sección 2.3.4, ejemplo c). El periodo de lectura o escritura viene dado por el tiempo de procesamiento entre dos lecturas consecutivas, siempre que el ciclo dure más de 45 microsegundos, en caso contrario, el periodo viene impuesto por el tiempo de conversión de (A/D: 45 microsegundos aproximadamente). Al trabajar en tiempo real el programador tiene que tener en cuenta los problemas asociados entre ancho de banda de la función y periodo de muestreo. (104). Aumentando el número de conversores de entrada o salida fácilmente

se podría realizar una multiplexación por programa.

b) lectura o escritura en memoria de una función completa:

Con las instrucciones INPF Fn y OUTF Fn se escribe o se lee de memoria un conjunto de puntos que define a una función.
Según se dijo en la sección 1.3 el proceso de entrada/salida puede regularse utilizando dos parámetros:

. nº de páginas utilizadas para memorizar cada función.

. velocidad de muestreo.

Con las instrucciones:

LOAD A,PNP y LOADF PNP,A

Se puede definir el número de páginas que se desea ocupen las funciones (PNP: parámetro número de páginas). Con:

LOAD A,PR y LOADF PR,A

se puede dar un parámetro de ralentización de velocidad de conversión. Para PR=0 se consigue la mayor velocidad de conversión y para PR=255 la mínima velocidad.

Los dos parámetros pueden ser redefinidos dentro de un mismo programa (secc. 2.3.3 ejemplo a). Si no se definen, la conversión se efectua a la máxima velocidad y se utiliza sólo una página de memoria por función.

Cuando las funciones ocupan más de una página el programador debe tener precaución al dar nombre a las funciones. En Fn, n es el número de la página de comienzo de la función. Si PNP=3 y se definen F0 y F2, hay un solapamiento entre los 256 últimos puntos de F0 y los 256 primeros puntos de F2 (ver secc. 1.5, figura 1.8).

. exponentes de funciones:

Los exponentes de funciones se dan con las instrucciones:

LOAD A,XFn y LOADF XFn,A

Por ejemplo, si se quiere que el exponente de la función F3 sea + 7, se puede definir así:

```
LOAD   C,*7          1000 1000
                     0000 0111

LOADF  XF7,C         0110 1110
                     0000 0111
```

. puntos de una función:

Con las instrucciones:

LOAD Fn(C),A y LOADF A,Fn(C)

pueden darse valores o leerse, respectivamente puntos dentro de una página de una función. Con estas instrucciones, además de mejorar las posibilidades operativas del lenguaje, puede aumentarse la zona de memoria de programas y datos, ya que en la página Fn podrían almacenarse datos que no fuesen necesariamente puntos de una función.

. funciones constantes:

Con la instrucción:

LOADF Fn,A

se puede definir una función constante: La función Fn toma para todos sus puntos el valor del acumulador.

. anidamientos de subrutinas:

Hay posibilidad de llamadas a subrutinas hasta con siete niveles de anidamiento.

. instrucción ER.-

Esta instrucción sirve para intercalar en un programa LPF una rutina escrita en LM, ya sea del S.O.,ya sea definida arbitraria

mente por el utilizador en memoria RAM. La rutina en LM debe acabar con RET para que una vez ejecutada, el control vuelva al supervisor del LPF. La instrucción ER tiene tres octetos, el segundo debe contener la dirección baja y el tercero la dirección alta del comienzo de la rutina LM

EJEMPLOS:

a) Leer 512 puntos de dos funciones X e Y a la mínima velocidad posible y evaluar la siguiente expresión:

$$Z = 9.\int_{0}^{t} x.dt + y^2$$

Una vez hallada la función Z:

1) En el dispositivo de salida S1 escribir:
   1) el mínimo y el máximo de la función Z
   2) los valores de t (dentro de los 100 primeros puntos) para los que la función Z es máxima.
2) Escribir la función resultado a la máxima velocidad posible. Como esta información se va a visualizar en un osciloscopio, escribirla reiterativamente de forma indefinida.

SOLUCION:

se denomina Z:F0; X:F2; Y:F4; función auxiliar : F6.

```
LOAD PNP,*2
LOAD PR,*255
INPF F2
INPF F4
INTF F0,F2
LOAD A,*9
LOADF F6,A
MULF F0,F6
LOADF F6,F0
LOADF F0,F4
```

```
        MULTF FO,F4
        ADDF  FO,F6
        MINF  A,FO
        OUT   2
        MAXF  A,FO
        OUT   2
        LOAD  B,A
        LOAD  C,*O
a       LOAD  A,FO(C)
        SUB   A,B
        JUMP  AZ  d
b       LOAD  A,C
        ADD   A,*1
        SUB   A,*100
        JUMP  AN,a
c       LOADF PR,*O
        OUTF  FO
        JUMP  c
d       LOAD  A,C
        OUT   1
        JUMP  b
```

b) se desea:

1) Memorizar una función en la primera página de memoria. Esta función se forma a partir de los puntos leidos del conversor de la siguiente forma:

   primer punto: primer valor leido.

   otros puntos: incremento (positivo o negativo)respecto al valor leido anteriormente.

2) Efectuar un <u>análisis de amplitudes</u> contando:

   nº total de incrementos de valor 0

   nº total de incrementos de valor 1

   nº total de incrementos de valor 2

.....
nº total de incrementos de valor 255

representar a la salida las dos funciones (función de incrementos y de análisis de amplitudes).

solución:

En FO se memoriza la función incremental y en F1 la de análisis de amplitudes de incrementos.

```
        LOAD  C,*0
        LOAD  B,C
a       INP   0
        LOAD  B,A
        SUB   A,(200)
        LOAD  FO(C),A
        LOAD  (200),B
        LOAD  B,C
        LOAD  C,A
        LOADF A,F1(C)
        LOAD  A,*1
        LOADF F1(C)
        LOAD  A,B
        ADD   A,*1
        JUMP  z b
        LOAD  C,A
        JUMP  a
b       OUTF  FO
        OUTF  F1
        HLT
```

Hay que tener en cuenta que el periodo de muestreo viene impuesto por el tiempo de ejecución del lazo del programa.

c) Evaluar en tiempo real la expresión:

$$Z = (X-38)^2$$

donde X es la función de entrada y Z la de salida

```
a       INP  0
        SUB  A,*38
        LOAD B,A
        MUL  A,B
        OUT  0
        JUMP a
```

número de instrucciones nemotécnicas: 6

número de octetos binarios:8

2.4 LENGUAJE DE CONTROL (LC):

El lenguaje de control tiene por objeto la utilización de los programas del MONITOR.

2.4.1 FORMATOS DE INSTRUCCIONES:

Todas las instrucciones constan de un sólo octeto.

2.4.2 REPERTORIO DE INSTRUCCIONES:

En la tabla 2.3 se indican las siete instrucciones de control que posibilitan la utilización de los programas monitores. Las instrucciones nemotécnicas van precedidas de un punto.

Al ejecutar las instrucciones

.LEER, .ESCRIBIR, .LEER PF y . ESCRIBIR PF

el sistema operativo demanda información relativa a las direcciones de comienzo de la operación a realizar (ver secc. 2.6).

NEMOTECNICO	BINARIO	OPERACION O TAREA A REALIZAR
.LEER	0000 0101	leer de memoria
.ESCRIBIR	0000 1101	escribir en memoria RAM
.LEER PF	0001 0101	leer página programa fuente
.ESCRIBIR PF	0001 1101	escribir o cargar programa F.
.EJC PM	0010 0101	ejecutar un progr. en leng.básico
.EJC PF	0011 1101	ejecutar un programa fuente
.LEER RTROS	0010 1101	lectura del contenido de los registros

Tabla 2.3
instrucciones del lenguaje de Control.

2.4.3 DESCRIPCION DE LOS OBJETIVOS DE LOS PROGRAMAS MONITORES:

- PROGRAMAS DEL MONITOR DE LECTURAS Y ESCRITURA EN MEMORIA:

Con el programa de lectura se puede leer el contenido de la memoria a partir de la dirección absoluta indicada por el operador.

Con el programa de escritura se puede grabar en cualquier zona de la memoria RAM. Caso de que se intente escribir en la memoria ROM, lo cual es físicamente imposible, el S.O. da un mensaje de error (secc. 2.6, tabla 2.4).

- PROGRAMAS DE LECTURA Y ESCRITURA DE PROGRAMAS FUENTE:

Con estos programas se puede leer y escribir en la zona de memoria asignada al programa fuente (pág. 8). El programador únicamente tiene que dar, cuando el microordenador lo solicita,

la dirección relativa dentro de su área de memoria donde desea comenzar a leer o escribir; de esta forma se puede con facilidad corregir un programa ya memorizado.

- PROGRAMAS DE EJECUCION DE PROGRAMAS MAQUINA Y FUENTE:

Estas rutinas hacen que se ejecuten los programas máquina y fuente a partir de la dirección absoluta 8/0. Caso de que no interese comenzar las ejecuciones en esta dirección se puede hacerlo en cualquier otra utilizando una instrucción JUMP.

- PROGRAMA DE LECTURA DE REGISTRO:

Este programa escribe en el dispositivo de salida S1 el contenido de los registros A,B,C,D,E y H. El contenido del registro L se pierde. El programa es especialmente útil para detectar posibles errores en un punto concreto de un programa.

- EJEMPLOS:

a) Memorizar y ejecutar un programa fuente a partir de la dirección relativa 0:

```
.ESCRIBIR PF                    0001 1101
0                               0000 0000
                                ---------
---------------
---------------  progr.Fte.     ---------
---------------                 ---------
                                0011 1101
.EJC PF
```

b) Escribir un programa máquina a partir de la dirección 11/32, posteriormente leerlo para ver si ha sido escrito en memoria correctamente, y por último ejecutarlo:

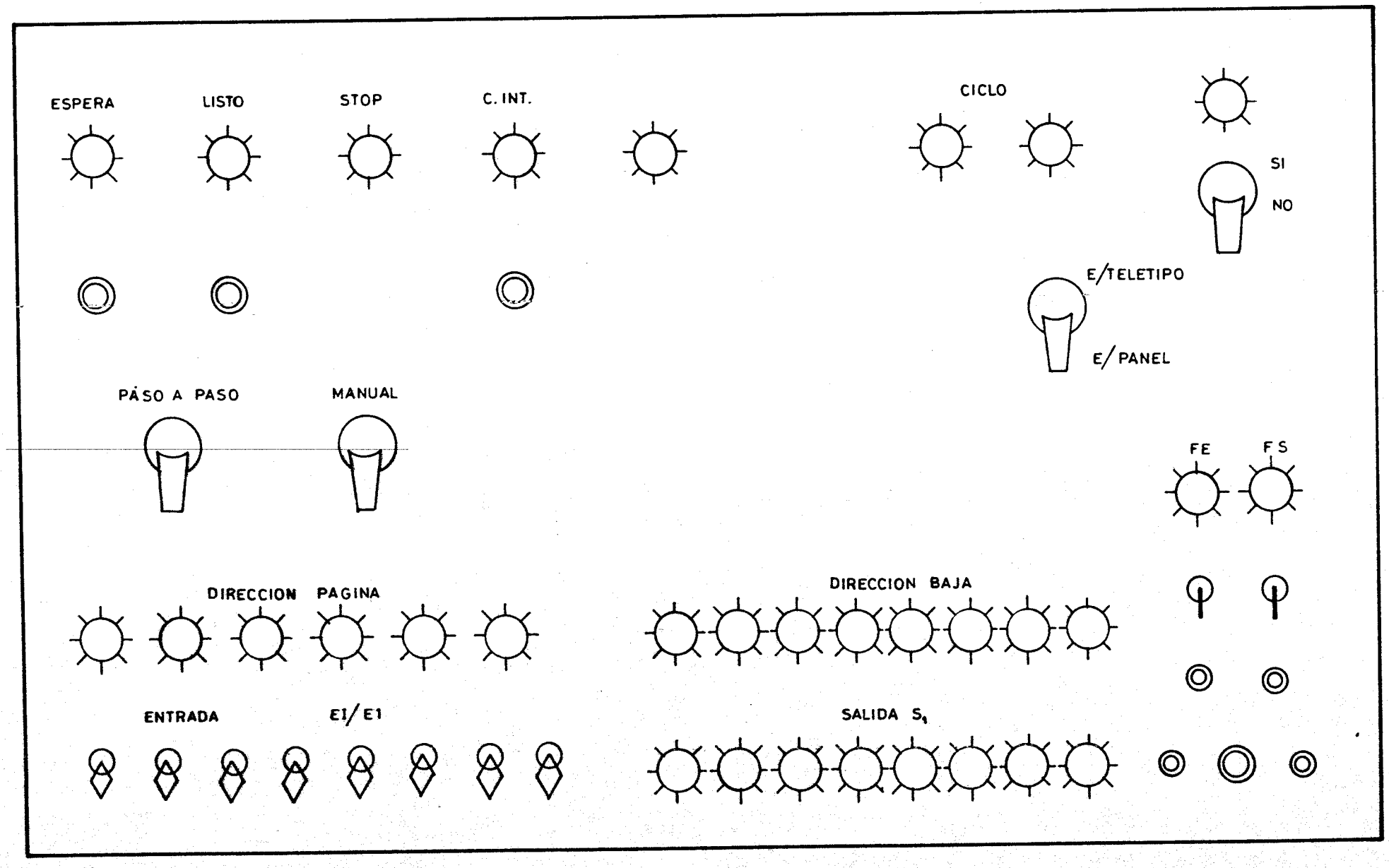


FIG. 2-1 PANEL DE CONTROL

.ESCRIBIR	0000 1101
32	0010 0000
11	0000 1011
----------	---------
---------- progr. máquina	---------
----------	---------
.LEER	0000 0101
32	0010 0000
11	0000 1011
.ESCRIBIR	0000 1101
0	0000 0000
8	0000 1000
JUMP	0100 1000
32	0010 0000
11	0000 1011
.EJC PM	0010 0101

Como se verá más adelante las instrucciones de control únicamente se pueden dar al microprocesador en ciclos de interrupción.

2.5 DESCRIPCION DE LA CONSOLA:

En la figura 2.1 se muestra un dibujo de la consola del microordenador. Está constituida por:

indicadores luminosos:

ESPERA: se enciende si el microprocesador está en estado de ESPERA.

LISTO: se enciende si el microprocesador no está en estado de ESPERA.

STOP: se enciende si el microprocesador está en estado de STOP.

C. INT: se enciende si se está ejecutando un ciclo de interrupción.

CICLO: constituido por dos indicadores que dan el contenido de los bits CH7 y CH6, es decir el ciclo que se está ejecutando:

00	PCI	(lectura de memoria)
01	PCC	(E/S)
10	PCR	(lectura de memoria)
11	PCW	(escritura en memoria)

SI/NO: se enciende cuando la tensión de la red está aplicada a las alimentaciones.

INDICADORES DE DIRECCION DE MEMORIA:

Están constituidos por dos grupos:

uno de 6 indicadores (bits CH5 y CH0)

otro de 8 indicadores (bits CL7 a CL0).

El contenido de estos indicadores tiene un significado distinto dependiendo de los ciclos:

ciclos PCI: dirección de memoria que indica el contador de programas (CH: página de memoria, CL dirección dentro de la página).

ciclos PCR: dirección de memoria donde se lee un dato.

ciclos PCW: dirección de memoria donde se escribe un dato.

ciclos PCC: en C1: contenido registro A (dato de salida)
en CH: 6 últimos bits de la instrucción de E/S que se está ejecutando.

SALIDA S1: 8 indicadores que se utilizan como dispositivo de salida S1. En ellos se visualizan los mensajes que da automáticamente el sistema operativo.

FE: indicador que se enciende al conectar las alimentaciones del conversor A/D de entrada.

FS: indicador luminoso que se enciende al conectar las alimentaciones del conversor D/A de salida.

pulsadores:

LISTO: pulsándolo, el microprocesador puede salir del estado de ESPERA.

ESPERA: pulsándolo, el microprocesador entra en estado de ESPERA.

INTERRUPCION: pulsándolo, el microordenador entra en un ciclo de interrupción; es decir, capta la instrucción depositada en los interruptores EI y la ejecuta.

interruptores:

SI/NO: conexión de alimentaciones a la red eléctrica.

PASO A PASO: este interruptor permite hacer entrar el microprocesador en ESPERA, al pulsar LISTO ejecuta el siguiente ciclo de instrucción y vuelve a entrar en ESPERA. De esta manera se puede ejecutar un programa memorizado en memoria "paso a paso", verificándose con facilidad los posibles errores del programa.

MANUAL: este interruptor posibilita que el microprocesador entre en ESPERA y ejecute ciclos de interrupción; es decir actúa como PASO A PASO pero captando las instrucciones que se van dando manualmente en los interruptores EI.

EI/E1: (Entrada ciclos Interrupción/Entrada dispositivo E1)
Con estos interruptores se dan al microordenador:

. instrucciones máquina e instrucciones de control durante los ciclos de interrupción o datos (como dispositivo de entrada E1) en el caso de que el interruptor E/PANEL-E/TELETIPO está convenientemente posicionado.

E/PANEL-E/TELETIPO: Este interruptor sirve para hacer que el dispositivo de entrada E1 esté constituido por los interruptores EI/E1 o por las básculas RS E1 (que pueden estar conexionadas a un teletipo u otro dispositivo).

FE: conecta o desconecta las alimentaciones del conversor A/D de entrada (+15V y -15V).

FS: conecta o desconecta las alimentaciones del conversor D/A de salida (+15V y -15V).

ISX: interruptor para trabajar o no con sincronismo externo.

conectores:

FE: para conectar el dispositivo o transductor que genera las Funciones de Entrada.

FS: para conectar el dispositivo que capta la Función de Salida (osciloscopio, registrador gráfico,...)

SIN IN: entrada de señal de sincronismo externo.

SIN OUT: salida de señal de sincronismo.

M: masa o tierra.

2.6 UTILIZACION Y EJEMPLOS:

Al conectar (interruptor SI/NO) a la red el microordenador, este entra indefinidamente en estado STOP. Para salir de él es necesario pulsar C. INT. El microordenador capta la instrucción depositada en los interruptores EI/E1 y la ejecuta. Esta instrucción será usualmente del lenguaje máquina o del lenguaje de control,...

Caso de que se quiera empezar a ejecutar el programa a partir de la posición 0 de la memoria ROM hay que colocar en los interruptores la instrucción:

NOP 1100 0000

esta instrucción (no operativa), debe colocarse siempre que entre el microordenador en estado de STOP y se desee que el programa continúe normalmente. Por ejemplo, si hay un "overflow" en una operación, el sistema operativo hace entrar el microordenador en STOP; si se quiere que a pesar del overflow coninúe el programa, debe darse la instrucción NOP.

Cuando se ejecutan programas el microordenador puede dar mensajes (secc. 2.7) por los indicadores S1; estos mensajes siempre van acompañados de una entrada en los estados de STOP o ESPERA. Se debe operar de la siguiente forma:

I) STOP: demanda de una instrucción máquina o de control (fin de ejecución de un programa,...). Se debe colocar en EI/E1 una instrucción de control o máquina para ejecutar otra "tarea". Para salir del estado de STOP pulsar C. INT.

II) ESPERA: el microordenador puede entrar en este estado por los siguientes motivos:

a) demanda de información acerca de direcciones de memoria, datos o instrucciones a memorizar. (ver secc. 2.7).

b) cuando hay una entrada o salida de una función. El operador debe dar al interruptor de las alimentaciones del conversor (interruptores FE/FS) y conectar el periférico de entrada/salida (en los conectores FE o FS).

ejemplo a) Escritura de un programa en lenguaje fuente (ver secc. 2.4.3 ejemplo a)

1) Poner SI/NO en posición SI. El microprocesador entra en STOP.

2) Colocar en EI/E1 la instrucción:

.ESCRIBIR PF 0001 1101

3) Pulsar C.INT: el microordenador sale de STOP.

4) Entra en ESPERA y da por S2 el mensaje 0000 0001; es decir, (secc. 2.7) solicita la dirección de memoria a partir de la cual se desea escribir,
Colocar en EI/E1: 0000 0000 y pulsar LISTO.

5) entra en ESPERA

6) Depositar en EI/E1 el octeto a memorizar. Pulsar LISTO.

7) Volver al punto 5

Cuando se finaliza de memorizar toda la información el microordenador está en el punto 5, es decir en ESPERA.

Ejemplo b) Ejecutar el programa memorizado en el ejemplo a):

1) El microordenador está en ESPERA (punto 5 del ejemplo a)

2) Colocar en EI/E1 la instrucción de control:

.EJC PF 0011 1101

3) Pulsar C.INT. El microordenador capta la instrucción dada en 2 y transfiere el control al programa supervisor del monitor ejecutándose el programa fuente.

Suponiendo que en el transcurso del programa hay, por ejemplo, un overflow el microprocesador entra en STOP y el S.O. da el mensaje de overflow 1111 0010. Para este caso, hay varias soluciones posibles, por ejemplo:

a) <u>continuar</u> el programa:

. colocar en EI/E1 la instrucción:

NOP 1100 0000

. pulsar C.INT

b) hacer otra tarea: por ejemplo, leer otro programa:

. actuar de acuerdo con el ejemplo a) a partir del punto 2.

Suponiendo que hay una función como dato de entrada: el microordenador entra en estado de ESPERA y da el siguiente mensaje:

1111 0100. (entrada de función)

Efectuar las siguientes operaciones:

1) conectar las alimentaciones del conversor A/D de entrada (dar al interruptor FS, se enciende el indicador FS).

2) conectar el dispositivo de entrada en el conector FS (generador por ejemplo)

3) pulsar LISTO, el programa continúa.

2.7 MENSAJES DEL SISTEMA OPERATIVO:

Los mensajes se dan al operador a través del dispositivo de salida S1; es decir, por indicadores luminosos del panel de control. Se pueden clasificar en tres tipos: mensajes de DEMANDA de información, de DIAGNOSTICO y de ERROR. Con las demandas de datos o direcciones entra siempre en estado de ESPERA, en los demás casos entra en STOP.

En la tabla 2.4 se recogen todos los mensajes posibles.

CODIGO	ESTADO	MENSAJE	TIPO
1111 1111	STOP	demanda de una instrucción	DEMANDA
0000 0001	ESPERA	demanda de dirección baja de m.	DEMANDA
1000 0000	ESPERA	demanda de dirección alta de m.	DEMANDA
0000 1000		inicio de carga de un programa máquina.	DEMANDA
0000 1111	STOP	se intenta leer o escribir más allá de la capacidad de memoria	ERROR
0001 1111	STOP	se intenta escribir en m ROM	ERROR
1111 0000	STOP	el dividendo es mayor en valor absoluto que el divisor.	DIAGNOSTICO
1111 0001	STOP	desbordamiento de la mantisa	DIAGNOSTICO
1111 0010	STOP	desbordamiento del exponente.	DIAGNOSTICO
1111 0100	ESPERA	comienzo de lectura de función.	DEMANDA
1111 0101	ESPERA	final de lectura de una función.	DEMANDA
1111 0110	ESPERA	comienzo de escritura de una función.	DEMANDA
1111 0111	ESPERA	final de escritura de una función.	DEMANDA
0010 1111	STOP	leer o escribir PF más hallá de la pag.	ERROR

Tabla 2.4

Relación de mensajes que puede dar el sistema operativo.

2.8 SINOPSIS:

En este capítulo se han descrito los lenguajes:

- máquina (LM)
- ensamblador (LE)
- para operar con funciones (LPF) y
- de control (LC).

También se ha explicado como utilizar en todas sus posibilidades la consola del dispositivo.

La exposición se ha acompañado de ejemplos aclaratorios de los principales puntos tratados.

Con los conceptos expuestos en este capítulo y el anterior queda completamente planteado y definido el sistema a diseñar.

CAPITULO III

ALGORITMOS Y PROGRAMAS DEL SISTEMA OPERATIVO.

3.1 INTRODUCCION:

Este capítulo presenta los aspectos relacionados con el diseño e implementación del S.O.del microordenador. En primer lugar se hacen unas consideraciones acerca de la aritmética utilizada y los algoritmos numéricos para operar con funciones. Posteriormente se presentan los organigramas y programas del sistema. Por último se indica cómo se ha imbricado la totalidad de programas con el fin de conseguir un óptimo rendimiento de la capacidad de memoria.

3.2 CONSIDERACIONES GENERALES:

3.2.1 CAPACIDAD DE CALCULO DE UN ORDENADOR, ALGORITMOS DE EVALUACION DE FUNCIONES MATEMATICAS:

Un ordenador, desde el punto de vista de cálculo, está orientado a la realización de operaciones matemáticas, ya sean aritméticas (suma, resta, producto, división,...), ya sean trascendentes (trigonométricas, logarítmicas,...).

Generalmente los circuitos de la unidad aritmética realizan sólo operaciones elementales siendo necesario para poder realizar otras funciones la construcción de algorítmos y programas utilizando las mencionadas operaciones básicas.

Existe gran cantidad de algoritmos para el cálculo del producto y la división (11), (21),(62) (40), (22), (79), (13), (14), (68), (79),. Los algoritmos para la evaluación de funciones trascendentes suelen utilizar diversos métodos (62), como desarrollos en series de potencia, residuos (29), (40), aproximaciones polinómicas, fracciones continuas (98), polinomios de Chebyschev,....

Para definir un algoritmo es necesario tener en cuenta:

1) la complejidad estructural de la unidad aritmética. A veces la organización de ésta, e incluso la de la unidad de control, puede venir impuesta por los métodos algorítmicos (60), (61), (62).

2) la rapidez operativa. Los algoritmos, para conseguir velocidades óptimas, deben ser simples y converger lo más rápidamente posible. Este factor adquiere gran importancia en sistemas que trabajan

en tiempo real: adquisición y tratamiento de datos, control de procesos,...

3) la simplicidad de implementación, para lo cual es conveniente partir de un principio unitario algorítmico para todas las operaciones a realizar, tal como se efectua en los trabajos sobre algoritmos de las referencias (60), (61), (62).

4) la elección del sistema o base de numeración. En la actualidad no sólo se utilizan bases binarias (84), sino también negativas (103), (127), u otras (1), (29), (40 , c 18).
También influye en la definición de los algoritmos la elección de coma flotante o no.

5) el control de propagación de errores (2), (82) y de la precisición deseable (19).

Un aumento de velocidad operativa puede lograrse implementando electrónicamente (es decir, por hardware) la mayor cantidad posible de funciones; esto lleva consigo el aumento de la complejidad de las unidades aritmética y de control, por lo que tradicionalmente los algoritmos se implementan por software, con lo que el tiempo de evaluación empeora. Por ejemplo, la realización de técnicas de redondeo en aritmética de coma flotante por hardware puede reducir de 1/5 a 1/10 el tiempo requerido si se llevasen a cabo por software (1). El desarrollo de la microprogramación y tecnología de memorias ROM ha logrado que se adopte una solución intermedia que consiste en "cablear" o grabar en memorias del tipo citado las rutinas algorítmicas. Este método, frontera entre el hardware y el software, es lo que se ha dado en llamar "firmware" (79 p100).

En el presente trabajo los métodos algorítmicos se han desarrollado de acuerdo con los requerimientos que impone el microprocesador elegido; es decir se han realizado para un hard ware determinado previamente. Los algoritmos se implementan, formando parte del S.O., en la memoria ROM escritos en lenguaje máquina. Los algoritmos utilizados para el producto (11) y la división (79) se han adaptado a las operaciones que puede realizar el microprocesador. Hay que mencionar que la filosofía del prototipo está orientada a la resolución de problemas característicos que aparecen al operar con señales analógicas digitalizadas, por lo que no se han implementado funciones trascendentes No obstante como es lógico, las funciones trascendentes puede implementarse mediante programa.

3.2.2 DEFINICION DE SUBRUTINAS DEL SISTEMA OPERATIVO:

Según se dijo en la sección 1.4.2.1 el S.O. está constituido por:

- programas del MINITOR
- rutinas del TRADUCTOR de programas escritos en LPF.
- subrutinas del minotor, auxiliares y elementales.

Cuando uno, o varios de los programas.No tiene un conjunto de instrucciones comunes, estas se pueden separar de dichos programas, denominados "principales", para pasar a constituir una "subrutina". La creación de subrutinas da lugar a una gran simplificación en el análisis y programación de problemas informáticos. La definición o no de una subrutina es arbitraria y depende del programador; en el presente trabajo se han seguido dos criterios, dados a continuación por orden de importancia:

1) Capacidad de memoria: La definición de una subrutina lleva consigo una economía de memoria cuando el número de octetos de la rutina a definir más el número de octetos de las instrucciones de llamada es menor que el número de octetos total en el caso de no definir la subrutina, Denominando ni el número de octetos de las instrucciones que constituirían la rutina y np el número de llamadas a la rutina caso de definirla, se tiene:

número de octetos de la rutina = (ni + 1 octeto de instrucción RET)

número de octetos de llamadas: np.3.

Para conseguir una disminución de la capacidad de memoria necesaria para almacenar los programas se ha de verificar la siguiente desigualdad:

$$3.np + ni + 1 < np.ni.$$

Con lo que np en función de ni ha de ser:

$$np > \frac{ni + 1}{ni - 1}$$

En la tabla 3.1 se presentan algunos valores de np, ni y el número de octetos ahorrados. Desde el punto de vista de economía de memoria, interesa definir una subrutina cuando el número de octetos es superior a ocho, aunque sólo sea llamada dos veces.

np	ni	octetos ahorrados
6	4	1
4	5	3
3	6	2
3	7	4
2	8	1
2	9	2
2	10	3
2	11	4
2	13	6
2	14	7

tabla 3.1
relación entre nº de octetos (ni) y nº de llamadas (np), a partir de los cuales interesa definir una rutina para economizar posiciones de memoria.

2) Tiempo: al utilizar una subrutina, el programa principal invierte, además del tiempo de ejecución de la subrutina, el tiempo de llamada (ejecución de instrucción CALL o JUMP) y de retorno (ejecución de RET). Estos tiempos aproximadamente son: (45), (66),:

JUMP y CALL: 55 microsegundos (11 estados)
RET: 25 microsegundos (5 estados)

Es decir, la ejecución de una subrutina lleva consigo, cada vez que es llamada, un tiempo adicional de 80 microsegundos.

Este tiempo no parece excesivo pero hay numerosos programas en los que las llamadas a las rutinas se hacen detro de lazos iterativos; así, por ejemplo, las rutinas de operar con funciones tienen lazos que se ejecutan por lo menos 256 veces, por lo que hacer una llamada a una subrutina dentro del lazo supone un tiempo suplementario de procesamiento de unos 20.000 microsegundos.

En la estructuración del prototipo objeto de esta investigación se ha seguido fundamentalmente el primer criterio, ya que la capacidad de memoria disponible para el S.O. es limitada (2k de memoria REPROM). No obstante, en los casos de rutinas para operar con funciones se ha tenido también en cuenta el problema de la velocidad. En el caso de lectura o escritura de una función se ha buscado que el tiempo sea el mínimo posible, ya que el tiempo de ejecución de un lazo en general es el que determina el periodo de muestreo, T.

3.2.3 ERRORES DE ENTRADA, CUANDO ESTA SE HACE A TRAVES DEL CONVERSOR A/D:

Según se dijo en la secc. 1.3 el microordenador admite la posibilidad de que las entradas de funciones se den en forma analó-

gica. Un conversor A/D realiza en la entrada un muestreo en amplitud y una digitalización. Esta conversión da lugar a unos errores, que, por haberlos que tener en cuenta, se analizan brevemente a continuación.

Los errores que se cometen en el proceso de entrada son debidos al muestreo, al conversor A/D, a la digitalización de amplitudes:

- errores de muestreo: Este error es originado por el muestreo en sí; es decir al hecho de no considerar todos los puntos de la función dentro del intervalo elegido, sino sólo algunos; por lo tanto está ligado a la frecuencia de muestreo. Para evitarlo debe aplicarse el teorema de muestreo de Shannon (9,124), (104), (105). Este teorema indica que la frecuencia de muestreo debe ser al menos el doble de la frecuencia de la componente de Fourier más elevada que forma parte de la señal a muestrear. Caso de no ser así, la función original no puede reconstruirse en su forma analógica, ya que se introduce un error imposible de eliminar con posteriores manipulaciones (filtraje,...).

- errores inherentes al conversor: son originados por los ajustes a cero falta de monotonicidad en la conversión,etc.,,, (108).

- error de cuantización o resolución: este error es ocasionado por el uso de una aritmética finita para representar valores que, por ser continuos, pueden venir dados por infinitos dígitos, El error máximo cometido es de $\pm$ 1/2 del valor del bit menos significativo. Téngase presente que al utilizar un número finito de bits para representar los números, el fondo de escala queda dividido en cuantos; el valor de un cuanto, q, es precisamente el valor del bit menos significativo. En el presente caso, 8 bits, la cota de error es:

$$\frac{q}{2} \cdot 100 = 0,39\ \%$$

Se demuestra (121) que el efecto producido por el error de cuantización es equivalente a la adición de un ruido aleatorio a la señal analógica y que (56c2) el error estadístico de un conversor A/D típico de 10 bits, considerando los errores de cuantización e inherente al conversor, oscila alrededor de un 0,24%.

Por otro lado se llega a la conclusión (108p41) de que el error cuadrático medio viene dado por la expresión:

$$\overline{E^2} = q^2/12$$

Para 8 bits se tiene:

$$\overline{E^2} = 1,3 \cdot 10^{-6}$$

Como es lógico, para disminuir el error absoluto de cuantización, una función debe estar amplificada al máximo posible dentro de los límites impuestos por el conversor.

Los algoritmos con los que opera el ordenador se han realizado de tal forma que se minimice la acumulación sistemática de errores y estén en lo posible dentro de las cotas de error de entrada que se acaban de analizar.

3.3 ARITMETICA UTILIZADA:

Antes de definir los algoritmos y programas del S.O. conviene hacer algunas consideraciones acerca de la aritmética utilizada; éste es el objetivo de esta sección.

En primer lugar, secc. 3.3.1, se especifica como se realiza la representación binaria.

La notación utilizada para las funciones es en coma flotante, secc. 3.3.2, con lo que:

- la precisión máxima que puede conseguirse queda fijada por el número de bits de la mantisa, y
- el número de bits del exponente determina, junto con el de la mantisa, el factor de escala de las funciones.

Según se dijo en el epígrafe anterior (secc. 3.2.3) es conveniente que los datos o muestras de las funciones de entrada vengan amplificadas al máximo posible con objeto de reducir la cota de error absoluto de cuantización. Esto equivale a decir que, para minimizar el error de los resultados de salida, las funciones de entrada deben estar "normalizadas", entendiendo por normalización la definición que se da en la secc. 3.3.2.2.

Los algoritmos se han elegido procurando que la distribución de errores sea insesgada y el valor de la varianza el menor posible. Los errores de cálculo, siempre que sea factible, deben caer dentro del rango del error con que vienen afectados los datos de entrada (secc. 3.2.3). Para conseguir estos objetivos se utilizan diversas técnicas de redondeo (secc. 3.3.3) y se buscan algoritmos operativos adecuados.

3.3.1 REPRESENTACION BINARIA DE LOS DATOS:

Los datos se representan en complemento a 2 en palabras de 8 bits, indicando el bit de la izquierda el signo y suponiendo el punto que separa la parte entera de la fraccionaria colo-

cado entre el bit de signo y el bit más significativo; es decir, el formato de la representación de los datos en el interior del ordenador es el siguiente:

SIGNO	D6	D5	D4	D3	D2	D1	D0

La representación en complemento a dos tiene ventajas al efectuar operaciones aritméticas (por ejemplo, la adición y sustración se hacen directamente, sea cual sea el signo de los datos, sin más que considerar el signo como el bit más significativo del dato), pero en las entradas y salidas se necesita usualmente una conversión de código para los números negativos.

La notación en complemento a 2 se utiliza en el presente trabajo por tres razones prácticas:

1) el microprocesador utilizado efectua las operaciones elementales en complemento a 2 (65), (66).

2) el conversor A/D tiene la posibilidad de representar directamente los números en complemento a 2, sin necesidad de un tratamiento externo de conversión de código (113 ,p42).

3) existen algoritmos de fácil implementación para el producto y división utilizando la notación en complemento a 2 (40),(79).

3.3.2 ALGUNAS CONSIDERACIONES A TENER EN CUENTA SOBRE LA UTILIZACION DE COMA FLOTANTE:

Para aumentar el rango de los datos y resultados, dentro del ordenador se suele utilizar una representación en "como flotante" (79) (54). En el presente trabajo esta notación se usa para

las funciones permitiendo utilizar un "factor de escala" para cada una de ellas.

3.3.2.1 REPRESENTACION:

Un número en coma flotante, base dos, se representa de la siguiente forma:

$$A = M \cdot 2^{E} \quad \text{o} \quad A = (M,E)$$

donde M es la mantisa y E es el exponente.

En el microordenador, las funciones se procesan en coma flotante. Una función, f(t), queda definida por un conjunto de valores, Fn(kT), dados por el conversor A/D y que constituyen las mantisas de la función (secc. 1.3). Por programa y con las instrucciones adecuadas (secc. 2.3.3.) se dan los exponentes o factores de escala XFn. Cada función, por lo tanto, está determinada en el memoria del ordenador por:

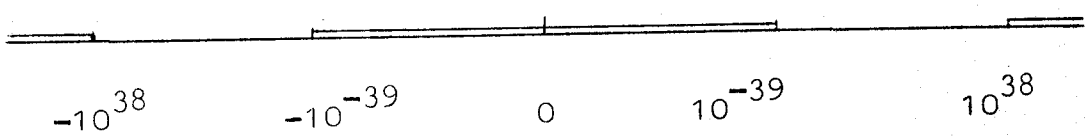
- un conjunto de mantisas : Fn(kT), donde k=0, ,2,..., y
- un exponente: XFn.

Los exponente, al igual que las mantisas, se almacenan en palabras de 8 bits, incluyendo el signo. El que tanto los exponentes como las mantisas tengan el mismo número de bits viene impuesto por el microprocesador, que está organizado en octetos. Las operaciones en coma flotante se realizan simulándolas por software y no por hardware.

Con el número de bits mencionados el rango de los números que pueden ser representados es:

- valor absoluto máximo: $\pm (2^7 - 1) \cdot 2^{(2^7 - 1)} = \pm 2{,}1 \cdot 10^{38}$

- valor absoluto mímimo: $\pm 2^0 \cdot 2^{-(2^7-1)} = \pm 5{,}8 \cdot 10^{-39}$



-10^{38} -10^{-39} 0 10^{-39} 10^{38}

La precisión que puede conseguirse con siete bits es de 3 cifras decimales.

3.3.2.2 NORMALIZACION Y ECUALIZACION:

Hay dos operaciones ligadas con la aritmética en coma flotante que se denominan normalización y ecualización.

Un número en coma flotante puede tener distintas representaciones; así, por ejemplo,

$0.0110 \cdot 2^5$ y $0.1100 \cdot 2^4$

representan al mismo número. Un número se dice normalizado cuando su exponente está ajustado de tal forma que el digito más significativo de su mantisa es:

1 para los números positivos o

0 para los números negativos(complemento a 2).

Es decir, un número está normalizado si se verifica que (79 p173):

$$D7 \oplus D6 = 1$$

Como entre funciones se opera con conjuntos de mantisas que tienen un exponente común, no tiene sentido hablar de normalización de todos los puntos de una función. Se dice que una función o página está normalizada cuando al menos uno de sus puntos lo está.

Según se dijo en la introducción a esta sección, los datos de entrada deben estar normalizados. Hay operaciones en las que, a pesar de que los datos estén normalizados, el resultado no lo está; por ejemplo:

$$\begin{array}{rl} & 0 \;.\; 1011 \;\;.\;\; 2^5 \\ - & 0 \;.\; 1001 \;\;.\;\; 2^5 \\ \hline & 0 \;.\; 0011 \;\;.\;\; 2^5 \end{array}$$

En las operaciones en que se prevea que pueda darse esta situación se debe realizar una "post-normalización".

Hay que tener en cuenta que al efectuar una normalización se introducen en la mantisa ceros no significativos. Una sencuencia de operaciones que lleven consigo normalizaciones puede dar lugar a una pérdida progresiva de significancia (40 .p375).

Ecualizar dos números quiere decir igualar sus exponentes. Esta operación se realiza sin más que efectuar desplazamiento en l mantisa; por ejemplo:

números sin ecualizar		números ecualizados
$0.000101 \;.\; 2^{14}$	=	$0.101000 \;.\; 2^{11}$
$0.001010 \;.\; 2^{11}$	=	$0.001010 \;.\; 2^{11}$

Los desplazamientos a la derecha hay que hacerlos (22 p12):

- completando con ceros, para los números positivos, y
- completando con unos, para los números negativos.

Al efectuar una ecualización en una unidad aritmética o en un microprocesador hay que tener en cuenta que:

1) se debe elevar siempre el exponente menor y no bajar el mayor, ya que en este caso se perderían los dígitos más significativos de la mantisa del número mayor.

2) si la diferencia entre los exponentes es mayor que el número de bits, n, de la mantisa, la ecualización da lugar a un "falso cero", (mantisa cero). En este caso no debe efectuarse la operación ya que el dato de menor exponente es despreciable, teniendo en cuenta el número de bits de precisión con que se opera, frente al mayor exponente; es decir, en una adición o sustración el resultado sería igual al dato de mayor exponente con igual o distinto signo:

$$A = (MA, EA)$$
$$B = (MB, EB)$$

Si $EA > EB + n$ al ecualizar resulta:

$$A = (MA, EA) \text{ y}$$
$$B = (0, EA),$$

con lo que las cifras significativas de B se pierden.
Los algoritmos, en este caso, deben dar el resultado directamente, sin efectuar ecualización previa:

$$A + B = A$$
$$A - B = A$$
$$B + A = A$$
$$B - A = -A$$

3.3.2.3 OPERACIONES BASICAS EN COMA FLOTANTE:

Sean:

$A = (AM, AE)$, $B=(BM, BE)$ y $R= (RM, RE)$

tres números expresados en coma flotante.

Para SUMAR o RESTAR A y B hay que seguir los siguientes pasos:

1) ecualizar A y B; es decir hacer AE=BE
2) sumar o restar las mantisas: $RM = AM \pm BM$

3) el exponente del resultado es: RE = AE = BE

4) efectuar una post-normalización.

Para MULTIPLICAR o DIVIDIR A y B, hay que seguir los siguientes pasos:

1) multiplicar o dividir las mantisas:

RM = AM . BM o RM = AM/BM

2) sumar o restar los exponentes:

RE = AE + BE o RE = AE - BE

3) efectuar una post-normalización.

3.3.3 REDONDEOS:

El resultado de efectuar determinadas operaciones, como el producto o división, puede ocupar un número de bits mayor que cada uno de los datos; dentro del ordenador, sin embargo, siempre se representan con el mismo número de bits. Si se efectua un truncamiento, el máximo error que se da es de (40 p241):

$$\frac{2^{(-7+E)}}{2} = 2^{E-8}$$

donde E es la parte exponencial del número.

Para minimizar este error hay que preveer sistemas de redondeo que den una aproximación al número lo más aceptable posible.

Los sistemas de redondeo dentro de la aritmética de ordenadores son numerosos (1), (22), (50), (60), (79); los más utilizados por su fácil implementación son:

1) sumar (restar, en el caso de números negativos) un 1 en el bit más significativo que no se conserva. Este redondeo es equivalente al que se efectua usualmente con números decimales. Con este método (22 p50) se obtiene una varianza de $0,29 \cdot 2^{E-7}$.

2) hacer siempre igual a 1 (0, para los números negativos) el bit menos significativo a conservar, ya sea este 0 o 1. Estadísticamente se logra una varianza de $0.58.2^{E-7}$.

Al tener el primer método una varianza menor, según se dijo en la introducción a esta sección, 3.3, es más adecuado que el segundo; no obstante, es más fácil llevar a la práctica el segundo. En efecto, si, por ejemplo, el ordenador está organizado en octetos, el primer método implica que la unidad aritmética pueda efectuar adiciones y sustraciones con palabras de 9 bits, además, este primer método da lugar a una disminución de velocidad cuando la suma o resta da lugar a arrastres.

El criterio que se ha seguido en el microordenador es utilizar el segundo método, salvo en los casos en que pueda llevarse fácilmente a la práctica el primero.

3.4 ALGORITMOS NUMERICOS PARA OPERAR CON FUNCIONES:

Según se estableció en las secciones 1.3 y 3.3.2.1 una función queda representada para su tratamiento por un conjunto de puntos. Es decir, realizar una operación entre funciones es realizar un proceso iterativo con conjuntos de valores. En esta sección se exponen los métodos algorítmicos que se han seguido para las operaciones básicas entre funciones.

El número de puntos memorizados de una función es 256, o un múltiplo de 256. Para simplificar la notación, a lo largo de esta sección se consideran solamente 256 puntos.

Sean:

$F \equiv (FO, F1, F2, \ldots, F255)$, XF

$G \equiv (Go, G1, G2, \ldots, G255)$, XG y

$R \equiv (RO, R1, R2, \ldots, R255)$, XR

tres funciones definidas en la memoria del microordenador. Fi, Gi, y Ri representan las mantisas y XF, XG y XR los exponentes de cada función. F y G se consideran como datos y R como resultado.

Las operaciones con funciones que se definen en el microordenador son:

- ADICION: R <----- F + G

Esquemáticamente los pasos a realizar son:

(1) ecualizar F y G; es decir, hacer XF = XG.

(2) hacer $R \equiv (FO+GO, F1+G1, F2+G2, \ldots, F255+G255)$, y XR = XF = XG.

- SUSTRACCION: R <----- F - G

Los pasos a seguir son:

(1) ecualizar F y G; es decir, hacer XF=XG

(2) hacer R = (FO-GO, F1-G1, F2-G2, ... , F255-G255), y XR = XF = XG.

- MULTIPLICACION: R <----- F.G

Las operaciones a realizar son:

R = (FO.GO, F1.G1, ... , F255.G255), y

XR = XF + XG.

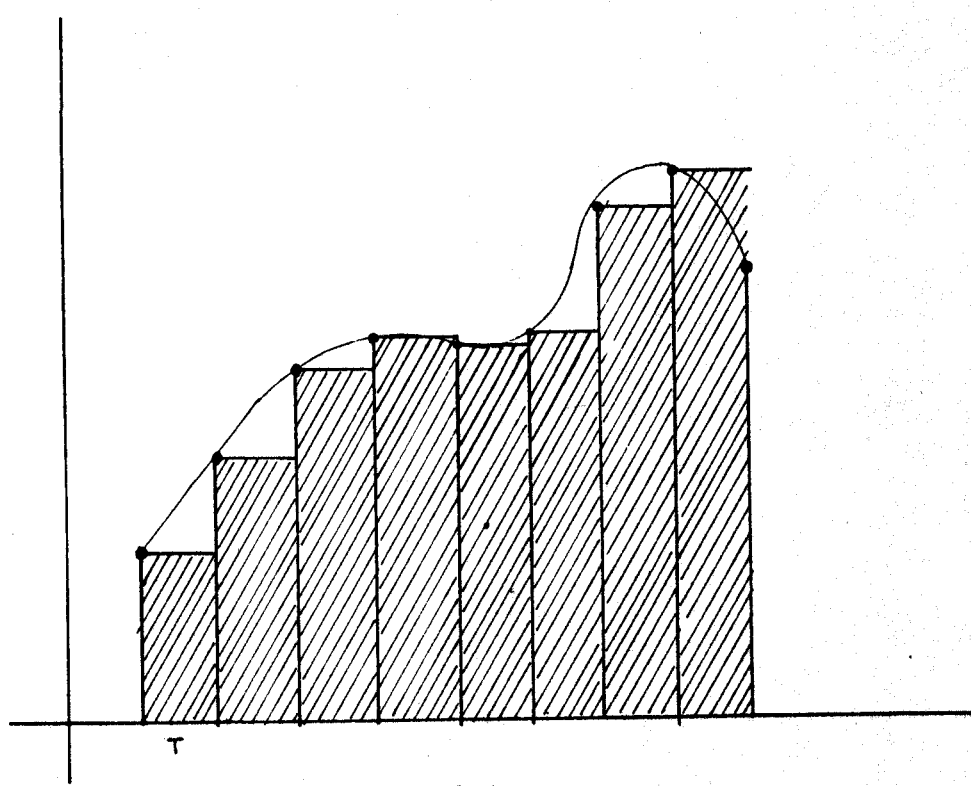


FIG. 3-1

INTEGRACION NUMERICA POR EL METODO DE LA REGLA RECTANGULAR

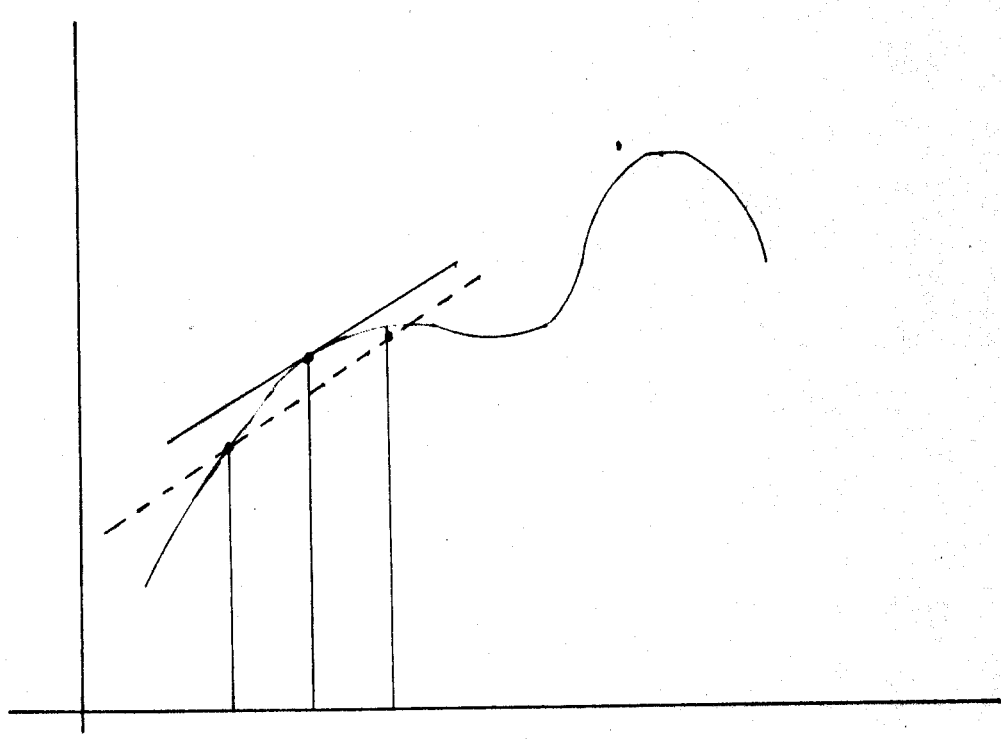


FIG. 3-2

DERIVACION NUMERICA POR EL METODO DE LAS DIFERENCIAS CENTRALES

- DIVISION: R <------ F/G

Las operaciones a realizar son:

(1) ajustar los exponentes de tal forma que cada uno de los puntos de F sea menor en valor absoluto que su correspondiente de G; es decir, se ha de verificar:

$$|F0| < |G0| \ , \ |F1| < |G1| \ , \ \ldots \ , \ |F255| < |G255|$$

(2) calcular R = (F0/G0, F1/G1, F2/G2, ... , F255/G255), y XR = XF - XG.

- INTEGRAL INDEFINIDA: R <------ $\int F.dt$

Entre los diversos métodos de integración numérica (69)(74)(83), como son las reglas rectangular, trapezoidal, de Simpson,... se ha utilizado el primer método por su facilidad de implementación y mayor rapidez comparado con los demás. No obstante este método da lugar a un mayor error por truncamiento.

De la figura 3.1 se deduce que:

R = (R0, R1, R2, ... , R255). T. donde:

R0 = 0

R1 = F0

R2 = R0 + F1

- .....

R255 = R254 + F254.

Se demuestra (83) que una cota del error debida al truncamiento (superficie no considerada debajo de la función), es, para cada punto k de la función integral:

$$E_T \leq T. \ \frac{F(k)-F(0)}{2}$$

- DERIVADA DE UNA FUNCION: $R \longleftarrow \frac{df}{dt}$

Haciendo un desarrollo de Taylor para la función F en un entorno de los puntos F(KT+T) y F/KT-T), y tomando sólo los tres primeros términos se tiene:

$$F(KT+T) = F(K) + F'(KT).T + \frac{F''(KT)}{2}$$

$$F(KT-T) = F(K) - F'(KT).T + \frac{F''(KT)}{2}$$

restando las dos expresiones anteriores, se obtiene para la derivada de la función en un punto KT=i:

$$F'(i) = \frac{F(i+T) - F(i - T)}{2.T}$$

esta expresión, cuyo significado geométrico puede verse en la figura 3.2, da la derivada de la función para cualquier punto i. El método se denomina de las "primeras diferencias centrales" (69 p325). Para el primer punto de la función se toma:

$$F'(0) = \frac{F(1) - F(0)}{T}$$

que se denomina "primera diferencia hacia adelante"; y para el último punto se toma "la primera diferencia hacia atrás:

$$F'(255) = \frac{F(255) - F(254)}{T}$$

Se tiene por lo tanto que la función derivada es:

$$R = \left(F1-FO,\ \frac{F2-FO}{2},\ \frac{F3-F1}{2},\ \ldots,\ \frac{F255-F253}{2},F255-F254\right)\frac{1}{T}$$

- ROTACION EN EL TIEMPO:

Esta operación tiene por objeto trasladar en el tiempo un número determinado de posiciones (dado por el contenido del registro C), los puntos de una función, pasando los últimos valores del intervalo memorizado a los primeros, o viceversa, dependiendo de que la rotación sea positiva o negativa, respectivamente.

(a) rotación positiva (C = 2, por ejemplo):

R = (F254, F255, FO, F1, ... , F253)

(b) rotación negativa (C = 2, por ejemplo):

R = (F2, F3, ... , F255, FO, F1)

- TRASLACION EN EL TIEMPO:

Esta operación tiene por objeto trasladar en el tiempo un número de posiciones determinado por el contenido del registro C, los puntos de una función completando las posiciones que queden libres del comienzo o final del intervalo de la función memorizada con ceros.

(a) traslación positiva R(t) ---> F(t+CT)

R=(0, 0, FO, F1, ... , F253)

(b) traslación negativa R(t) ---> F(t-CT)

R=(F2, F3, ... , F255, 0, 0).

- MAXIMO DE UNA FUNCION: A <----- máx (FO, F1, F2, ...,F255)

Para obtener el máximo de una función se siguen los siguientes pasos:

(1) hacer A igual al mínimo valor posible (A=1000 0000)
(2) hacer i = 0
(3) comparar A con F(i)
(4) si F(i) es mayor que A hacer A= F(i)
(5) si i es igual a 255 ir a (7)
(6) hacer i = i + 1 e ir a (3)
(7) fin, en A queda almacenado el máximo absoluto de la función F.

- MINIMO DE UNA FUNCION: A <------ mín (FO, F1, F2, F3,...,F255)

Se siguen los pasos siguientes:

(1) hacer A igual al máximo valor posible (A=0111 1111)
(2) hacer i = 0
(3) comparar A con F(i)
(4) si F(i) es menor que A hacer A=F(i)
(5) si i es igual a 255 ir a (7)
(6) hacer i = i + 1 e ir a (3)
(7) fin, en A queda almacenado el mínimo absoluto de la función F.

3.5 REALIZACION DE ALGORITMOS Y PROGRAMAS QUE CONSTITUYEN EL SISTEMA OPERATIVO:

En esta sección se presentan los algoritmos y programas que constituyen el sistema operativo. Cada algoritmo se representa en un organigrama donde, en general, cada bloque corresponde a una instrucción máquina del ordenador. Se ha hecho uso del lenguaje Iverson (apendice 1), de tal forma que el

programa es compresible directamente. Los puntos más dificiles de entender son aclarados aparte del organigrama.

Previamente a la descripción de los algoritmos en la sección 3.5.3, se indica en el siguiente epígrafe la finalidad de la página de memoria auxiliar del S.O..

3.5.1 PAGINA AUXILIAR DEL SISTEMA OPERATIVO:

Según se dijo en la sección 1.5.3, la página 9 de memoria se utiliza como zona de memoria auxiliar del S.O..

En la tabla 3.2 se indica el contenido y direcciones de la información que se memoriza en la página 9 bajo el control del S.O..

Las 8 primeras posiciones de memoria se reservan para memorizar direcciones de programa fuente que posibilitan la llamada y anidamiento de rutinas. En CP(0) se memoriza la dirección relativa del programa operativo.

Con la instrucción CALL "dirección" se efectua la siguiente operación:

CP(i+1) <----- CP(i) y CP(0) <---- "dirección"

donde i toma consecutivamente los valores i= 6,5,4, ... , 0 el contenido de CP(7) se pierde; es decir, sólo son posibles 7 niveles de anidamiento.

Con la instrucción RET se efectua la siguiente operación:

CP(i) <---- CP(i+1)

DIRECCION DECIMAL	CONTENIDO	
0	CP(7)	direcciones de anidamientos de subrutinas.
1	CP(6)	
2	CP(5)	
3	CP(4)	
4	CP(3)	
5	CP(2)	
6	CP(1)	
7	CP(0)	contador de programa
8	PNP	parámetro Nº de págnas./función.
9	PR	parámetro de ralentización.
10	XFO	exponentes de funciones
11	XF1	
12	XF2	
13	XF3	
14	XF4	
15	XF5	
16	XF6	
17	XF7	
.	.	
.	.	
.	.	
63	XF52	
64	RB2	registros salvados
65	RC2	
66	RD2	
67	RE2	
68		
69		
70		
71		
72	RA1	registros salvados
73	RB1	
74	RC1	
75	RD1	
76	RE1	
77	JUMP	
78	dirección baja de salto	
79	dirección alta de salto a describir por el supervisor para cada macro.	
80		
.		
.		

Tabla 3.2
Contenido de la página auxiliar
del sistema operativo (pág. 9)

donde i toma consecutivamente los valores i=0,1, ... , 6

En otras posiciones de memoria de la página 9 se almacenan los siguiente parámetros:

PNP (Nº págs./función)
PR (parámetro de ralentización) y
XFn (exponentes de las funciones)

Algunas rutinas del S.O. para operar más eficazmente necesitan casi todos los registros del microprocesador. Como de la ejecución de una macroinstrucción a la siguiente deben conservarse los contenidos de algunos de los registros se ha previsto en el comienzo de estas rutinas salvar el contenido de los registros memorizándolos en la página 9. Una vez ejecutado el programa el contenido inicial de los registros es obtenido de la citada página 9 de memoria. Estos procesos de "salvamento" y "restitución" de los registros se hacen con la ayuda de subrutinas.

- subrutina SE11: salva el contenido de los registros B,C,D, E, en las posiciones de memoria RB2, RC2, RD2 y RE2 respectivamente (ver tabla 3.2).
- subrutina SE12: recupera el contenido inicial de los registros B,C,D,E de las posiciones RB2, RC2, RD2, y RE2.
- subrutina SE9: salva el contenido de los registros A,B,C,D y E en las posiciones de memoria: RA1, RB1, RC1, RD1 y RE1 respectivamente.
- subrutina SE10: recupera el contenido inicial de A,B,C,D y E, a partir del contenido de las posiciones de memoria RA1, RB1, Rc1, RD1 y RE1, respectivamente.

El supervisor (secc. 1.4.2.3) debe dar un salto a la dirección de comienzo del programa correspondiente a la macroinstrucción en ejecución. Evidentemente esta dirección de bifurcación es una variable (para cada macroinstrucción hay una dirección distinta), por lo que no es posible efectuar directamente

el salto ya que según se vio en la secc. 2.2 en el lenguaje LM sólo existen instrucciones de bifurcación con direccionamiento absoluto. Este problema lo soluciona el supervisor escribiendo en la dirección 9/77 una instrucción de salto y transfiriendo posteriormente el control a esta dirección fija. De esta forma indirecta se ejecuta bifurcación a la dirección variable de comienzo de la rutina del S.O..

En resumen, la página 9 tiene por objeto:

(1) memorizar direcciones de tal forma que se permita el anidamiento de subrutinas.

(2) memorizar la dirección relativa de la instrucción que se está ejecutando (contador de programa).

(3) memorizar distintos parámetros del programa (PNP, PR y exponentes de funciones)

(4) salvar, memorizando en ella, el contenido de los registros al ejecutar macroinstrucciones en las que puedan ser destruidos.

(5) escribir en ella una instrucción de salto que posibilite la bifurcación a los programas del S.O. correspondientes a las macroinstrucciones.

3.5.2 RELACION DE PROGRAMAS DEL SISTEMA OPERATIVO:

A cada programa del S.O. se le da un nombre codificado de acuerdo con los siguientes criterios:

PMx programas del monitor
SMx subrutinas del monitor
SExx subrutinas elementales.
SAxx subrutinas auxiliares.
Sxxx rutinas operativas.

xx representa un número de orden.

Si una rutina está constituida por dos o más programas, estos se designan con la misma notación seguida de letras mayúsculas (ejemplo: S101, S101A, S101B).

Las direcciones de bifurcación o entrada dentro de una rutina se expecifican con el código del programa seguido de una letra minúscula (S101a, S101b,...)

Si un programa se corta por razones de imbricación, falta de capacidad,... su continuación se indica con el mismo nombre del programa precedido de la letra s (sPMO, sS123).

Si se dan varios saltos en el mismo programa se indica con la letra s seguida de un número de orden y a continuación el nombre del programa (s1PM7, s2PM7,...).

Las tablas siguientes se refieren a:

tabla 3.3 programas monitores.
tabla 3.4 rutinas elementales.
tabla 3.5 rutinas auxiliares.
tabla 3.6 rutinas operativas.

En cada tabla y para cada programa además del nombre codificado, se especifica la función que realiza, y dirección absoluta de comienzo. En las rutinas elementales y auxiliares se indican además los parámetros o registros de entrada, los de salida, y los intermedios. Esto hay que tenerlo muy en cuenta a la hora de la realización de programas ya que la información de determinados registros (A,B, y C) debe ser consistente

con la definición de los lenguajes dada en el capitulo anterior.

La consulta de las tablas mencionadas es de una gran utilidad para comprender los organigramas de los programas, ya que en la mayor parte de ellos existen numerosas llamadas a otras rutinas.

RUTINAS DEL MONITOR

NOMBRE	OPERACION	DIRECC.
PM0	Leer de memoria	0/0
PM1	escribir en memoria RAM	0/8
PM2	leer zona de memoria del programa fuente.	0/16
PM3	escribir o cargar programa fuente.	0/24
PM4	ejecutar programa máquina	0/32
PM5	lectura del contenido de los registros.	0/56
PM7	ejecutar programa fuente	0/40

SUBRUTINAS DEL MONITOR

NOMBRE	OPERACION	DIRECC.
SM1	inicialización de rutinas PM0 y PM1.	0/169
SM2	incremento de dirección de memoria.	0/185
SM3	detectar desbordamiento de memoria.	0/191
SM4	incremento de dirección en el interior pág. 8	0/199

Tabla 3.3

relación de rutinas y subrutinas

del monitor.

NOMBRE	OPERACION	DIRECCION	ENTRADA	SALIDA	INTERMEDIOS
SE1	multiplicación de dos números (A=B.C)	5/211	B,C	A	H,L,C,D,E
SE2	división de dos números (A=B/C)	6/0	B,C	A	H,L,D,B
SE3	suma de dos números (A=A + C)	7/114	A,C	A	
SE4	drción. absoluta a partir código.instrucción.	6/35	E		H,L,D
SE5	resta de dos números (A=A - C)	7/172	A,C	A	
SE6	ensamblaje de macroinstrucción (L←operador)	6/57		L	H,E
SE7	avance de punto de función (i=i+1)	4/162	D,L	D,L y c	H,A
SE8	retroceso de punto de función (i=i-1)	4/244	E,H,L	H,L y c	A
SE9	salvar registros A,B,C, y D	6/68	A,B,C,D		H,L
SE10	obtener registros A,B,C y D	8/80		A,B,C,D	H,L
SE11	salvar registros B,C,D y E	6/92	B,C,D,E		H,L
SE12	obtener registros B,C,D y E	4/110		B,C,D,E	H,L
SE13	mensaje de exponente overflow.	0/234			A
SE14	mensaje de mantisa overflow o underflow.	0/226			A
SE15	ensamblaje y obtción. drción. absoluta función.	1/57		E,L	H
SE17	desplazamiento con signo a dcha.	7/234	A		

Tabla 3.4
relación de subrutinas elementales

NOMBRE	OPERACION	DIRECC.	ENTRADA	SALIDA	INTERNOS
SA1	normalización de una función FE	3/191	E		A,B,C,D,L,H
SA2	ecualización entre FO y FE	6/102	E		A,B,C,D,L,H
SA3	salva de rgtros. y obtención direcc.absoluta de función.	3/220	A,B,C,D	L,E	H
SA4	C ←---- FE(i) A ←---- FO(i)	2/246	E,D	A,C,H,L	
SA5	B ←---- XFE A ←---- XFO	2/65	E	A,B	H,L
SA6	A ←---- FE(i)	0/250	E,L,D	A	H
SA7	L ←---- (m) (m: operando de macro)	2/185		L	H,L
SA9	anidamiento de direcciones en llamadas sub-rutinas.	4/202			E,D,L,H
SA10	desplazamiento de FE,C lugares a Dcha.XFE ←-- XFE+C.	6/130	C,E		A,C,D,L,H
SA11	desplazam.de FE,C lugares a Izda.XFE←XFE-C	6/159	C,E		A,C,D,H,L,
SA12	traslac.negativa de FE una posición.	0/205	E		A,D,H,L
SA13	desplazam.para que mantisas FO<FE en v.absol.	6/211	E		A,B,C,D,H,L
SA14	entrada de función con ralentización.	7/7	H,B,C,D,L		A,C,D,E,H,L
SA15	entrada de función sin ralentización.	7/24	C,D,H,L		A,C,D,H,L
SA16	inicialización rutinas E/S de funciones.	7/36	L		B,C,D,H,L,Z
SA17	salida de función con ralentización.	7/51	B,C,D,H,L		A,C,D,E,H,L
SA18	salida de función sin ralentización.	7/68	C,D,H,L		A,C,D,H,L
SA20	FO ←-- FE XFO ←-- XFE	7/80	E		A,B,H,L,D,C
SA21	traslación positiva de FE una posición.	7/98	E		A,B;D,H,L

Tabla 3.5
Relación de Subrutinas auxiliares.

tabla 3.6 RUTINAS OPERATIVAS

NOMBR.	OPERACION	DIRECC.
S000	EXM	5/216
S001	ADD A,B	5/0
S002	ADD A,(n)	5/8
S003	ADD A,*n	5/16
S004	SUB A,B	5/32
S005	SUB A,(n)	5/40
S006	SUB A,*n	5/48
S007	MUL A,B	5/64
S008	MUL A,(n)	5/72
S009	MUL A,*n	5/80
S010	DIV A,B	5/96
S011	DIV A,(n)	5/104
S012	DIV A,*n	5/112
S013	AND A,B	5/128
S014	AND A,(n)	5/136
S015	AND A,*n	5/144
S016	XOR A,B	5/160
S017	XOR A,(n)	5/168
S018	XOR A,*n	5/176
S019	OR A,B	5/192
S020	OR A,(n)	5/200
S021	OR A,*n	5/208
S022	DL A	5/24
S023	RL A	5/88
S025	OR A	5/56
S026	RR A	5/120
S051	LOAD A,*n	4/0
S052	LOAD A,B	4/136
S053	LOAD A,C	4/144
S054	LOAD A,(B)	4/152
S055	LOAD B,*n	4/32
S056	LOAD B,A	4/160
S057	LOAD D,C	4/76
S058	LOAD B,(B)	4/184
S059	LOAD C,*n	4/64
S060	LOAD C,A	4/191
S061	LOAD C,B	4/200
S062	LOAD C,(B)	4/216
S063	LOAD (C),*n	4/96
S064	LOAD (C),A	4/224
S065	LOAD (C),B	4/232
S066	LOAD (C),C	4/240

NOMBR.	OPERACION	DIRECC.
S081	JUMP C	4/88
S082	JUMP N	4/120
S083	JUMP AN C	4/8
S084	JUMP AN n	4/40
S085	JUMP AP C	4/16
S086	JUMP AP n	4/48
S087	JUMP AZ C	4/24
S088	JUMP AZ n	4/56
S091	CALL C	4/72
S092	CALL n	4/104
S093	RET	4/80
S101	ADD FO,Fn	1/0
S102	SUBF FO,Fn	1/64
S103	MULF FO,Fn	1/120
S104	DIVF FO,Fn	1/152
S105	INTF FO,Fn	1/184
S106	DRVF FO,Fn	1/224
S107	RTPF Fn,Fn	2/48
S108	RTNF Fn,Fn	2/72
S109	TCPF Fn,Fn	2/96
S110	TCNF Fn,Fn	2/112
S111	MINF A,Fn	2/128
S112	MAXF A,Fn	2/160
S120	NF FN	3/248
S121	LOADF FO,Fn	2/192
S122	LOADF Fn,FO	2/208
S123	DRF Fn,Fn	2/240
S124	DLF Fn,Fn	3/0
S125	LOADF Fn,A	3/16
S126	LOADF A,Fn(C)	3/40
S127	LOADF Fn(C),A	3/48
S130	LOADF PNP,*n	3/56
S131	LOADF PNP,C	3/80
S132	LOADF PR,*n	3/72
S133	LOADF PR,C	3/88
S134	LOADF A,PNP	3/96
S135	LOADF A,PR	3/104
S136	LOADF XFn,C	3/112
S137	LOADF A,XFn	3/120
S150	INPF FB	3/128
S151	INPF FC	3/152
S152	OUTF Fn	3/160
S153	OUTF FC	3/184
S999	HLT	7/248
S094	INP 0	0/224
S095	INP 1	0/232
S096	OUT 0	0/240
S097	OUT 1	0/248

3.5.3 ORGANIGRAMAS. PROGRAMACION DEL S.O.

3.5.3.1 PROGRAMAS Y SUBRUTINAS DEL MONITOR:

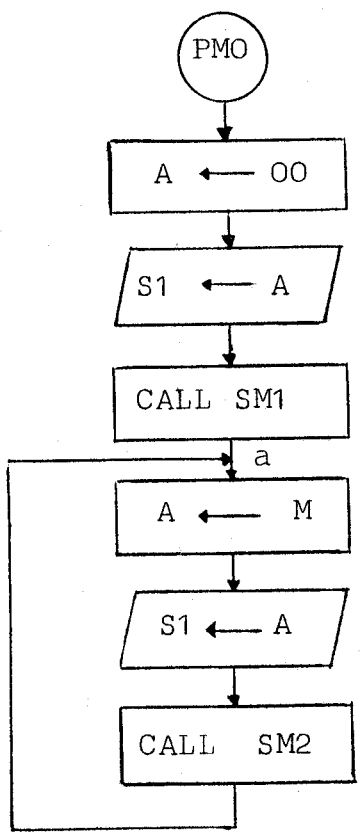
NOMBRE: PMO

OBJETIVO: leer de memoria a partir de una dirección dada.

SUBRUTINAS QUE UTILIZA: SM1 y SM2

CODIGO: Nemotécnico: .LEER
Binario: 0000 0101

DIRECCION DE ENTRADA: 0/0



EXPLICACION: Los registros H y L contienen la dirección de memoria a leer. Estos registros se inicializan externamente por medio de la subrutina SM1. El lazo de iteración contiene las instrucciones de lectura de memoria, escritura en el dispositivo de salida S1 e incremento de la dirección de memoria; esta última operación la realiza la subrutina SM2.

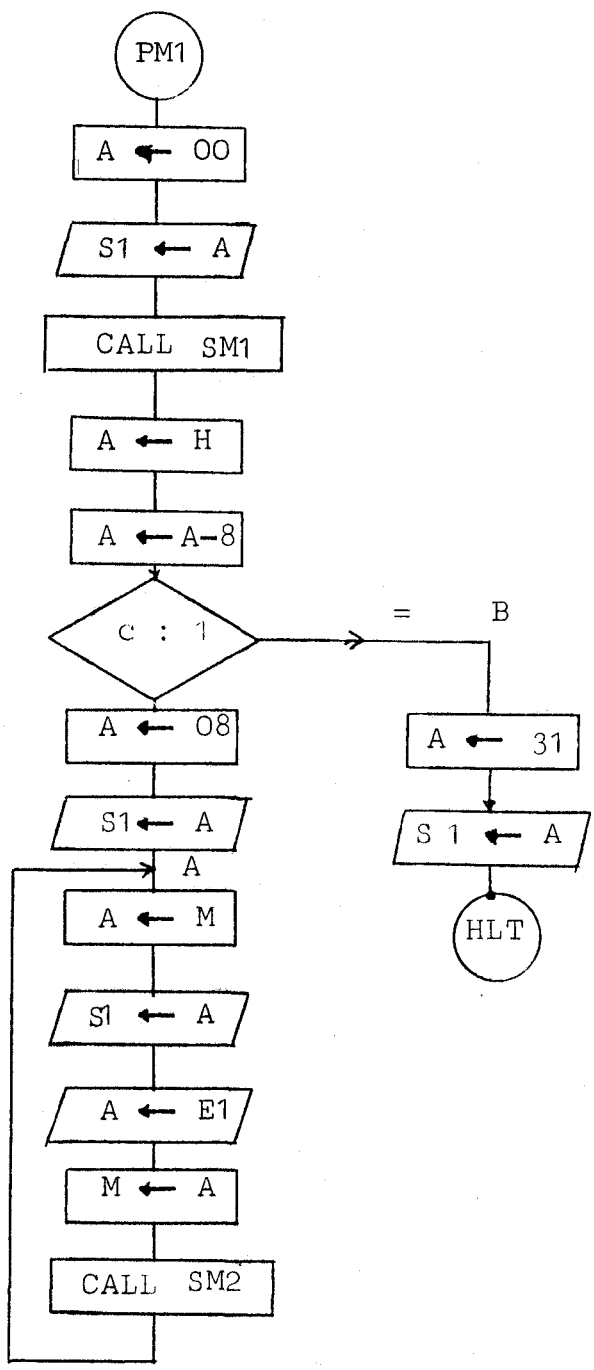
NOMBRE: PM1

OBJETIVO: escribir en memoria RAM a partir de una dirección dada.

SUBRUTINAS QUE UTILIZA: SM1 y SM2

CODIGO: Nemotécnico: .ESCRIBIR
Binario: 0000 1101

DIRECCION DE ENTRADA: 0/8



EXPLICACION: Esta rutina funciona de forma análoga a PMO. Se prevee, antes de entrar en el lazo de iteración, que por error se intente escribir dentro de las 8 primeras páginas (ROM), en este caso el S.O. da un mensaje entrado el sistema en STOP.

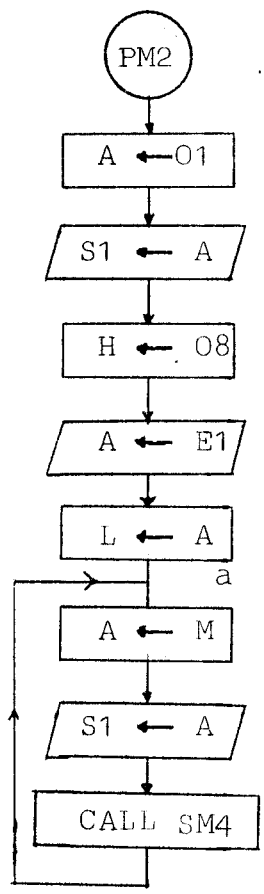
NOMBRE: PM2

OBJETIVO: leer en la zona de memoria RAM, reservada a la escritura de los programas LPF

SUBRUTINAS QUE UTILIZA: SM4

CODIGO: Nemotécnico: .LEER PF
Binario: 00010101

DIRECCION DE ENTRADA: 0/16



EXPLICACION: Este programa es similar a PMO, salvo que la lectura se efectúa en la página 8 de memoria.

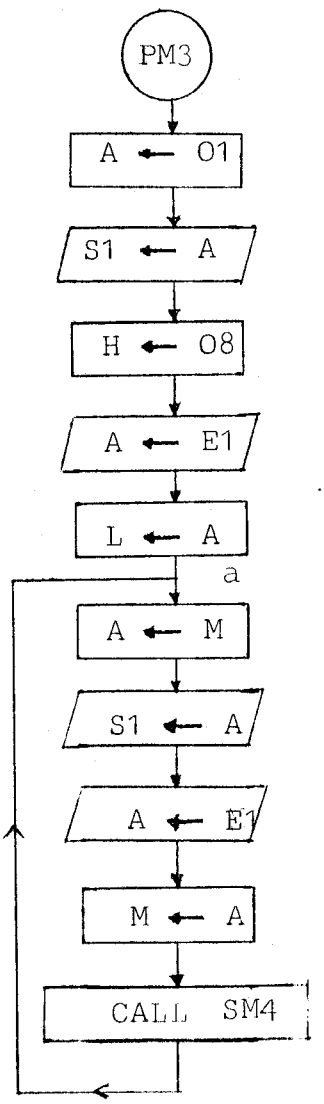
NOMBRE: PM3

OBJETIVO: escribir a partir de una posición dada, en la zona de memoria reservada a almacenar los programas fuentes ("RUTINA DE CARGA")

SUBRUTINAS QUE UTILIZA: SM4

CODIGO: Nemotécnico: .ESCRIBIR LPF
Binario: 0001 1101

DIRECCION DE ENTRADA: 0/24



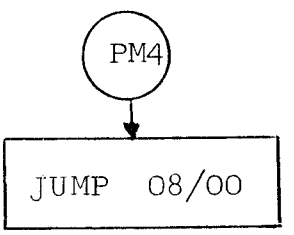
NOMBRE: PM4

OBJETIVO: ejecución de un programa escrito en lenguaje máquina.

SUBRUTINAS QUE UTILIZA:

CODIGO: Nemotécnico: .EJC PM
Binario: 0010 0101

DIRECCION DE ENTRADA: 0/32



EXPLICACION: El programa efectua una bifurcación a la dirección 8/0, y por tanto ejecuta las instrucciones máquina memorizadas a partir de ella.

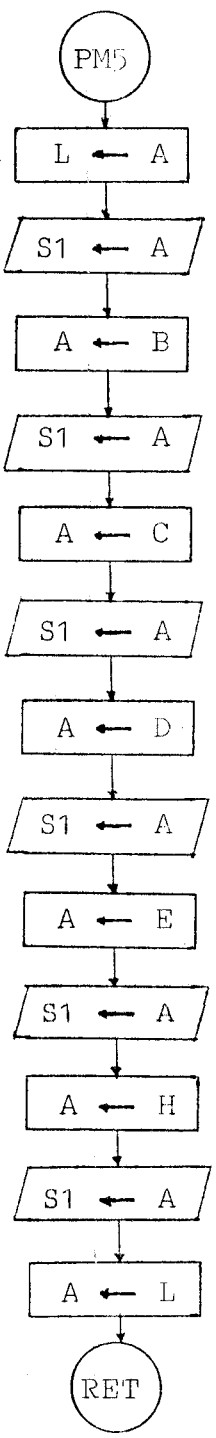
NOMBRE: PM5

OBJETIVO: visualizar en el dispositivo de salida S1 el contenido de los registros A, B, C, D y H del microordenador.

SUBRUTINAS QUE UTILIZA:

CODIGO: Nemotécnico: .LEER RTROS
Binario: 0010 1101

DIRECCION DE ENTRADA: 0/40



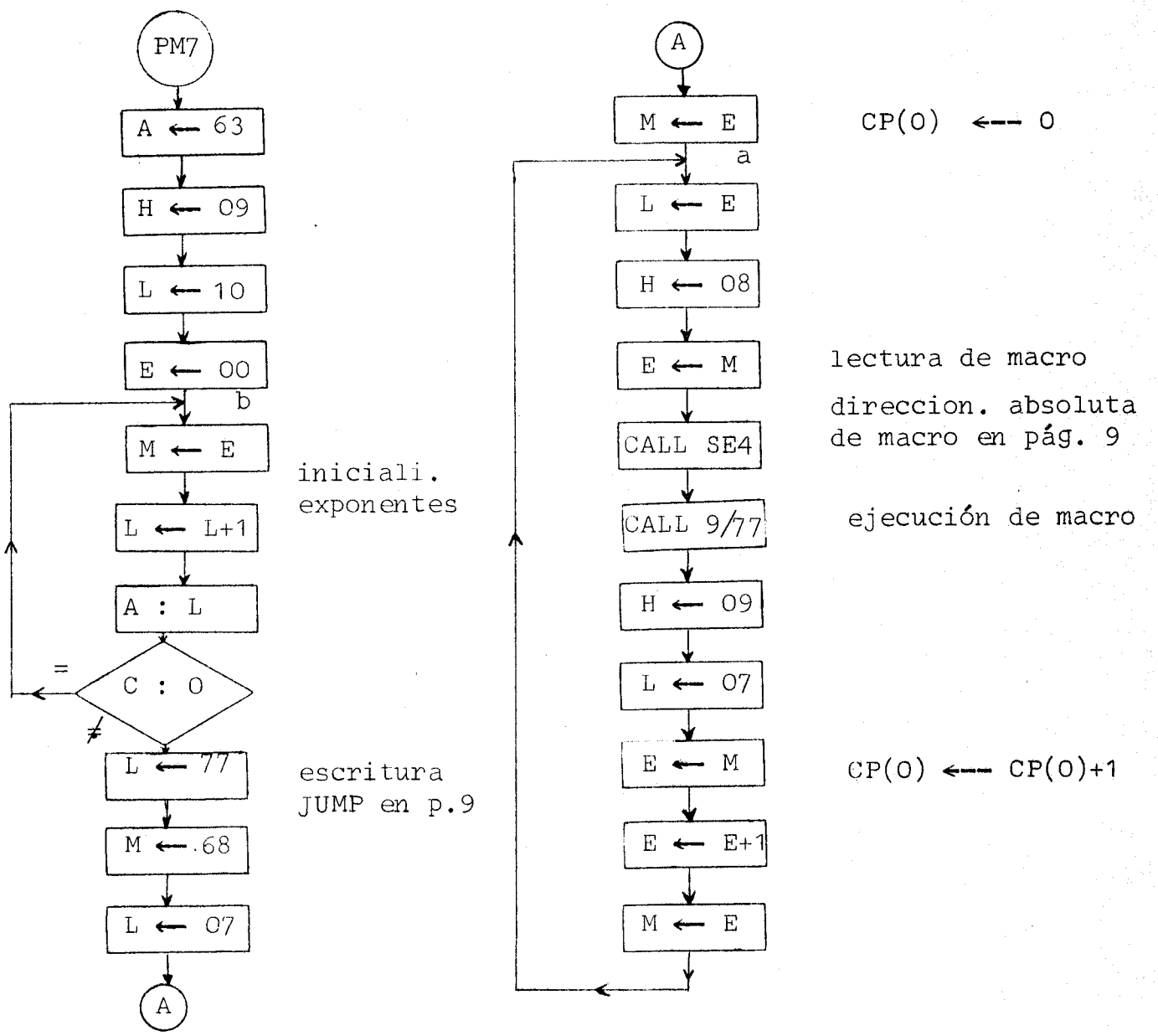
NOMBRE: PM7

OBJETIVO: controlar la ejecución de un programa fuente (PROGRAMA SUPERVISOR)

SUBRUTINAS QUE UTILIZA: SE4

CODIGO: Nemotécnico: .EJC PF
Binario: 0011 1101

DIRECCION DE ENTRADA:



EXPLICACION:

El programa supervisor realiza las siguientes operaciones:

(1) inicializa los exponentes de las funciones a cero.
(2) escribe una instrucción de salto en la dirección 9/77 (secc. 3.5.1).
(3) pone a cero el contador de programa: CP(0) = 0.
(4) lee de la dirección 8/CP(0) de memoria el código de la macroinstrucción a ejecutar.
(5) llama a la rutina SE4 que calcula a partir del código-instrucción la dirección absoluta y la escribe en las posiciones de memoria 9/78 y 9/79, completando la instrucción de JUMP del punto 2.
(6) se transfiere el control, por medio de una instrucción CALL, a la dirección 9/77 y se ejecuta la macroinstrucción, una vez acabada la ejecución el control vuelve al programa supervisor, y
(7) se lee de 9/07 el valor del contador de programa, CP(0), incrementandolo en 1, y se efectua un salto al punto 4.

Obsérvese que el programa utiliza los registros E,H y L, es decir, su contenido no se conserva de macroinstrucción a macroinstrucción. Este hecho es consistente con las características del lenguaje operativo (capítulo 2).

SUBRUTINAS DEL MONITOR

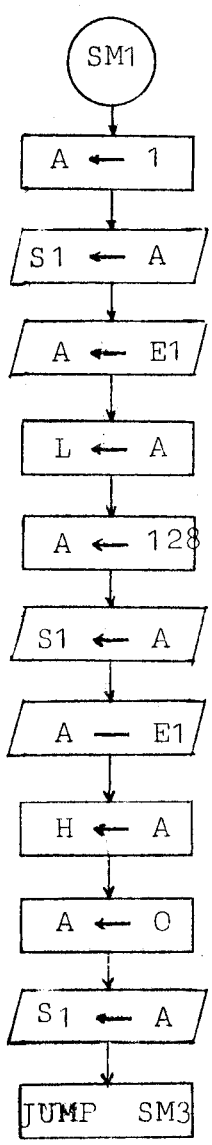
NOMBRE: SM1

OBJETIVO: controla, dando los mensajes oportunos, la lectura de las direcciones iniciales de lectura o escritura en memoria, necesarias para los programas PM0 y PM1.

SUBRUTINAS QUE UTILIZA: SM3

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 0/169



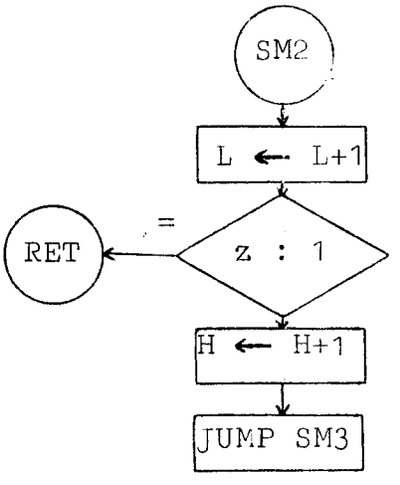
NOMBRE: SM2

OBJETIVO: incrementa en 1 la dirección de memoria contenida en los registros H y L.

SUBRUTINAS QUE UTILIZA: SM3

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 0/185



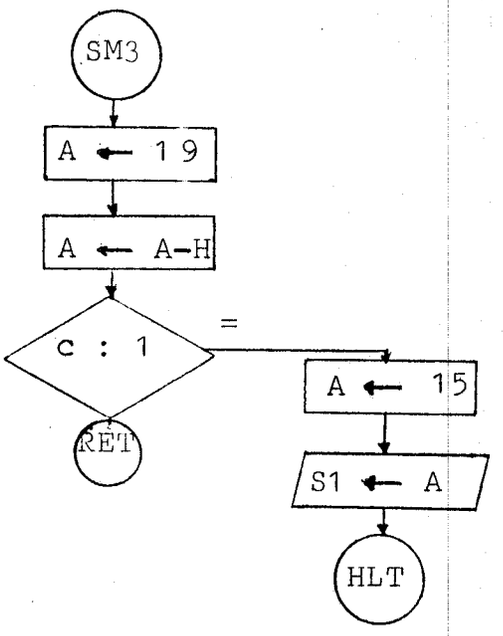
NOMBRE: SM3

OBJETIVO: detecta si hay desbordamiento de la capacidad de memoria: intentar leer o escribir a partir de la página 20 de memoria.

SUBRUTINAS QUE UTILIZA:

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 0/191



NOMBRE: SM4

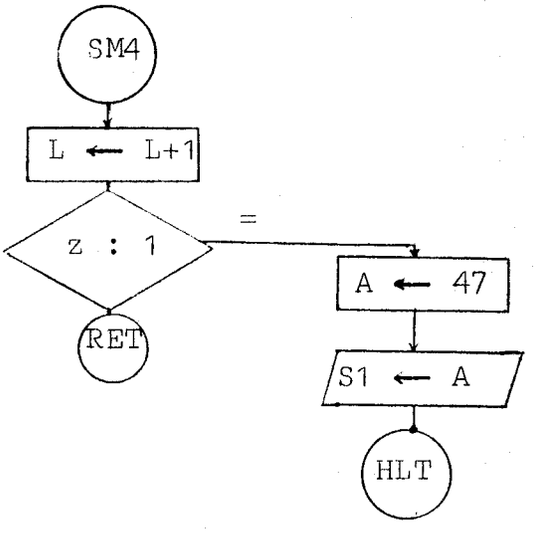
OBJETIVO: incremento de dirección relativa dentro de la zona de memoria asignada para almacenamiento de programas fuentes.

SUBRUTINAS QUE UTILIZA:

CODIGO: Nemotécnico:

Binario:

DIRECCION DE ENTRADA: 0/199



3.5.3.2 SUBRUTINAS ELEMENTALES:

NOMBRE: SE1

OBJETIVO: multiplicar dos números: A ←— BxC

SUBRUTINAS QUE UTILIZA: SE11, SE12 y SE17

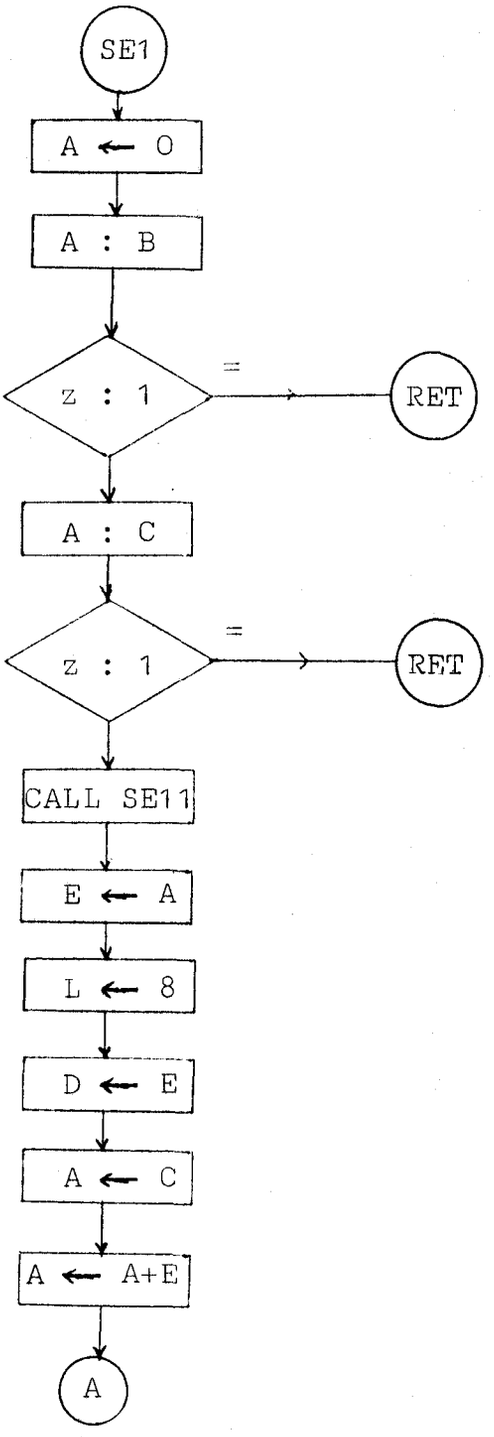
CODIGO DE LLAMADA: 5/D3

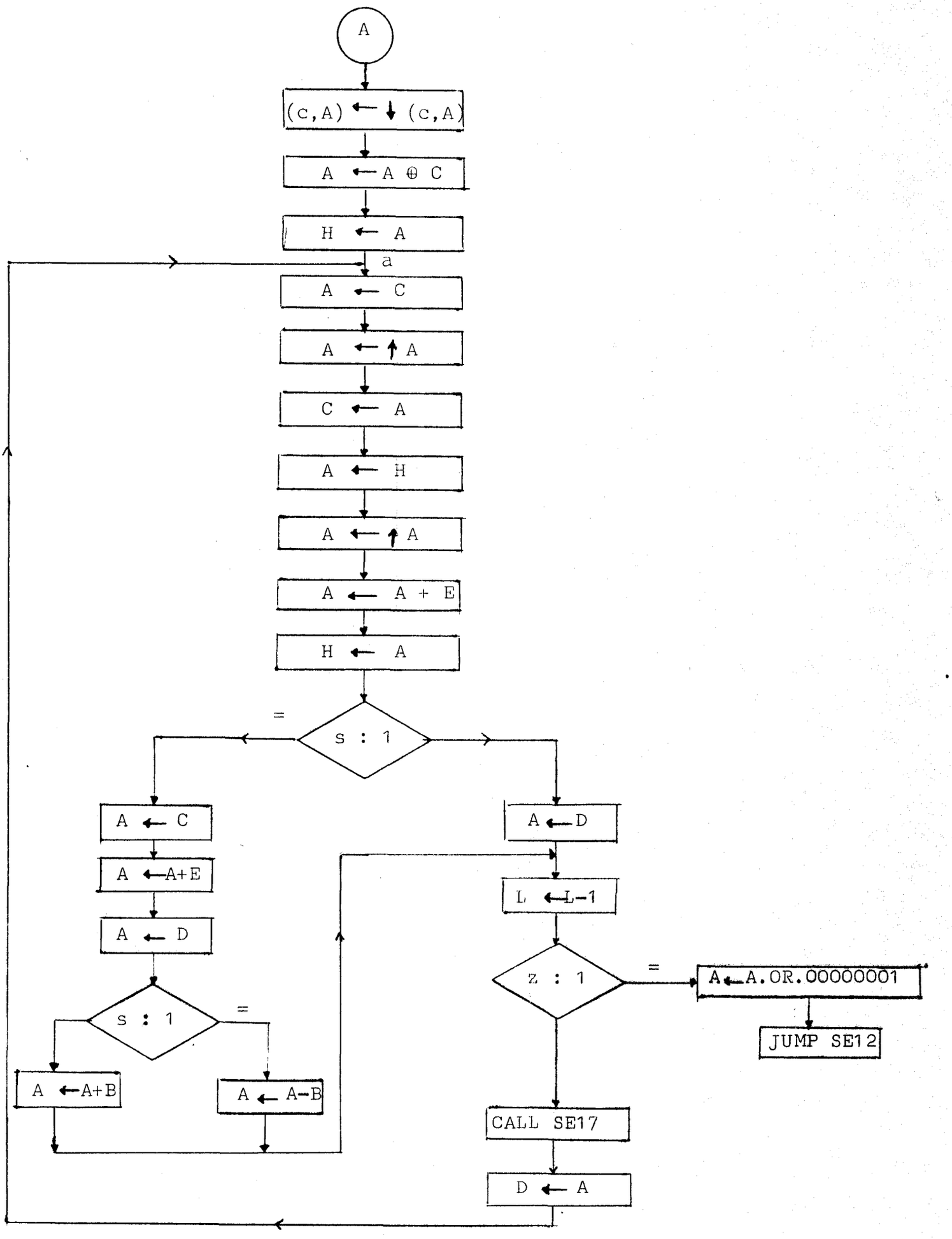
DIRECCION DE ENTRADA: 5/211

PARAMETROS DE ENTRADA: B y C

PARAMETROS DE SALIDA: A

PARAMETROS INTERMEDIOS: C, D, E, H y L




A
(c,A) ← ↓ (c,A)
A ← A ⊕ C
H ← A
a
A ← C
A ← ↑ A
C ← A
A ← H
A ← ↑ A
A ← A + E
H ← A
=
s : 1
A ← C
A ← A+E
A ← D
s : 1
=
A ← A+B
A ← A−B
A ← D
L ← L−1
z : 1
=
A ← A.OR.00000001
JUMP SE12
CALL SE17
D ← A

EXPLICACION:

El programa esta basado en la adaptación del algoritmo de BOOTH (11), (22), (79), a las operaciones que puede realizar el microprocesador.

En un registro, D, se almacenan los productos parciales; al final del proceso en D queda memorizado el producto.

Para multiplicar dos números:

$$r = r_n r_{n-1} r_{n-2} \ldots r_1 r_o, \quad y$$

$$s = s_n s_{n-1} s_{n-2} \ldots s_1 s_o$$

se examinan los digitos de posición i-ésima (r_i) y (i-1)-ésima (r_{i-1}) de r. Inicialmente se toma i=0, r_i=0, r_i=0 y D=0. Posteriormente i toma iterativamente los valores 1,2,...,n

(I) si r_i=0, r_{i-1}=0, multiplicar el producto parcial por 2^{-1}; es decir, efectuar un desplazamiento a la derecha.

(II) si r_i=0 y r_{i-1}=1, sumar s al producto parcial y multiplicar por 2^{-1}.

(III) si r_i=1 y r_{i-1}=1, sumar s al producto parcial y multiplicar por 2^{-1}.

(IV) si r_i=1 y r_{i-1}=1, multiplicar por 2^{-1} el producto parcial.

(V) no multiplicar por 2^{-1} en el paso iterativo i = n.

Para llevar a la práctica este algoritmo inicialmente hace H<--- r $\oplus$ r; es decir, un bit H_i de H es cero si r_i y r_{i-1} son iguales; en este caso, según el algoritmo (pasos I y IV), no es necesario efectuar suma o resta alguna. Si por el contrario, H es 1 quiere decir que $r_i \neq r_{i-1}$, en este caso si

r_i= 1 se resta a D el multiplicador , B(paso III), y si r_i=0 sumar B a D (paso II), Siempre, excepto cuando i=n, hay que efectuar después un desplazamiento a la derecha, Finalmente el resultado se redondea haciendo el bit menos significativo igual a 1. En L se memoriza i, es decir el paso iterativo.

NOMBRE: SE2

OBJETIVO: división de dos números: D ←— B/C

SUBRUTINAS QUE UTILIZA: SE11, SE12 y SE14

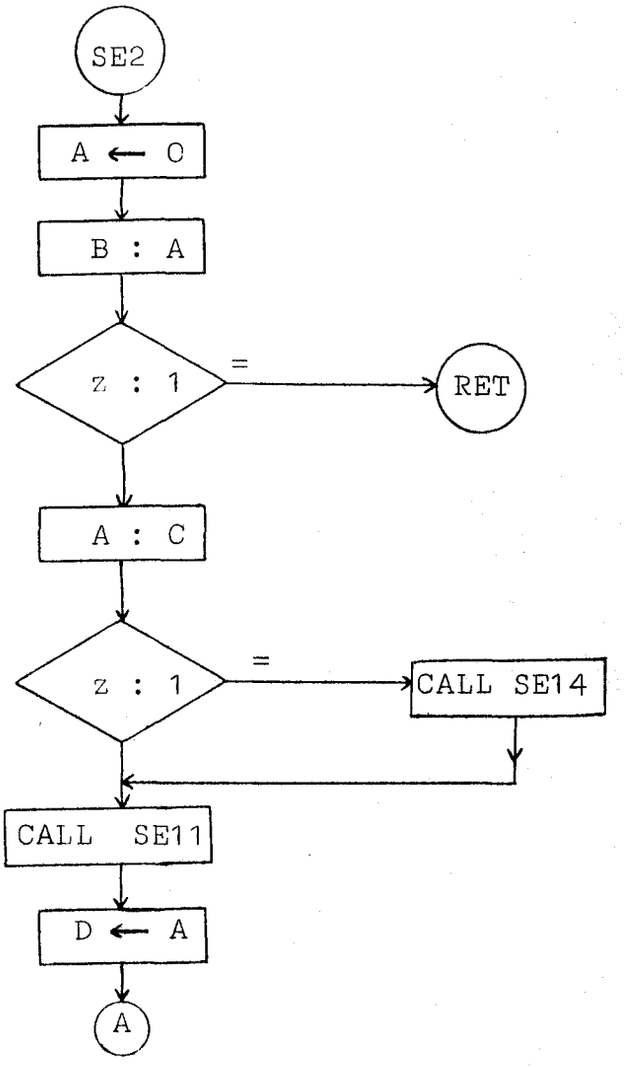
CODIGO DE LLAMADA: 6/00

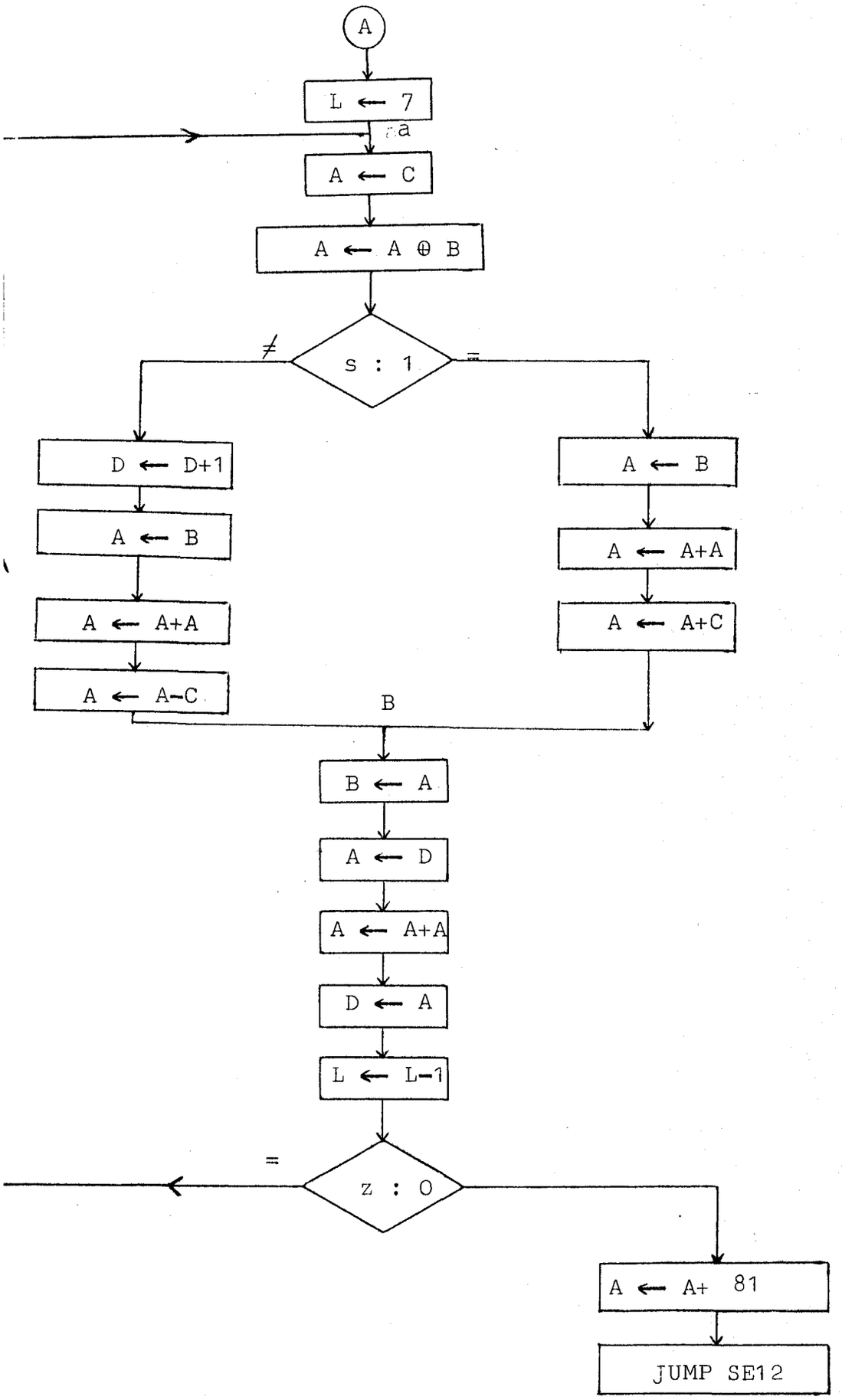
DIRECCION DE ENTRADA: 6/00

PARAMETROS DE ENTRADA: B y C

PARAMETROS DE SALIDA: A

PARAMETROS INTERMEDIOS: H, L, D y B




A
L ← 7
a
A ← C
A ← A ⊕ B
≠
s : 1
=
D ← D+1
A ← B
A ← A+A
A ← A-C
A ← B
A ← A+A
A ← A+C
B
B ← A
A ← D
A ← A+A
D ← A
L ← L-1
=
z : 0
A ← A+ 81
JUMP SE12

EXPLICACION:

Se utiliza, por su simplicidad, rapidez y no requerir correcciones posteriores debidas a los signos,... el algoritmo descrito en la referencia (79 p165).

En el registro B se memoriza el dividendo parcial (resto), en D el cociente, (que inicialmente se pasa a cero) y en C el divisor. Se consideran los números de izquierda a derecha de i = n a i= 0. El algoritmo consiste en lo siguiente:

(I) comparar los signos del divisor (C) y el resto (B).

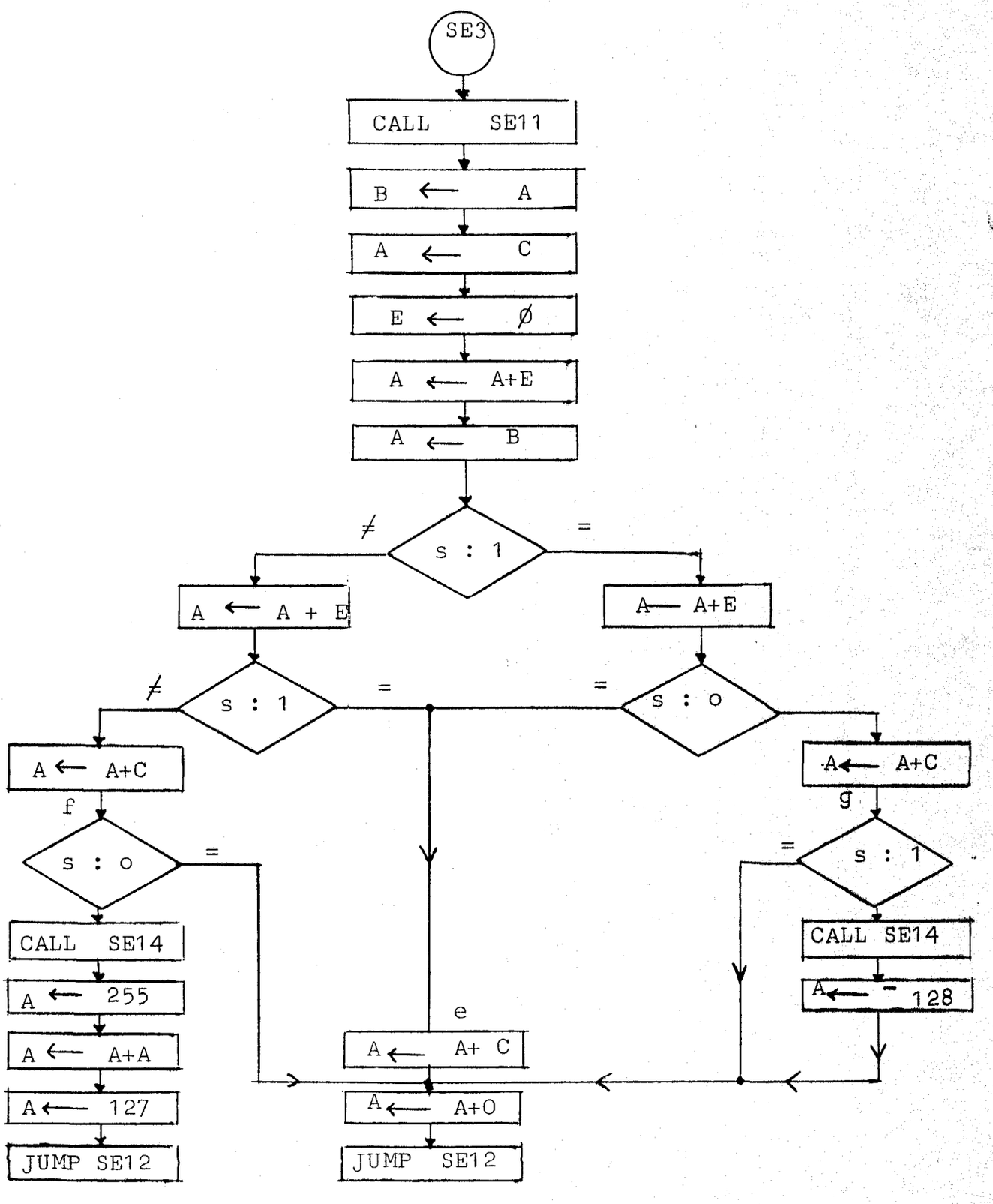
(II) si son iguales escribir un 1 en la posición correspondiente (i) del cociente, D. Multiplicar el resto por 2 y restarle el divisor (C).

(III) si los signos son distintos escribir un cero en la posición correspondiente del cociente, multiplicar el resto por 2 y sumarle el divisor.

(IV) una vez realizada la iteración n veces (el nº de bits de los datos es n+1) efectuar una corrección sumando el cociente obtenido, el número $2^{-n} + 1$. Con esta corrección se obtiene el cociente definitivo incluyendo su redondeo.

NOMBRE: SE3

OBJETIVO: suma de dos números: A ← A + C
SUBRUTINAS QUE UTILIZA: SE11, SE12 y SE14
DIRECCION DE ENTRADA: 7/114
PARAMETROS DE ENTRADA: A y C
PARAMETROS DE SALIDA: A y c
PARAMETROS INTERMEDIOS: -



EXPLICACION: (rutinas SE3 y SE5):

El microprocesador opera sin tener en cuenta el signo de los datos. Para implementar la suma y resta con signo de dos números en notación de complemento a dos y con lógica de "overflow" y "underflow", hay que considerar la siguiente tabla de verdad (donde Sa, Sb y Sr son los signos de los datos y resultados, respectivamente, y c la señal de "carry"):

				suma		resta	
Sa	Sb	c	Sr	overflow	underflow	overflow	underflow
0	0	0	0	0	0	0	0
0	0	0	1	1	0	x	x
0	0	1	0	x	x	x	x
0	0	1	1	x	x	0	0
0	1	0	0	x	x	x	x
0	1	0	1	0	0	x	x
0	1	1	0	0	0	0	0
0	1	1	1	x	x	1	0
1	0	0	0	x	x	0	1
1	0	0	1	0	0	0	0
1	0	1	0	0	0	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	0	0
1	1	0	1	x	x	x	x
1	1	1	0	0	1	x	x
1	1	1	1	0	0	0	0

Es decir:

SUMA: overflow = $\overline{Sa}.\overline{Sb}.Sr.\overline{c} = \overline{Sa}.\overline{Sb}.Sr$

underflow= $Sa.Sb.Sr.c = Sa.Sb.\overline{Sr}$

RESTA: overflow= $\overline{Sa}.Sb.Sr.C = \overline{Sa}.Sb.Sr$

underflow=$Sa.\overline{Sb}.\overline{Sr}.c = Sa.\overline{Sb}.\overline{Sr}$

Los programas realizan las operaciones lógicas anteriores. Caso de overflow el resultado se hace igual al mayor número posible (+127) y se activa el biestable de carry. Caso de underflow el resultado se hace igual al menor número posible (-128).

NOMBRE: SE4

OBJETIVO: obtiene a partir del código de la macroinstrucción la dirección absoluta de comienzo del microprograma que la ejecuta.

SUBRUTINAS QUE UTILIZA: -

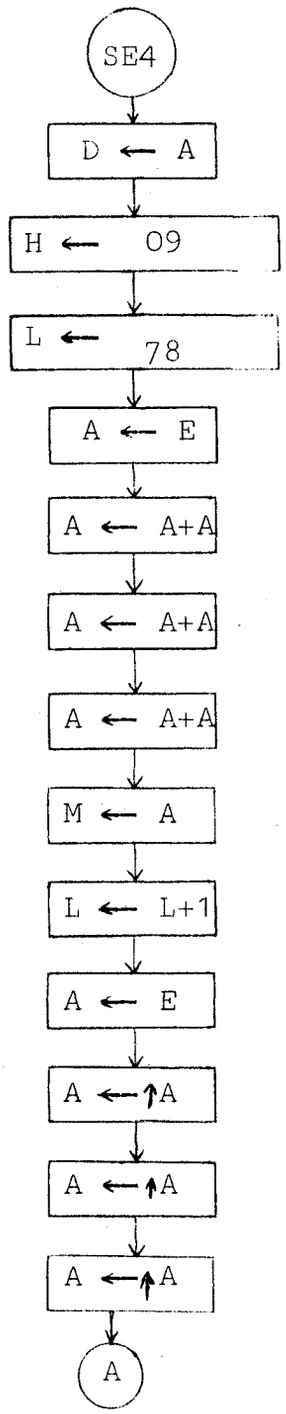
CODIGO DE LLAMADA:

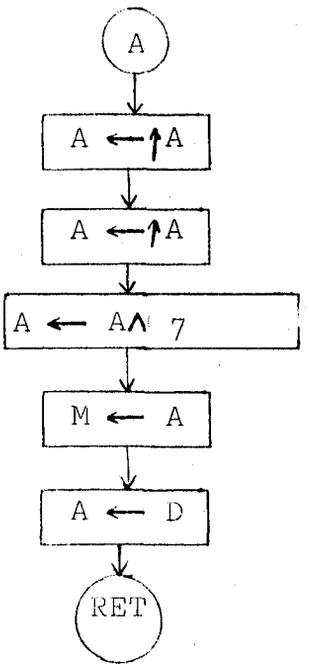
DIRECCION DE ENTRADA: 6/35

PARAMETROS DE ENTRADA: E

PARAMETROS DE SALIDA: -

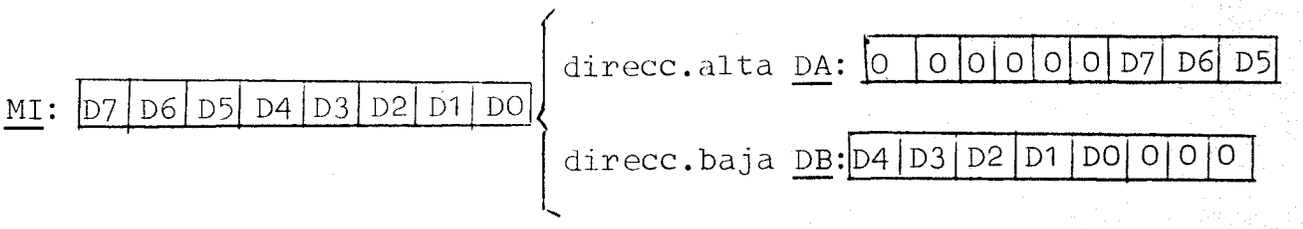
PARAMETROS INTERMEDIOS: D, H y L





EXPLICACION:

El programa realiza la siguiente conversión:



es decir: DA <--- 5 MI

DB <--- 3 MI

estos desplazamientos se realizan tal y como se indica en el organigrama.

Se memoriza:

DA en la dirección 09/4F y

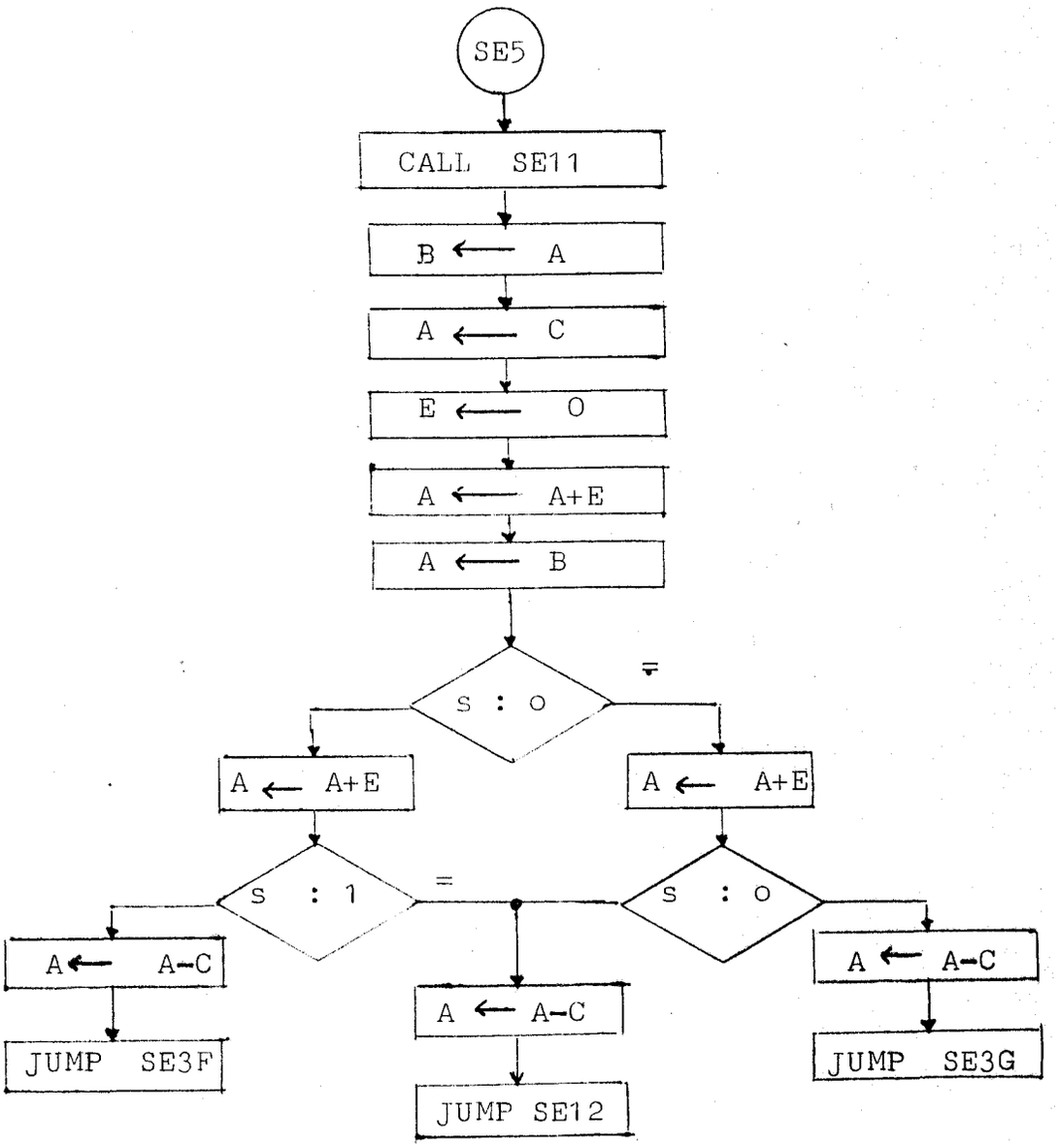
DB en la dirección 09/4E

con lo que en la página 9 queda completa la instrucción de salto al microprograma correspondiente a la macroinstrucción (secc. 3.5.1).

NOMBRE: SE5

OBJETIVO: resta de dos números: A ←— A − C

SUBRUTINAS QUE UTILIZA: SE3F, SE3G, SE11 y SE12
DIRECCION DE ENTRADA: 7/172
PARAMETROS DE ENTRADA: A y C
PARAMETROS DE SALIDA: A y c
PARAMETROS INTERMEDIOS: -



NOMBRE: SE6

OBJETIVO: ensamblaje de macroinstrucción, o lectura del operador correspondiente a la macroinstruccion en ejecución.

SUBRUTINAS QUE UTILIZA: -

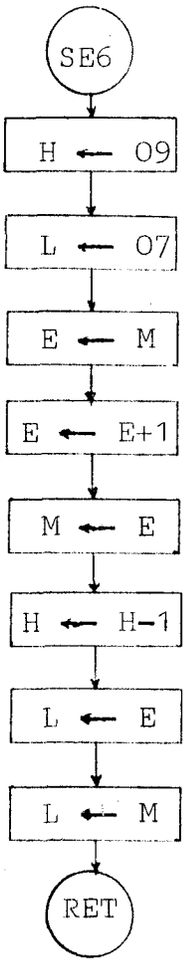
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 6/57

PARAMETROS DE ENTRADA: -

PARAMETROS DE SALIDA: L (operador)

PARAMETROS INTERMEDIOS: E y L



EXPLICACION: El programa efectúa las siguientes operaciones:

1º CP(0) ←— CP(0) + 1
2º L ←—— (8/CP(0));

es decir, en L queda almacenado el operador de la macroinstrucción.

NOMBRE: SE7

OBJETIVO: incremento de punto de página de función para macroinstrucciones de operaciones entre dos funciones.

SUBRUTINAS QUE UTILIZA: -

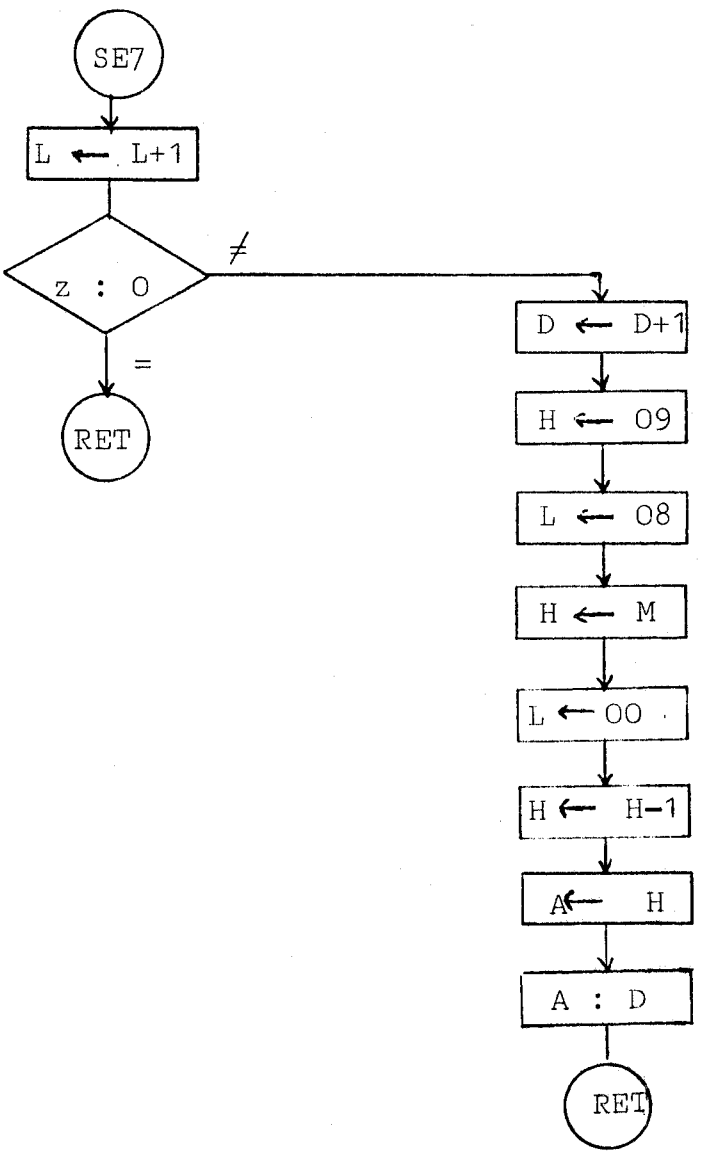
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 4/162

PARAMETROS DE ENTRADA: D, L

PARAMETROS DE SALIDA: D, L y C

PARAMETROS INTERMEDIOS: A y H



EXPLICACION:

En L está almacenado la dirección del octeto dentro de la página y en D el valor que hay que sumar a la dirección de página donde comienzan las funciones FO y Fn para obtener la página de procesamiento. El programa incrementa el valor de L en 1; si es necesario efectuar un cambio de página se incrementa en 1 el valor de D y se lee el parámetro PNP (dirección 09/08) comparándolo con D; en el caso de haber finalizado el procesamiento de toda la función, la báscula de rebose toma el valor 1, pudiendose detectar esta situación en el programa de llamada.

NOMBRE: SE8

OBJETIVO: retroceso de punto de página de función.

SUBRUTINAS QUE UTILIZA: -

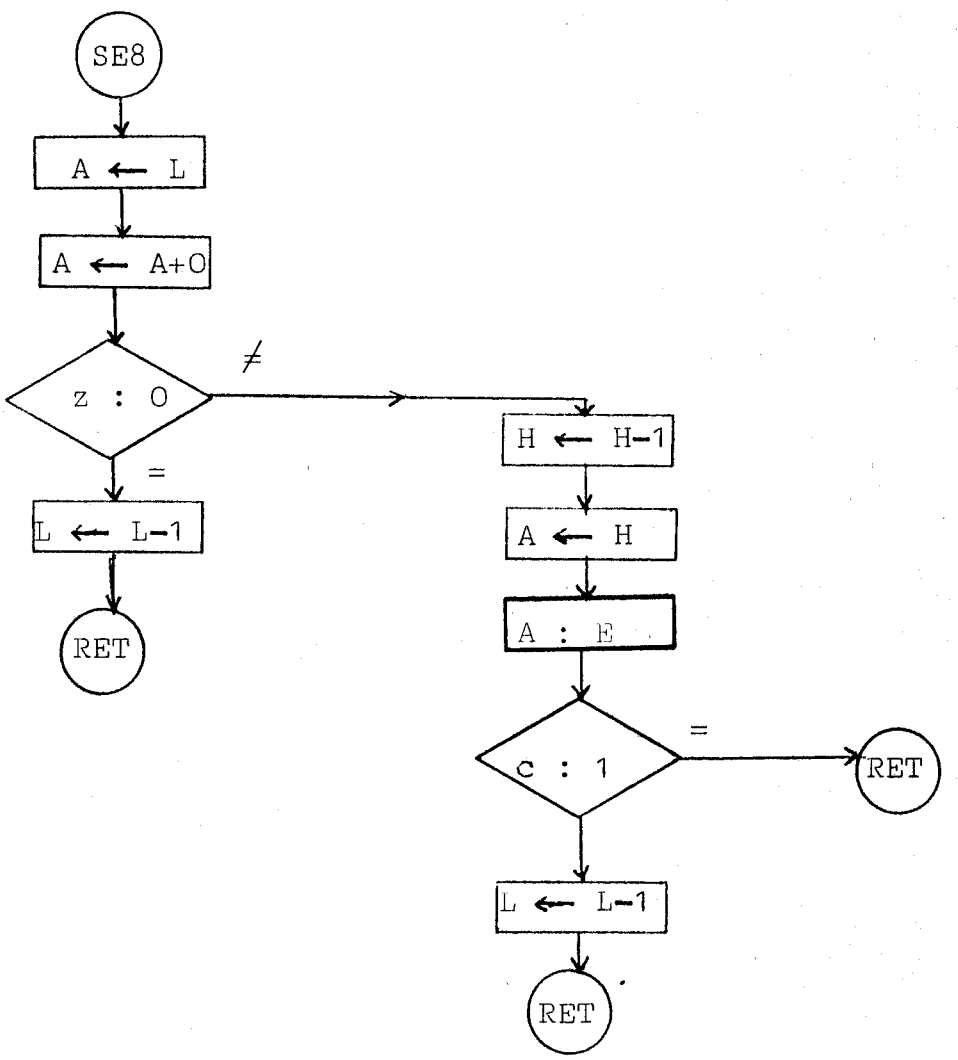
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 4/244

PARAMETROS DE ENTRADA: E, H y L

PARAMETROS DE SALIDA: H, L y c

PARAMETROS INTERMEDIOS: A



EXPLICACION:
E: página inicial de la función (FE)
H: página actual en procesamiento
L: punto dentro de la página de la función
c: igual a 1 si finaliza el procesamiento de la función

NOMBRE: SE9

OBJETIVO almacenar el contenido de los registros A, B, C y D en la página 9 de memoria.

SUBRUTINAS QUE UTILIZA: -

CODIGO DE LLAMADA:

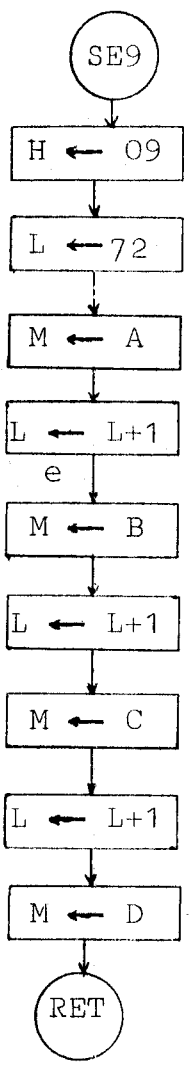
DIRECCION DE ENTRADA: 6/68

PARAMETROS DE ENTRADA: A, B, C y D

PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: H y L

PUNTO DE ENTRADA EXTERNO: SE9E



EXPLICACION: (secc. 3.5.1) Los contenidos de los registros A, B, C y D se salvan memorizandolos a partir de la dirección de memoria 9/72. Esta rutina se utiliza en aquellos programas que necesiten operar con los registros; al final, con SE10, se recupera el contenido inicial de los mismos.

NOMBRE: SE1O

OBJETIVO: almacenar en los registros A, B, C y D el contenido de los octetos de memoria 9/72 a 9/75

SUBRUTINAS QUE UTILIZA: -

CODIGO DE LLAMADA:

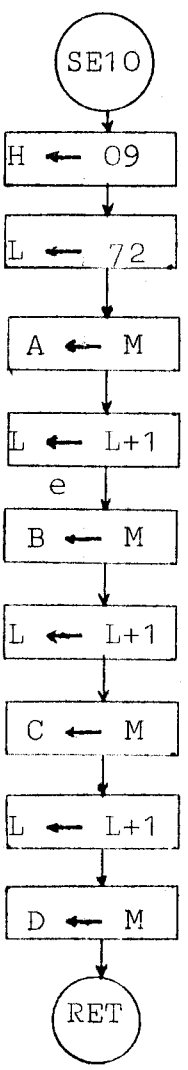
DIRECCION DE ENTRADA: 6/8O

PARAMETROS DE ENTRADA: -

PARAMETROS DE SALIDA: A, B, C y D

PARAMETROS INTERMEDIOS: H y L

PUNTO DE ENTRADA: SE1OE



EXPLICACION: ver rutina SE9.

NOMBRE: SE11

OBJETIVO: almacenar el contenido de los registros B, C, D y E en la página 9 de la memoria.

SUBRUTINAS QUE UTILIZA:

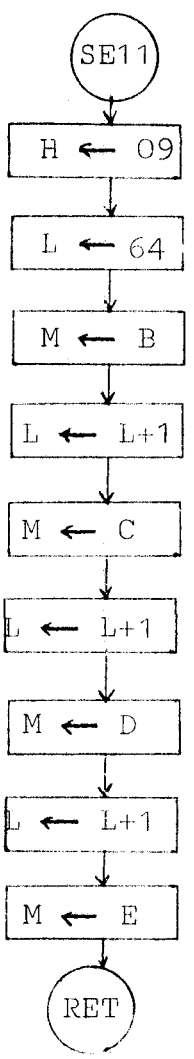
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 6/92

PARAMETROS DE ENTRADA: B, C, D y E

PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: H y L



EXPLICACION: Rutina análoga a la SE9, la memorización se realiza en las posiciones de memoria 9/64, 9/65, 9/66 y 9/67. (secc. 3.5.1).

NOMBRE: SE12

OBJETIVO: Almacenar en los registros B, C, D y E el contenido de los octetos 9/64 a 9/67.

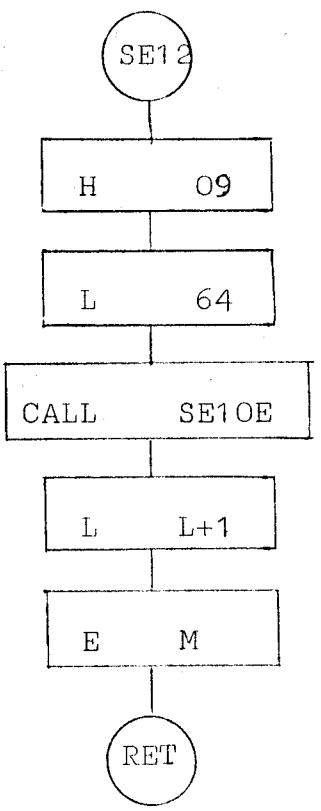
SUBRUTINAS QUE UTILIZA: SE10E

DIRECCION DE ENTRADA: 4/110

PARAMETROS DE ENTRADA: -

PARAMETROS DE SALIDA: B, C, D y E

PARAMETROS INTERMEDIOS: H y L



EXPLICACION: ver rutina SE10.

NOMBRE: SE13

OBJETIVO: escribir en el dispositivo de salida 01 un mensaje indicando que ha habido una saturación de capacidad en un exponente.

SUBRUTINAS QUE UTILIZA: -

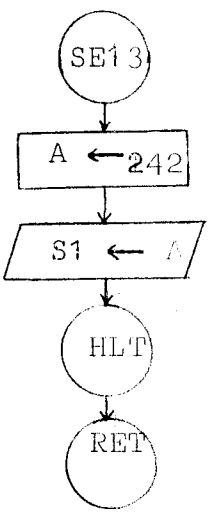
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 0/234

PARAMETROS DE ENTRADA: -

PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: A



NOMBRE: SE14

OBJETIVO: escribir en el dispositivo de salida S1 un mensaje indicando que ha habido desbordamiento en la capacidad de una mantisa

SUBRUTINAS QUE UTILIZA: -

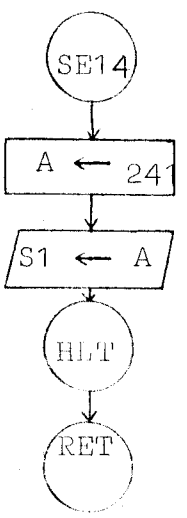
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 0/226

PARAMETROS DE ENTRADA: -

PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: A



NOMBRE: SE15

OBJETIVO: leer el parámetro (n) de la macroinstrucción que se esta ejecutando, y obtener la dirección absoluta de comienzo de una función (macroinstrucciones de formato 2, secc. 2.3.1)

SUBRUTINAS QUE UTILIZA: SA3A

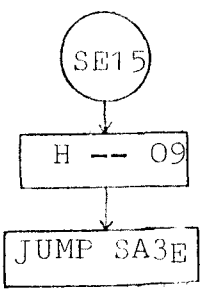
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 1/57

PARAMETROS_DE ENTRADA: -

PARAMETROS DE SALIDA: E, L

PARAMETROS INTERMEDIOS: H



EXPLICACION: obtiene a partir de la dirección simbólica de una función (n) la dirección absulta de comienzo (ver SA3)

NOMBRE: SE17

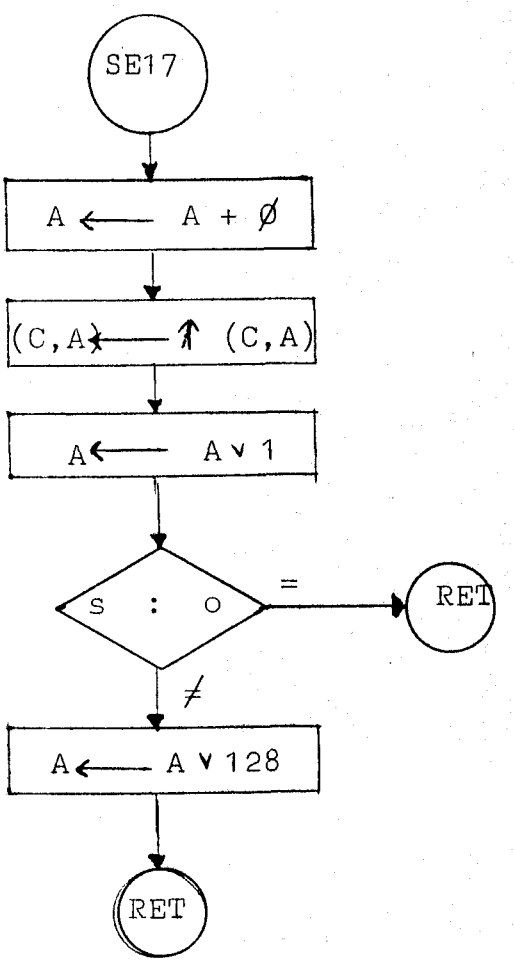
OBJETIVO: desplazamiento a derecha de un número (dividir por 2)

DIRECCION DE ENTRADA: 7/234

PARAMETROS DE ENTRADA: A

PARAMETROS DE SALIDA: A

PARAMETROS INTERMEDIOS: -



3.5.3.3 SUBRUTINAS AUXILIARES:

NOMBRE: SA1

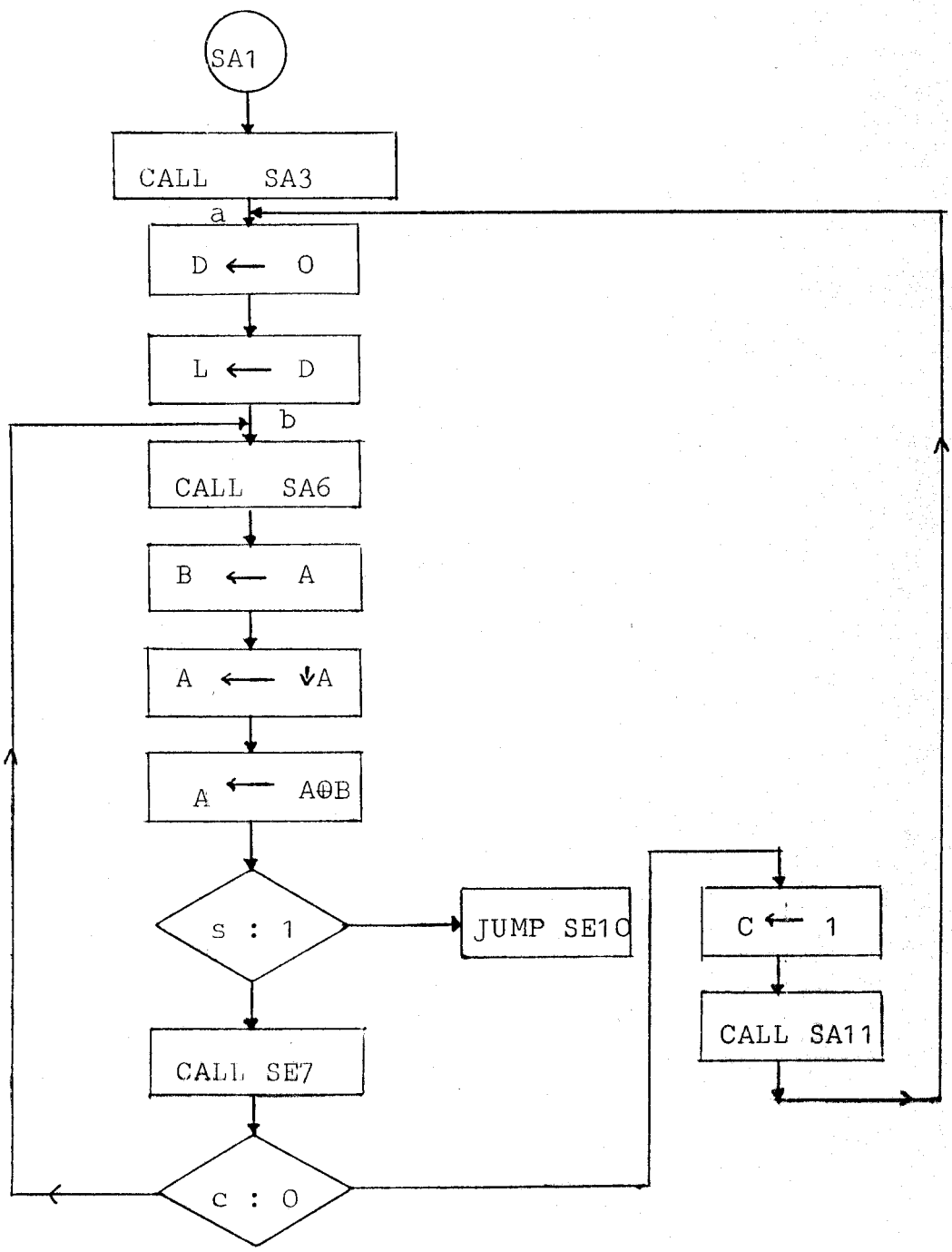
OBJETIVO: normalización de una función (FE)

SUBRUTINAS QUE UTILIZA: SA3, SA11, SE7 y SE10

DIRECCION DE ENTRADA: 3/191

PARAMETROS DE ENTRADA: E

PARAMETROS DE SALIDA: - INTERMEDIOS: A, B, C, D, H y L



EXPLICACION:

La condición de normalización (secc. 3.3.2) es que por lo menos un punto de la función cumpla la condición:

$$D7 \otimes D6 = 1$$

La rutina recorre toda la función punto a punto; si al final de recorrerla no se ha cumplido la condición de normalización en ningún punto, se llama a SA11 para desplazar las mantisas de la función un lugar a la izquierda y disminuir el exponente de la función en 1, volviendose a repetir la prueba desde el primer punto de la función. Al recorrer la función en el primer punto que se cumpla la condición de normalización se devuelve el control al programa de llamada, acabando el proceso de normalización.

NOMBRE: SA2

OBJETIVO: ecualización de FO y FE

SUBRUTINAS QUE UTILIZA: SE5, SA5 y SA10

CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 6/102

PARAMETROS DE ENTRADA: E

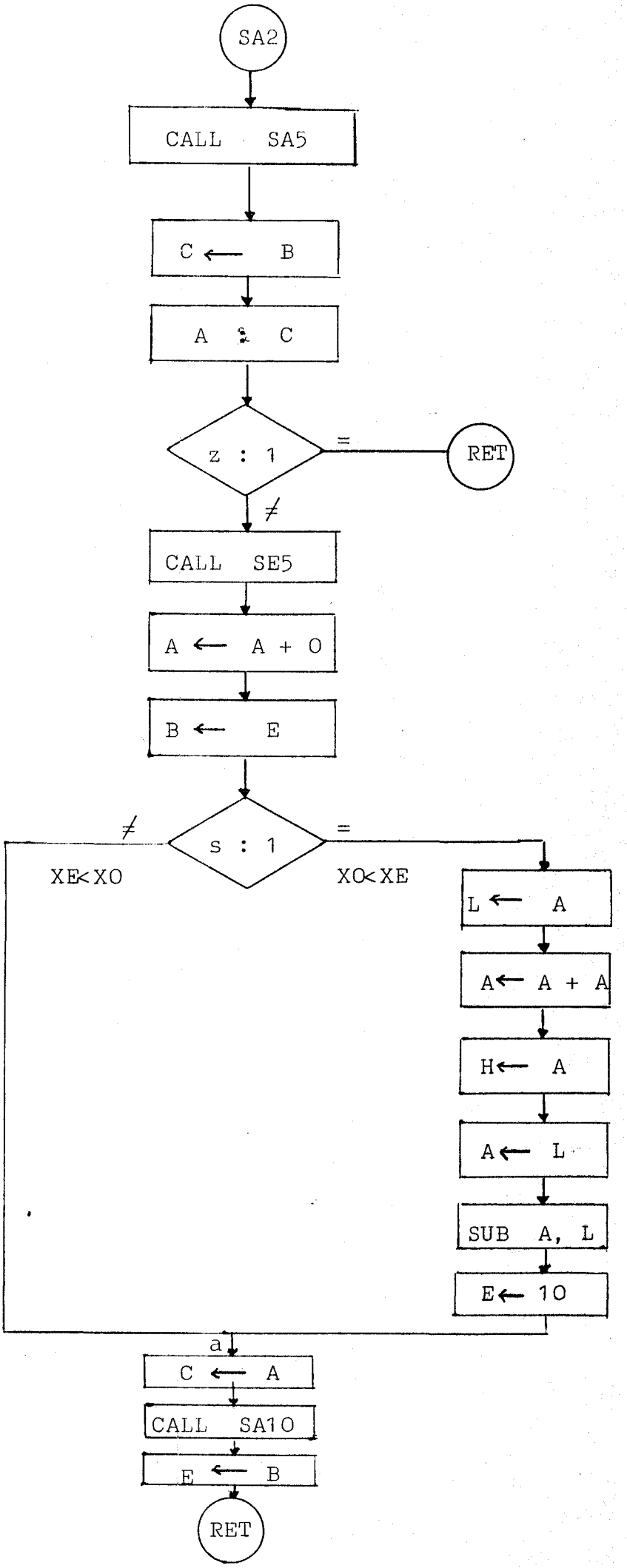
PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS A, B, C, D, H y L

EXPLICACION:

El algoritmo esencialmente consiste en lo siguiente:

1) hacer A = XFO - XFE
   si A es 0 ir a (5); sino ir a (2)
2) si c=1 es XFO < XFE, ir a (3); sino ir a (4)
3) número de desplazamientos ha hacer pra ecualizar: XFE-XFO, hacer E = 10 (desplazar la función FO)
4) hacer C=A, y llamar a SA10. Recuerdese que SE10 efectua desplazamientos a la derecha con modificación de exponente, los parámetros de entrada son: E(página a desplazar) y C (número de desplazamientos)
5) fin.


SA2
CALL SA5
C ← B
A ↕ C
z : 1
=
RET
≠
CALL SE5
A ← A + O
B ← E
s : 1
≠
XE<XO
=
XO<XE
L ← A
A ← A + A
H ← A
A ← L
SUB A, L
E ← 10
a
C ← A
CALL SA10
E ← B
RET

NOMBRE: SA3

OBJETIVO: comienzo de ejecución de macroinstrucciones con formato de tipo 2 (secc. 2.3.1)

SUBRUTINAS QUE UTILIZA: -

CODIGO DE LLAMADA:

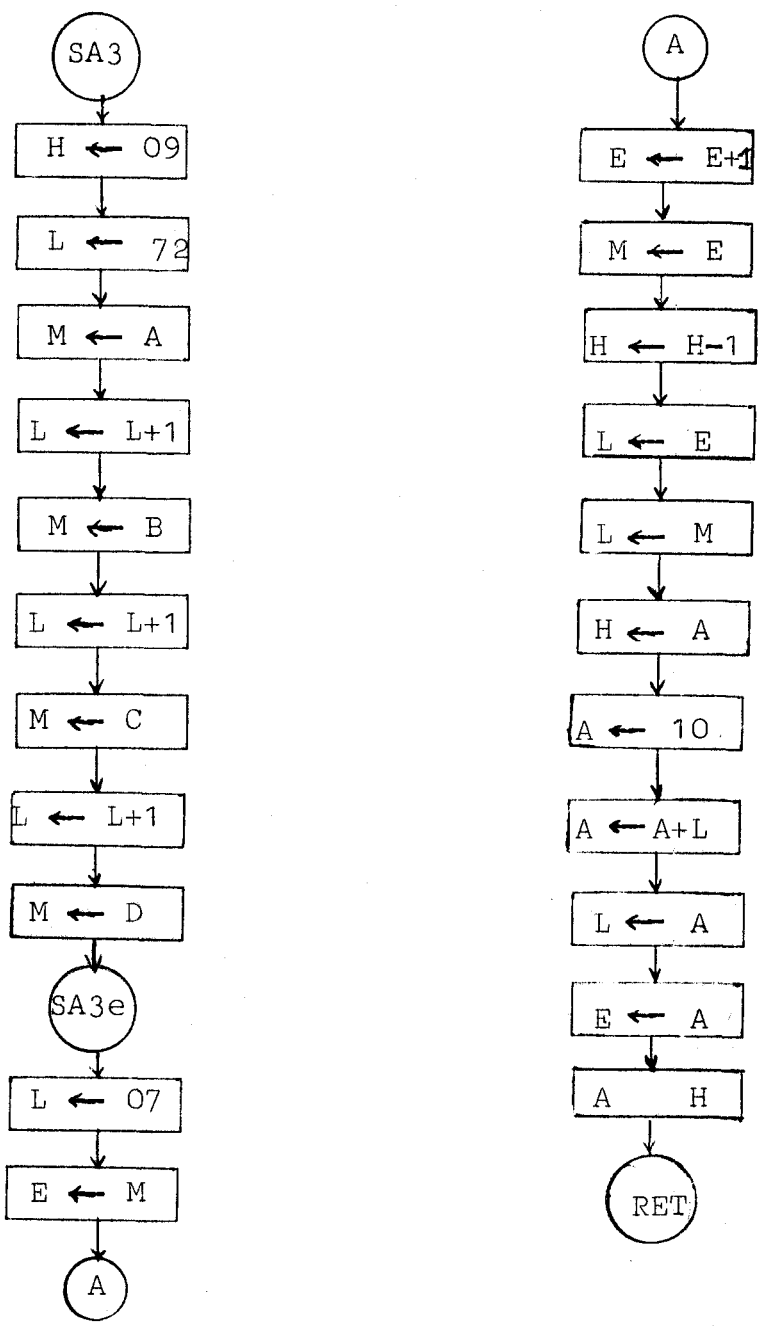
DIRECCION DE ENTRADA 3/220

PARAMETROS DE ENTRADA: A, B, C y D

PARAMETROS DE SALIDA: L y E

PARAMETROS INTERMEDIOS: H

PUNTOS DE ENTRADA EXTERNOS: SA3E



EXPLICACION:

1) Se salvan los registros A,B,C y D (ver rutina SE9)
2) Se ensambla la macroinstrucción, obteniendose la dirección simbólica de la función dada por el operador.
3) Se obtiene la dirección de la página correspondiente a la función a partir de su nombre simbólico.

NOMBRE: SA4

OBJETIVO: leer puntos de dos funciones en procesamiento:
C ←— FE(i) A ←— FO(i)

SUBRUTINAS QUE UTILIZA: -

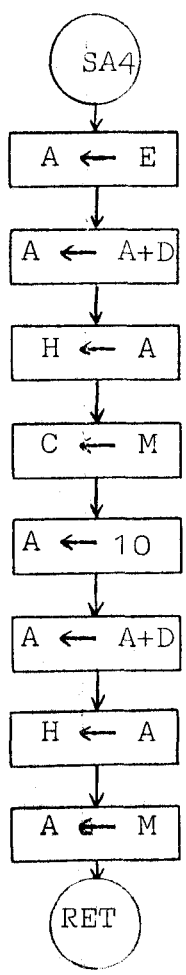
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 2/246

PARAMETROS DE ENTRADA: E,D

PARAMETROS DE SALIDA: C, A

PARAMETROS INTERMEDIOS: H y L



EXPLICACION: A partir de los parámetros de salida de la rutina SE7, se leen dos puntos de FO y FE de un instante dado (i).

NOMBRE: SA5

OBJETIVO: lectura de dos exponentes de funciones:
B ←--- XFE A ←--- XFO

SUBRUTINAS QUE UTILIZA: -

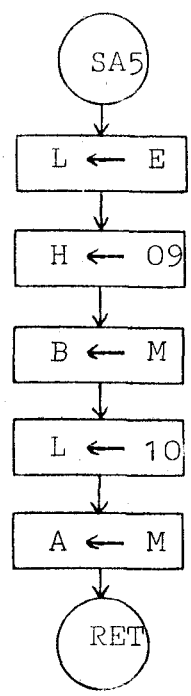
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 2/65

PARAMETROS DE ENTRADA: E

PARAMETROS DE SALIDA: A y B

PARAMETROS INTERMEDIOS: H y L



EXPLICACION: ver secc. 3.5.1.

NOMBRE: SA6

OBJETIVO: Lectura de un punto de una función:
A ←--- FE(i)

SUBRUTINAS QUE UTILIZA: -

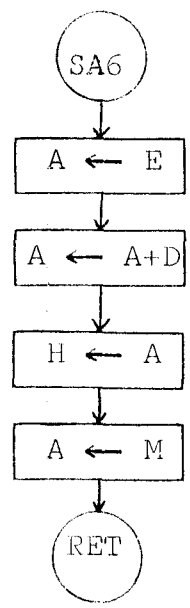
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 0/250

PARAMETROS DE ENTRADA: E, D y L (parámetros de rutina SE7)

PARAMETROS DE SALIDA: A

PARAMETROS INTERMEDIOS: H



NOMBRE: SA7

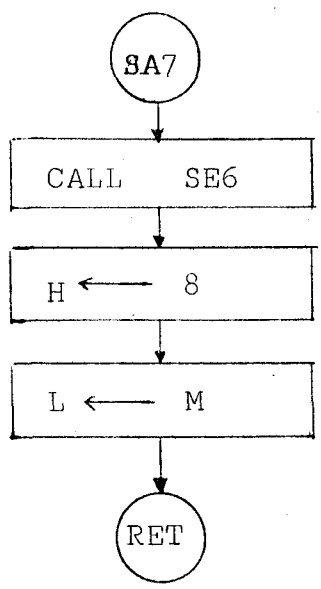
OBJETIVO: inicialización de rutinas que operan con valores almacenados en memoria, cuya dirección viene dada de forma inmediata: L ⟵ (m)

SUBRUTINAS QUE UTILIZA: SE6

DIRECCION DE ENTRADA: 2/185

PARAMETROS DE ENTRADA: -

PARAMETROS DE SALIDA: L INTERMEDIOS: H y E



NOMBRE: SA9

OBJETIVO: anidamiento de direcciones en llamadas a rutinas

SUBRUTINAS QUE UTILIZA: -

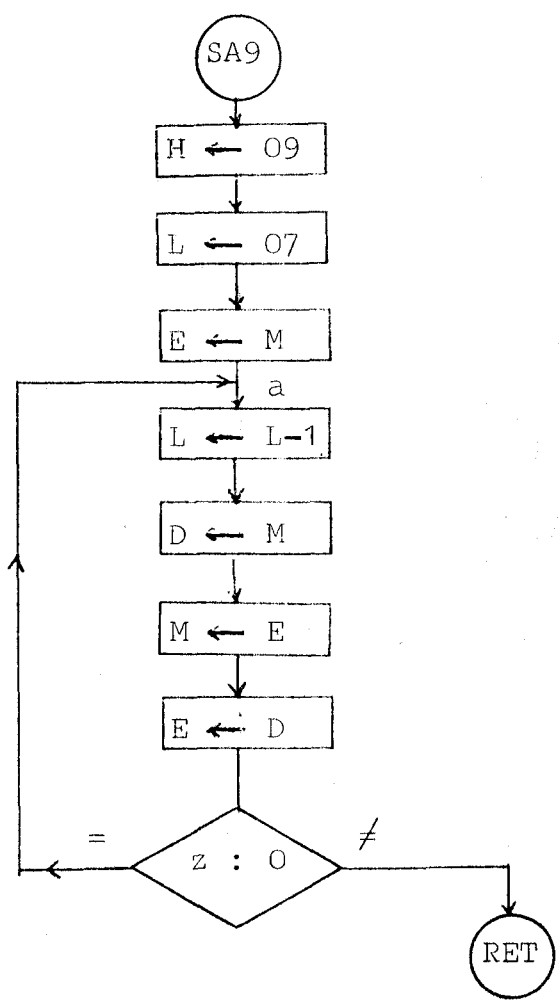
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 4/202

PARAMETROS DE ENTRADA: -

PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: H, L, E y D



EXPLICACION: Se efectua la siguiente operación:

CP(i+1) ←--- CP(i), donde i varia de 0 a 6

En la secc. 3.5.1 se indican las posiciones de memoria donde se almacena CP(0),...

NOMBRE: SA10

OBJETIVO : desplazamiento de todos los puntos de una función FE un número C de lugares a la derecha con modificacion de exponente.

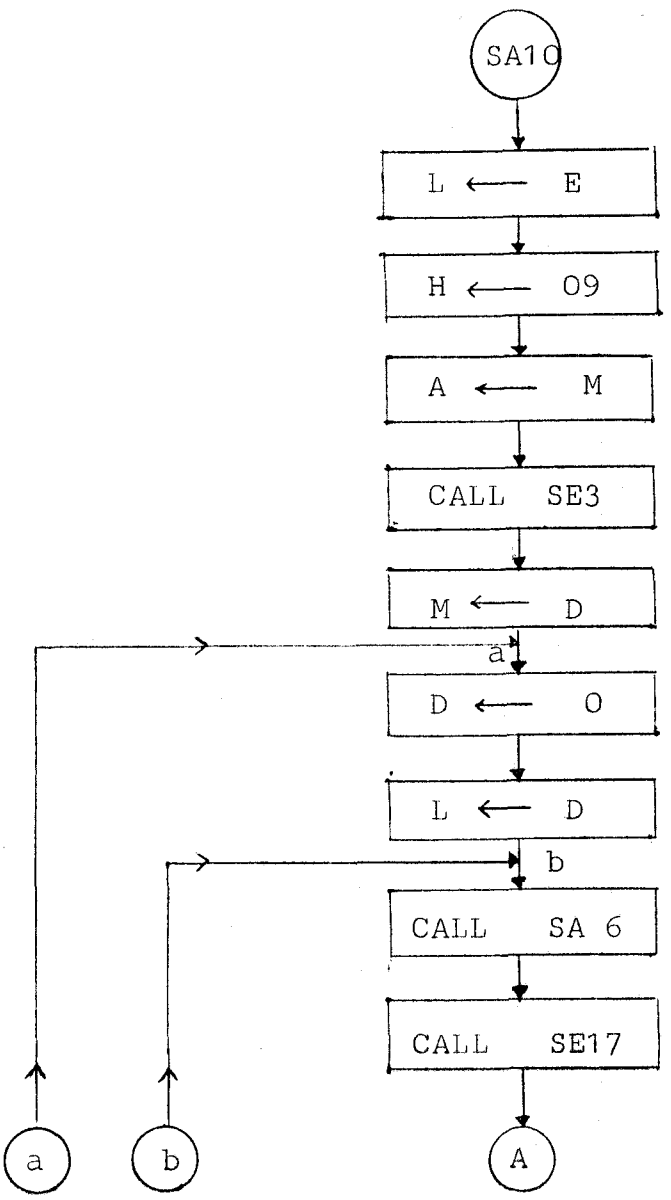
SUBRUTINAS QUE UTILIZA: SE3, SE7, SE13 y SA6

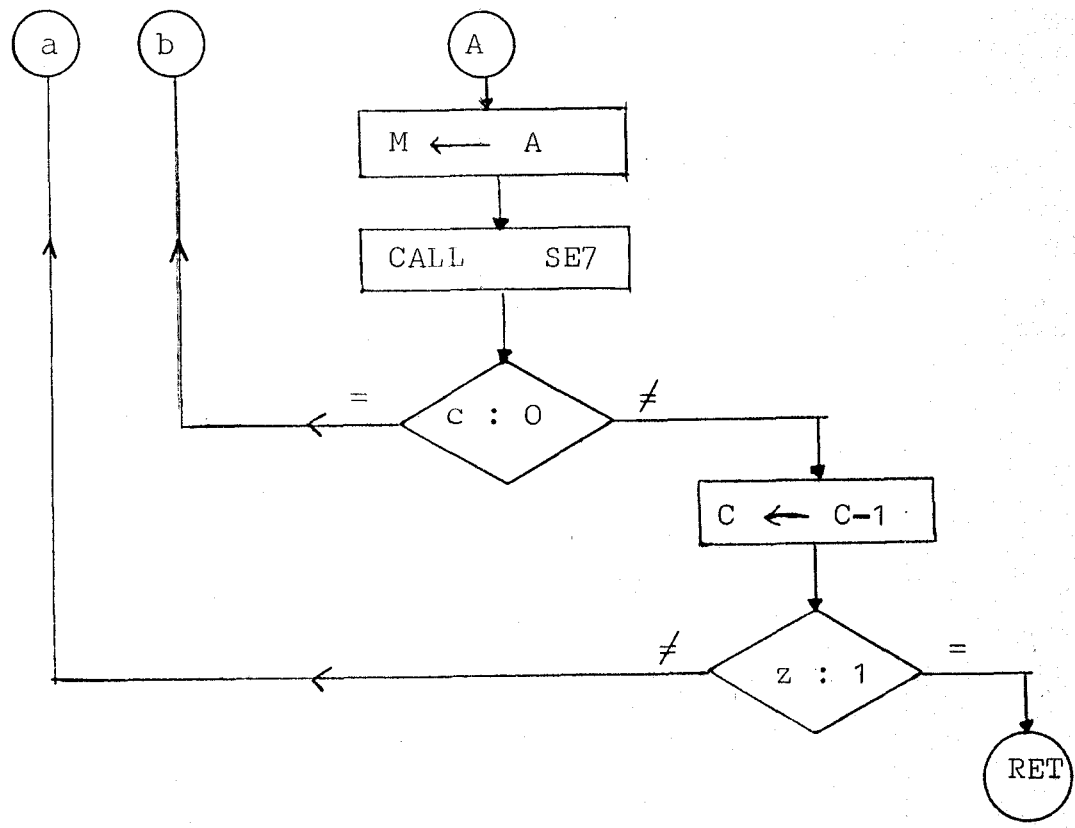
DIRECCION DE ENTRADA: 6/130

PARAMETROS DE ENTRADA: E y C

PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: A, C, D, L y H





EXPLICACION:

(1) se hace XFE ←--- XFE + C

(2) las mantisas se desplazan C lugares a la derecha, completando:
con ceros, los números positivos
con unos, los números negativos

NOMBRE: SA11

OBJETIVO: Desplazamiento de todos los puntos de una función (FE) un número (C) de posiciones a la izquierda con modificación de exponente.

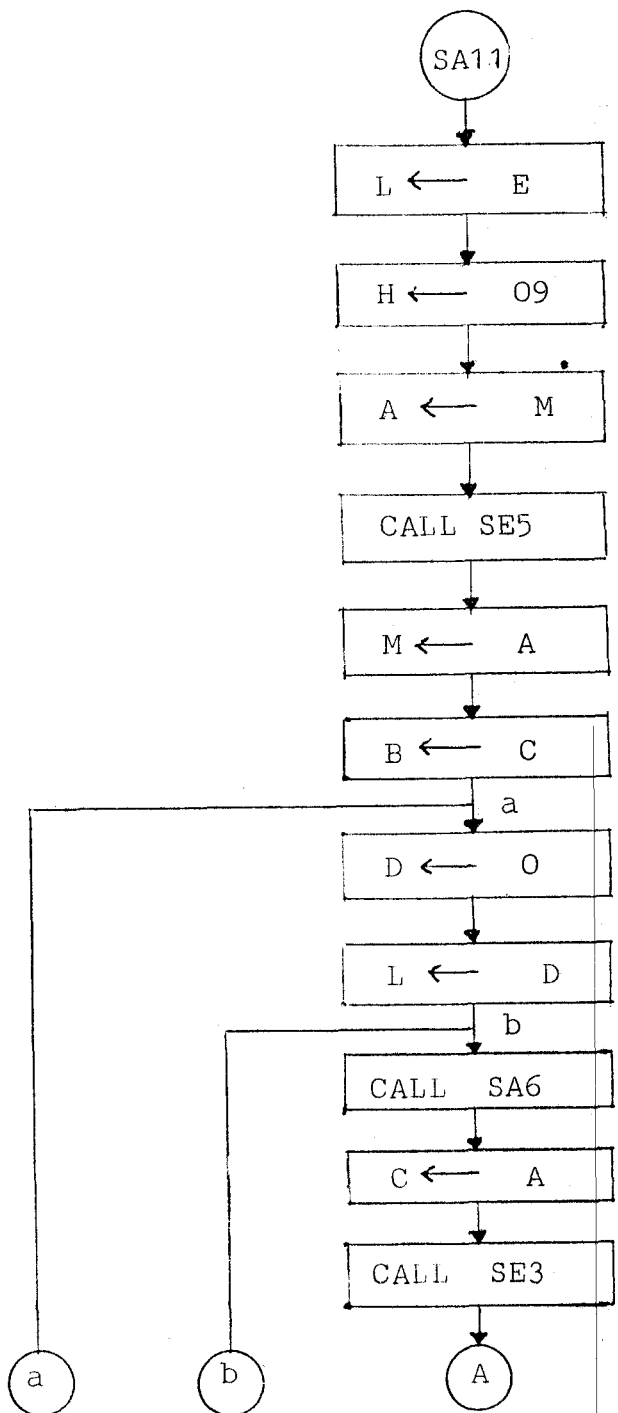
SUBRUTINAS QUE UTILIZA: SE3, SE5, SE7, SE13, SE14 y SA6

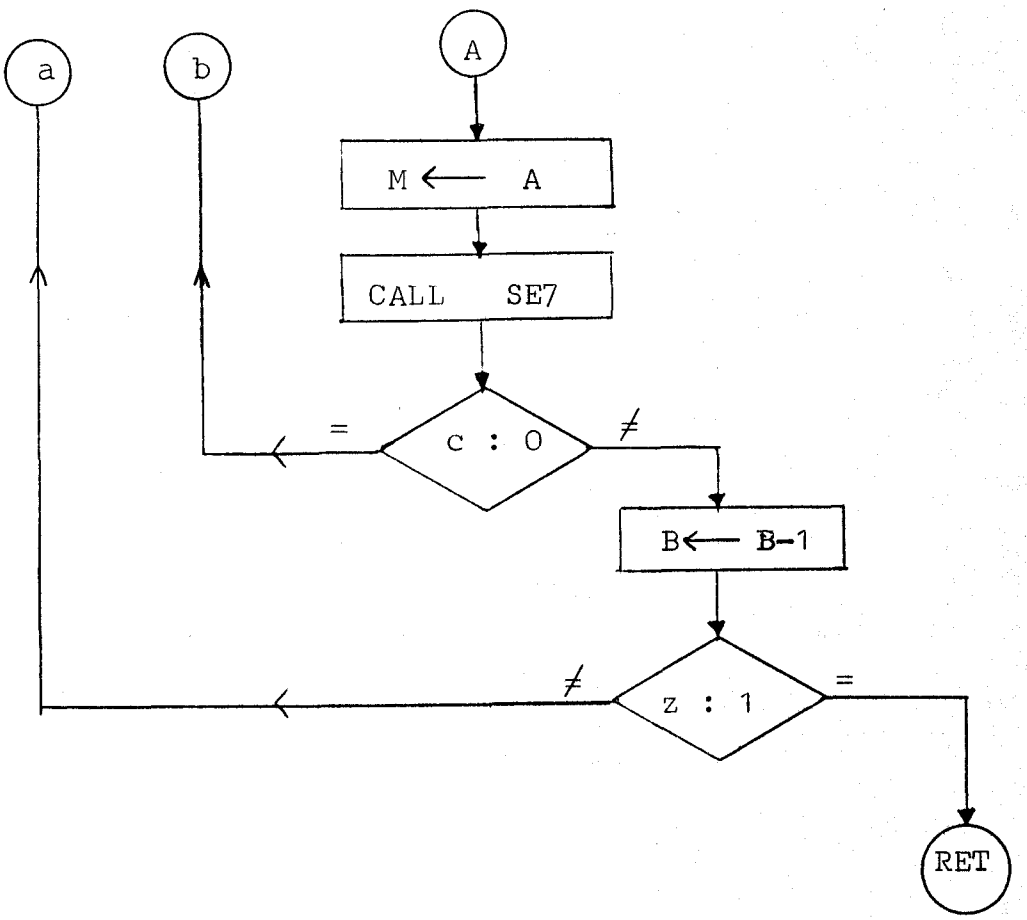
DIRECCION DE ENTRADA: 6/159

PARAMETROS DE ENTRADA: E y C

PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: A, C, D, L y H





EXPLICACION: rutina analoga a SA10.

NOMBRE: SA12

OBJETIVO: Traslación negativa de la función FE una posición.

SUBRUTINAS QUE UTILIZA: SA12A y SA12B

CODIGO DE LLAMADA:

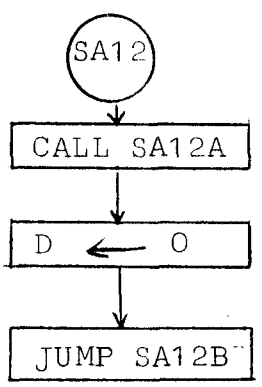
DIRECCION DE ENTRADA: 0/202

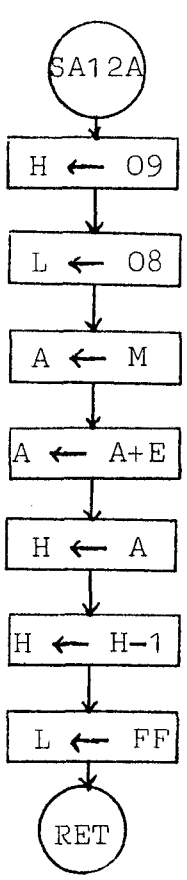
PARAMETROS DE ENTRADA: E

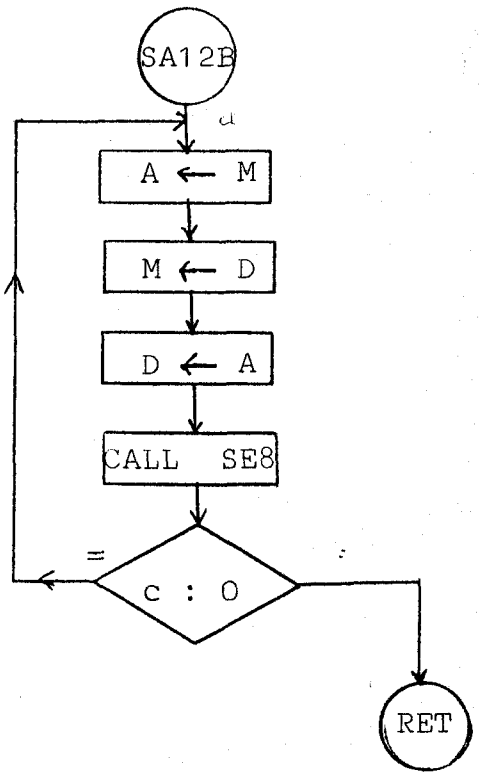
PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: A, D, H y L

PUNTOS DE ENTRADA EXTERNOS: SA12A y SA12B.







EXPLICACION:

La operación que hace esta rutina es:

$$FE = (FE1,\ FE2,\ldots,\ FE225,\ 0)$$

El recorrido de la función se hace con ayuda de la subrutina SE8, empezando por el último punto de la función (255 o múltiplo de 255).

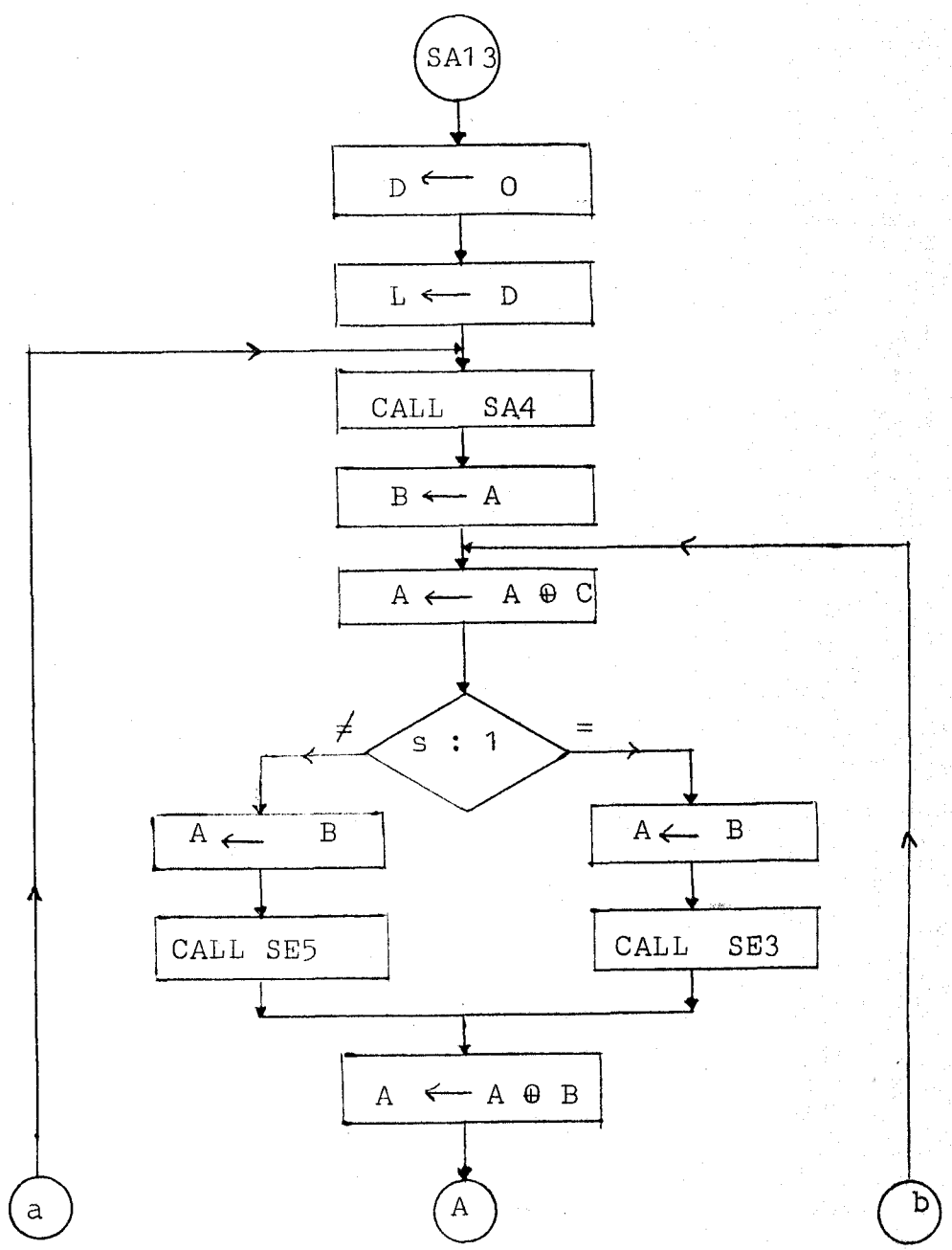
NOMBRE: SA13

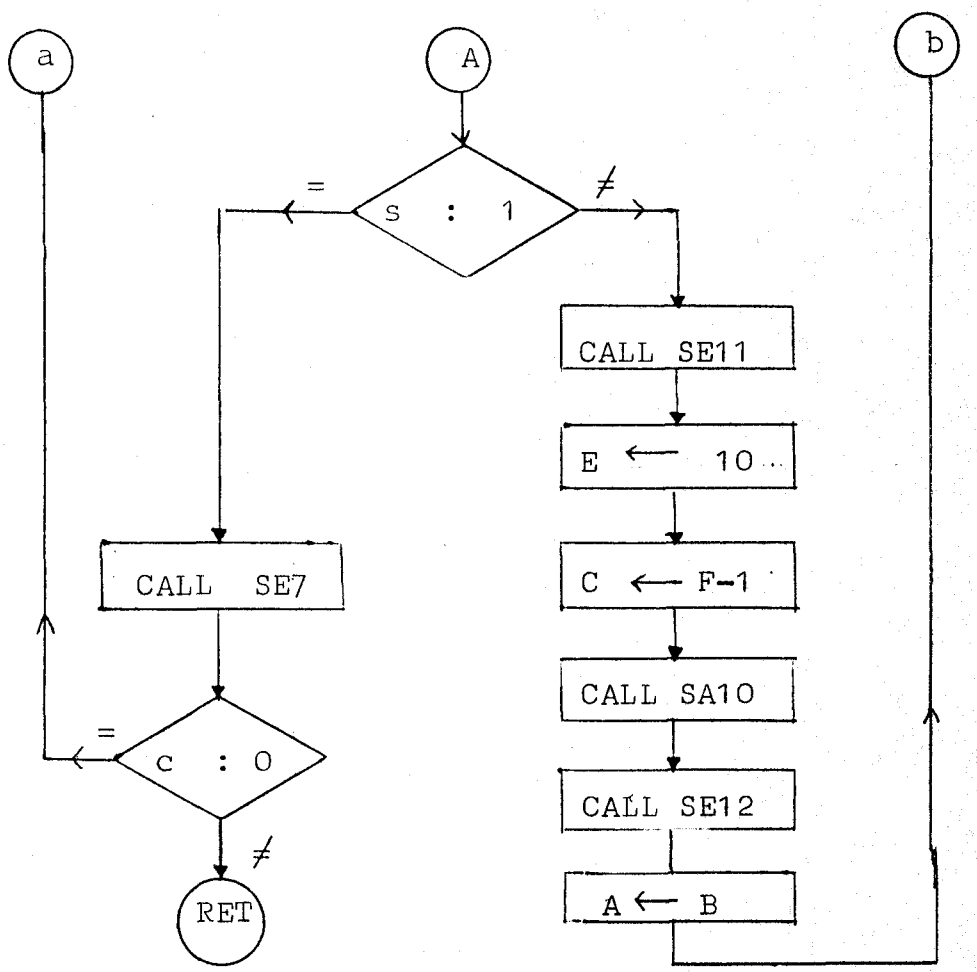
OBJETIVO: Desplazamiento de una función para que las mantisas de FO sean menores en valor absoluto que las correspondientes de FE.

SUBRUTINAS QUE UTILIZA: SE3, SE5, SE7, SE11, SE12, SA4 y SA10

DIRECCION DE ENTRADA: 6/211

PARAMETROS DE ENTRADA: E SALIDA: - INTERMEDIOS: A,B,C,D,L yH





EXPLICACION: Esta rutina se llama antes de ejecutar una división entre funciones (secc. 3.4). Por medio de las rutinas SE7 y SA4 se van recorriendo octeto a octeto todos los valores de las funciones FO y FE. El algoritmo que se utiliza para ver si un número, A, es mayor que otro, C, en valor absoluto es el siguiente:

(1) comparar los signos de A y C
(2) si son distintos sumar A y C
(3) si son iguales restar: A - C
(4) si el signo del resultado es igual al de A, A es mayor en valor absoluto que C, en caso contrario C es mayor que A.

En el algoritmo implementado la comparación entre signos se hace utilizando la operación exclusive-OR. Caso de que en un punto se verifique A<C se llama a la rutina SA10 para dividir las mantisas de FO por 2.

SUBRUTINAS DE ENTRADA Y SALIDA DE FUNCIONES:

Hay cinco subrutinas auxiliares de las rutinas de entrada/salida de funciones (S150, S151, S152, y S153). Estas cinco subrutinas son:

- entrada con ralentización de la velocidad de conversión: SA14
- entrada sin ralentización: SA13
- salida con ralentización de velocidad: SA17
- salida sin ralentización: SA18
- subrutina de inicialización de las anteriores: SA16.

Los parámetros de la subrutinas son:

- ralentización : B
- punto de la función (i): C y L
- página de la función: H
- número de orden de la página dentro de la función: D

La ralentización de la velocidad de conversión se realiza mediante un bucle. Se puede observar que la filosofía seguida es que el lazo iterativo sea lo más rápido posible y dentro de él no se llama a ninguna subrutina. También se ha realizado de tal forma que la primera iteración sea exactamente igual que las demás, con el objeto de que el muestreo sea completamente periódico. Se recuerda que el tiempo de ejecución del lazo es precisamente el periodo de muestreo.

NOMBRE: SA14

OBJETIVO: memorización de una función leída del conversor A/D con ralentización.

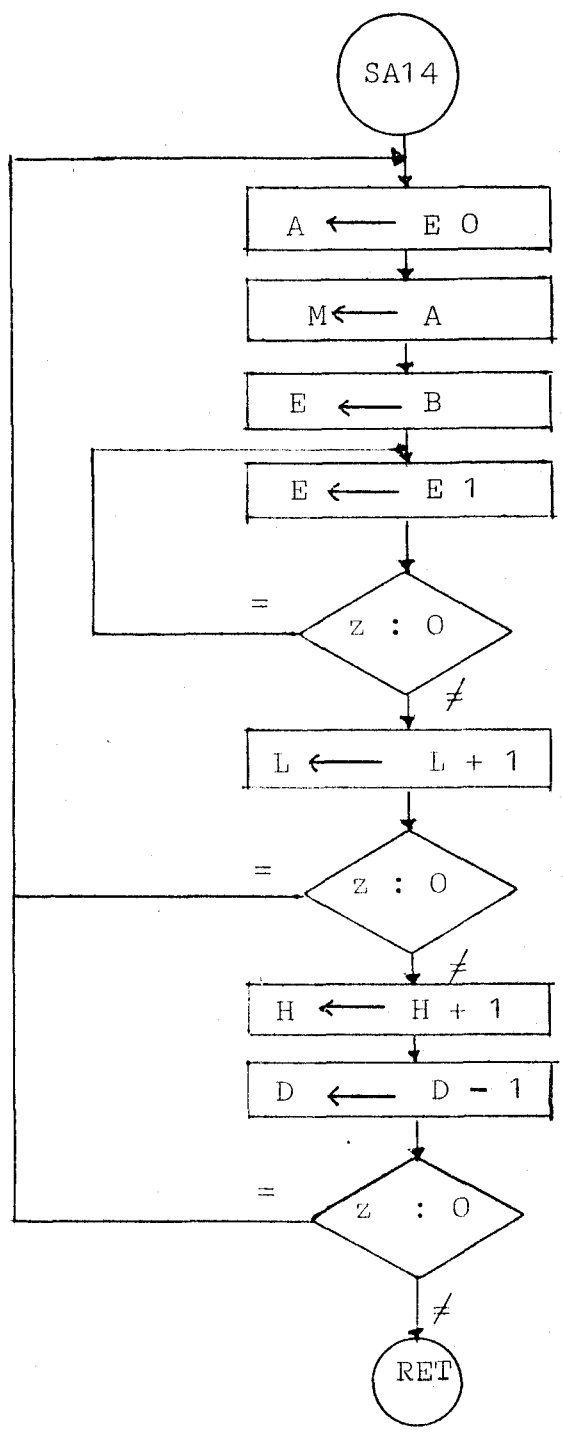
SUBRUTINAS QUE UTILIZA: -

DIRECCION DE ENTRADA: 7/7

PARAMETROS DE ENTRADA: B, C, D, H y L

PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: A, C, D, E H y L



NOMBRE: SA15

OBJETIVO: Memorización de una función leída del conversor A/D sin ralentización.

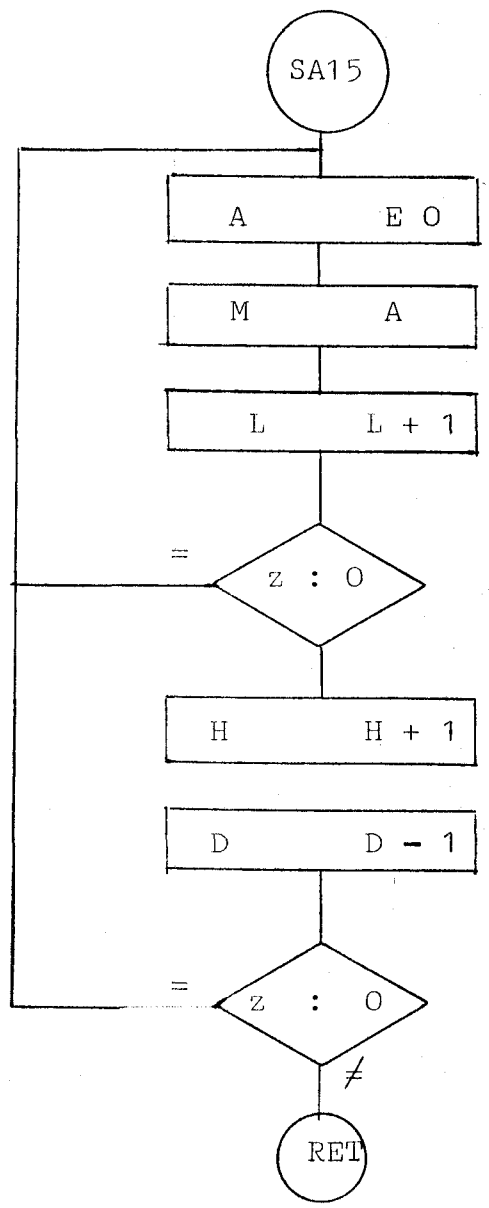
SUBRUTINAS QUE UTILIZA: -

DIRECCION DE ENTRADA: 9/24

PARAMETROS DE ENTRADA: C, D, H y L

PARAMETROS DE SALIDA: -

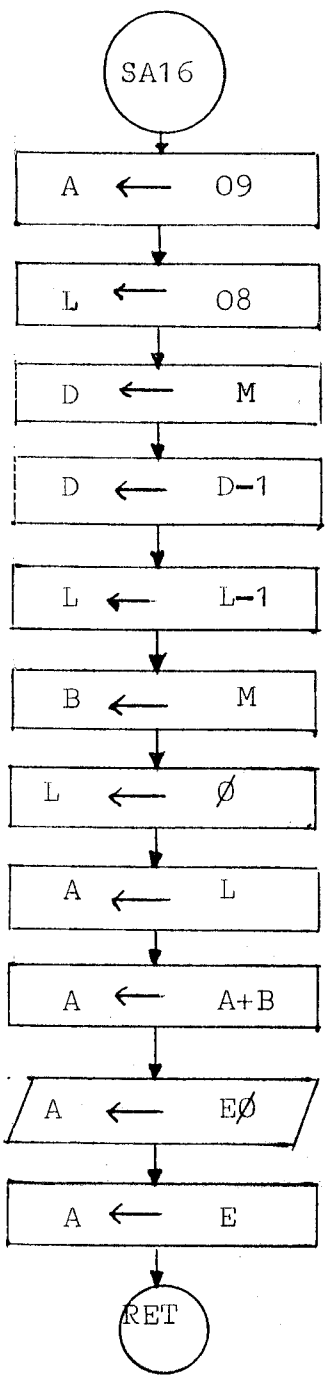
PARAMETROS INTERMEDIOS: A, C, D, H y L.



NOMBRE: SA16

OBJETIVO: inicialización de rutinas de E/S por conversores (rutinas SA14, SA15, SA17 y SA18).

SUBRUTINAS QUE UTILIZA: -
DIRECCION DE ENTRADA: 7/36
PARAMETROS DE ENTRADA: E y L
PARAMETROS DE SALIDA: B, C, D, H, L y z
PARAMETROS INTERMEDIOS: A



La rutina efectua las siguientes operaciones:

(1) leer los parámetros PNP y PR de la página 9

(2) inicializar los siguientes registros:

D = PNP

B = PR

L = 0 (dirección de octeto dentro de la página)

H = página inicial de la función.

NOMBRE: SA17

OBJETIVO: Salida con ralentización de una función almacenada en memoria, por medio del conversor D/A.

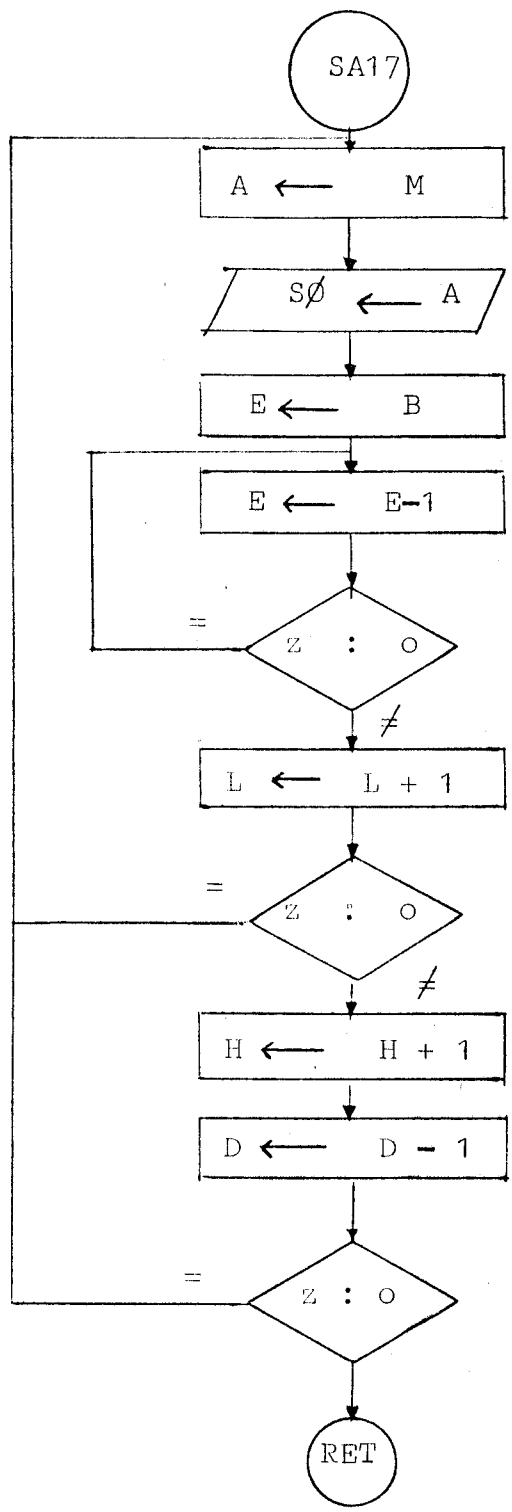
SUBRUTINAS QUE UTILIZA: -

DIRECCION DE ENTRADA: 7/51

PARAMETROS DE ENTRADA: B, C, D, H y L

PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: A, C, D, E, H y L



NOMBRE: SA18

OBJETIVO: Salida por medio del conversor D/A de una función almacenada en memoria.

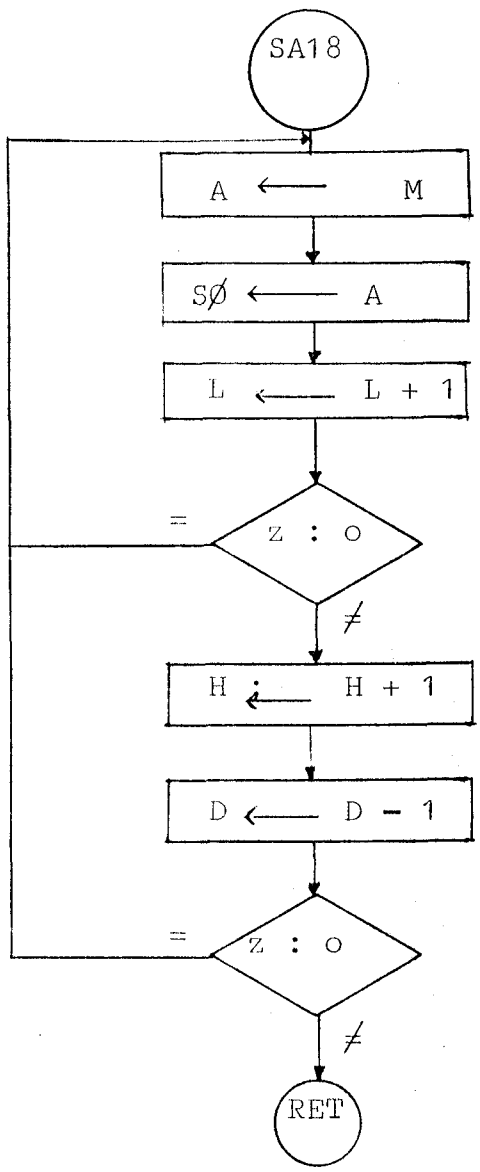
SUBRUTINAS QUE UTILIZA: -

DIRECCION DE ENTRADA: 7/68

PARAMETROS DE ENTRADA: C, D, H y L

PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: A, C, D, H y L.



NOMBRE: SA20

OBJETIVO: Transferencia de una función a otra, incluyendo exponente:

FO ←— FE

SUBRUTINAS QUE UTILIZA: SE7, SA4 y SA5

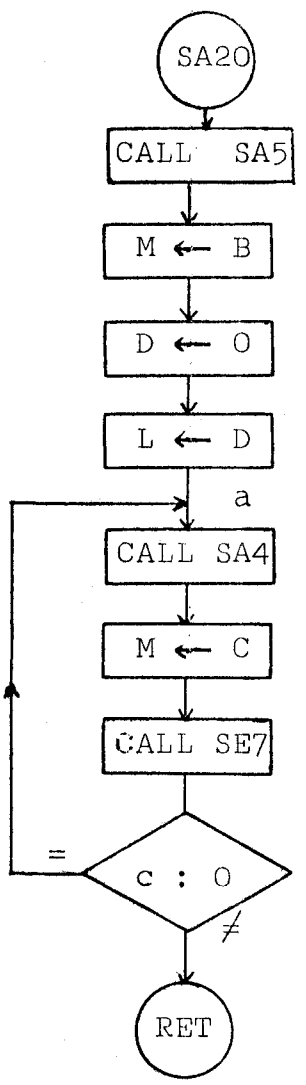
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 7/80

PARAMETROS DE ENTRADA: E

PARAMETROS DE SALIDA: -

PARAMETROS INTERMEDIOS: A, B, C, D, H y L



NOMBRE: SA21

OBJETIVO: Traslación positiva de FE una posición (periodo).

SUBRUTINAS QUE UTILIZA: SE7 y SA6

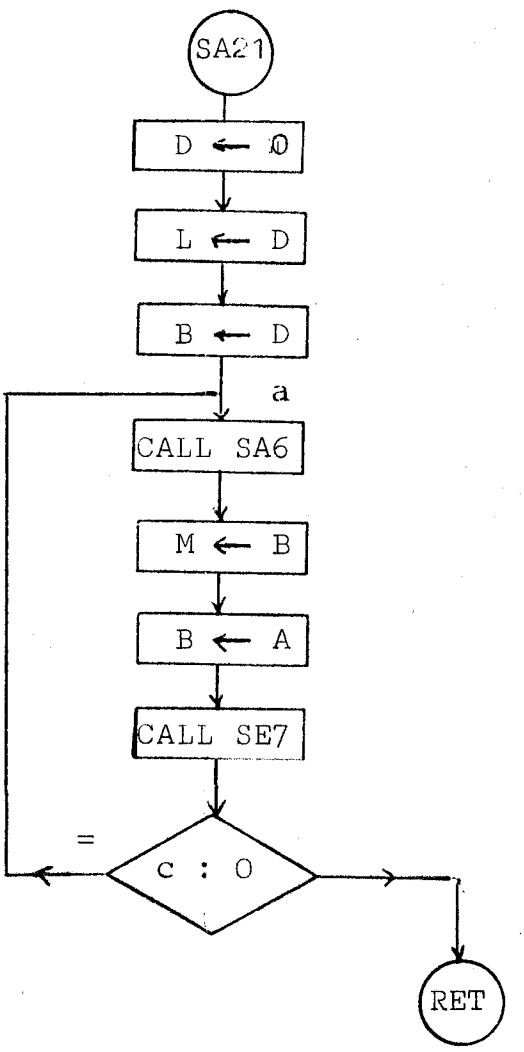
CODIGO DE LLAMADA:

DIRECCION DE ENTRADA: 7/98

PARAMETROS DE ENTRADA: E

PARAMETROS DE SALIDA: -

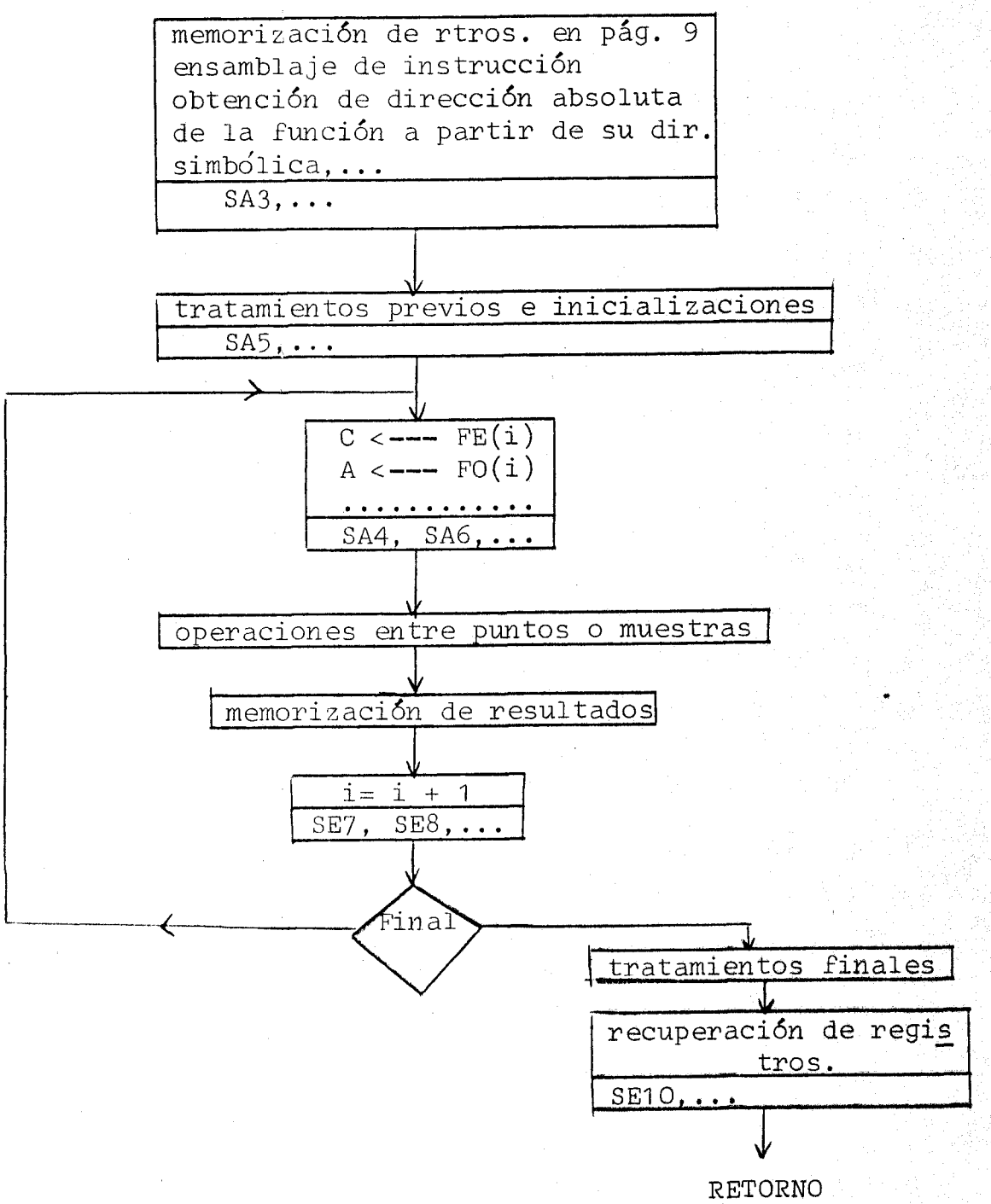
PARAMETROS INTERMEDIOS: A, B, D, H y L.



3.5.3.4 RUTINAS DEL LENGUAJE LPF

Antes de presentar los organigramas, se exponen las formas generales de la mayor parte de los algoritmos de las rutinas del traductor del LPF.

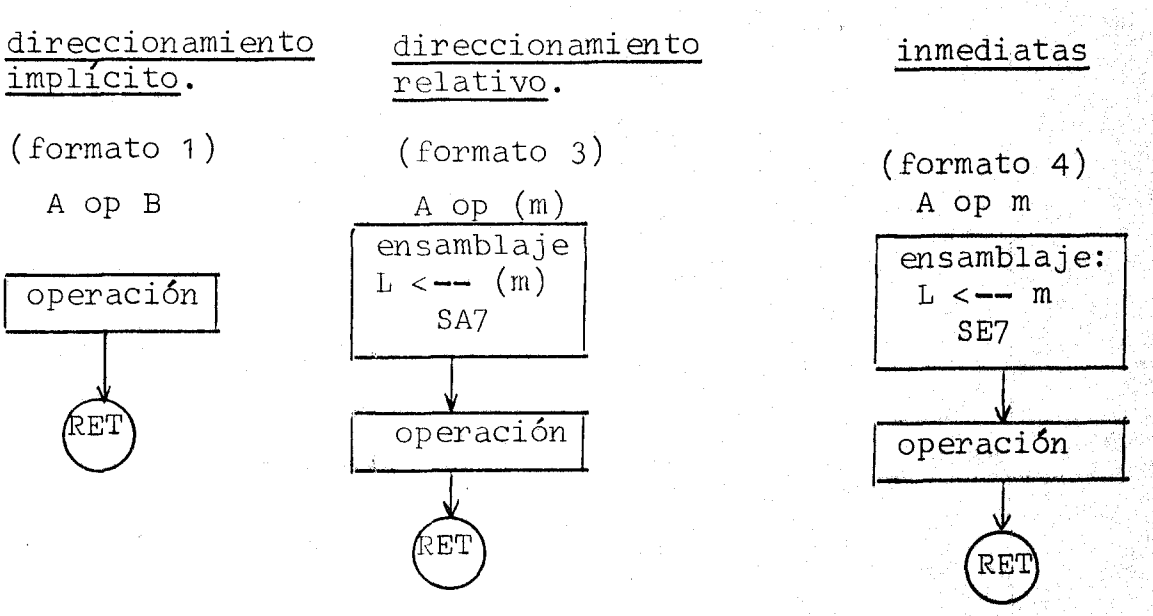
La forma más general de los algoritmos de las instrucciones para operar con funciones (instrucciones de formato tipo 2, secc. 2.3.1) es:



En diversas operaciones (adición, sustracción, integral,...) se prevee la posibilidad que dentro de una iteración se produzca un rebose ("overflow"); en este caso se comienza de nuevo todo el proceso iterativo, o en el caso de adición y sustracción se lanza un nuevo programa; evitandose el rebose y utilizando técnicas de redondeo adecuadas.

La mayoría de los programas utilizan con frecuencia las subrutinas auxiliares y elementales, lo cual hace conveniente que para una fácil compresión de los mismo se consulten las tablas 3.2. y 3.3..

La mayor parte de las instrucciones que no se realizan entre funciones tienen las siguientes formas generales:



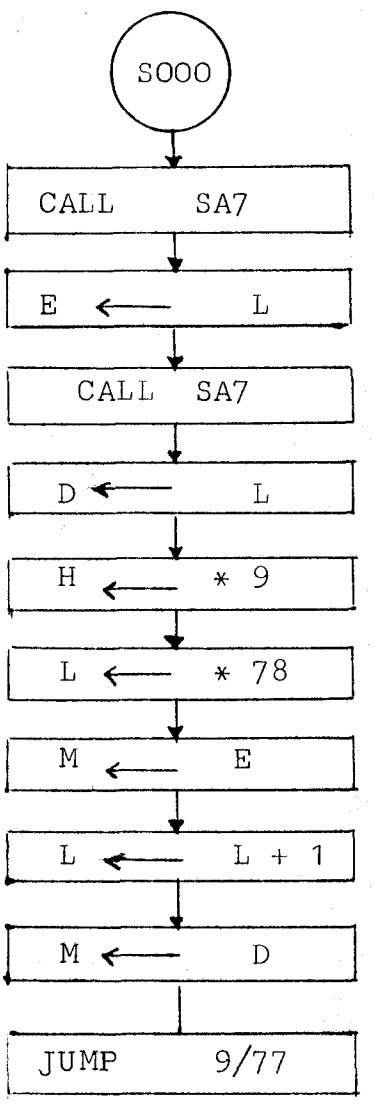
NOMBRE: SOOO

OBJETIVO: Intercalar rutinas escritas en LM dentro de un programa LPF.

SUBRUTINAS QUE UTILIZA: SA7

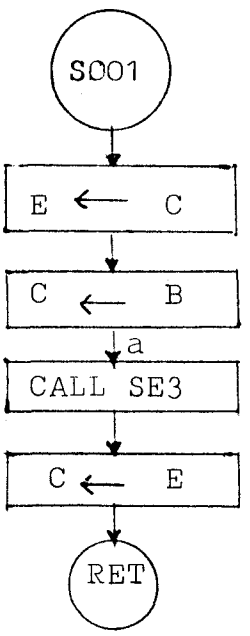
DIRECCION DE ENTRADA: 7/216

CODIGO: 1111-1011

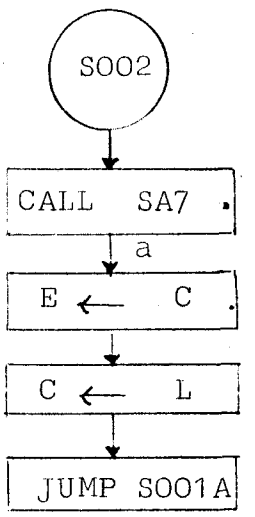


Explicación.- Llamando dos veces a la subrutina SA7 se obtienen en E y D, respectivamente, las direcciones baja y alta donde empieza la rutina en L.M. a ejecutar. Estas direcciones se escriben en las posiciones adecuadas de la página 9 para poder transferir el control a una dirección variable. La rutina en L.M. debe acabar con un RET, para retornar el control al supervisor del interprete (PM7).

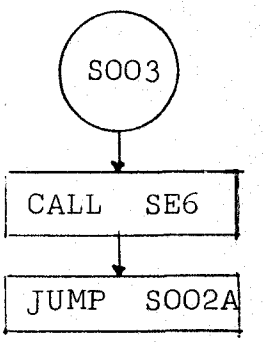
ADD A,B



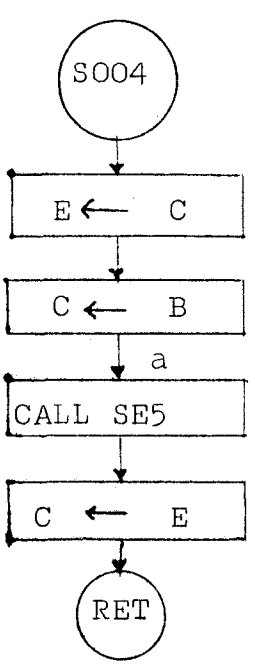
ADD A,(n)



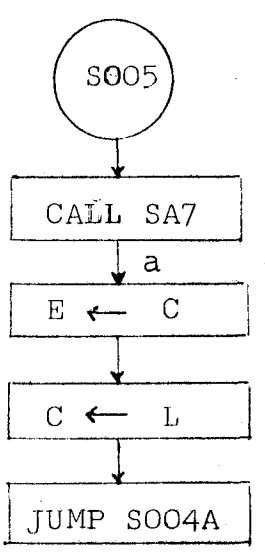
ADD A,*n



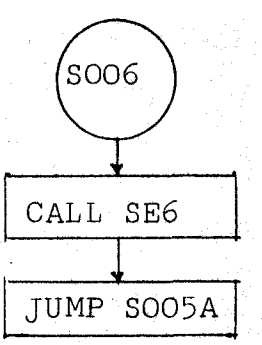
SUB A,B



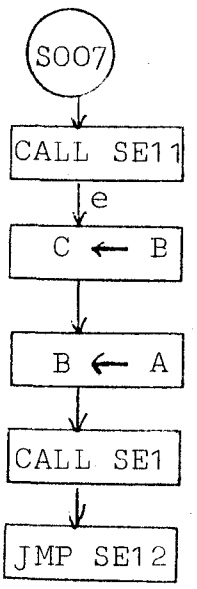
SUB A,(n)



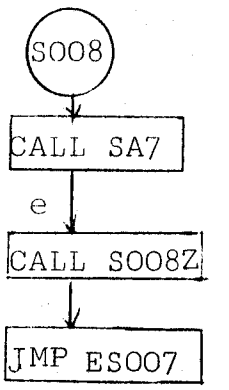
SUB A,*n



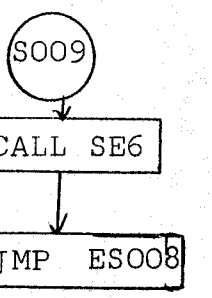
A<-- A*C



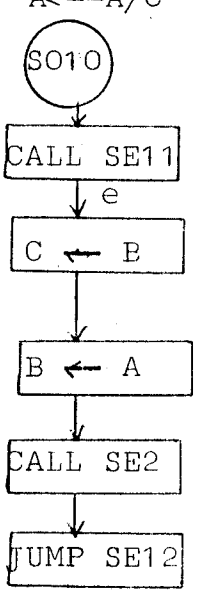
A<--Ax(n)



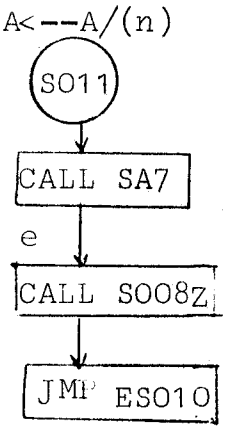
A<--A*n



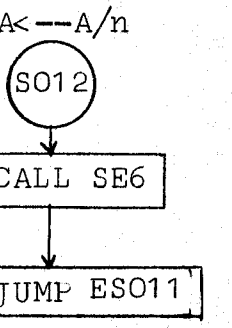
A<--A/C



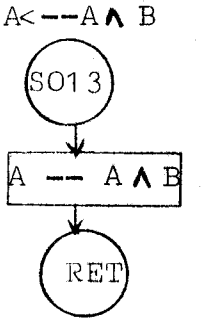
A<--A/(n)



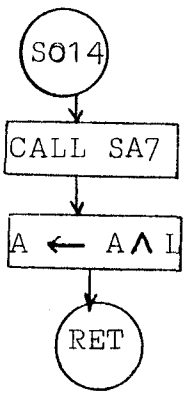
A<--A/n



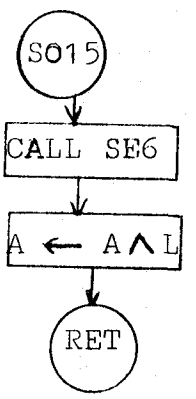
A<--A∧B



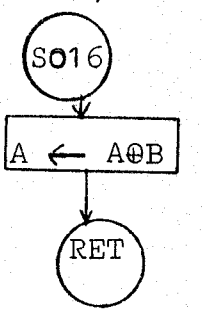
A<--A∧ (n)



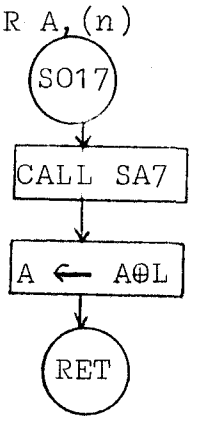
A<--A∧ *n



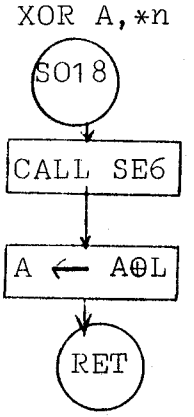
XOR A,B



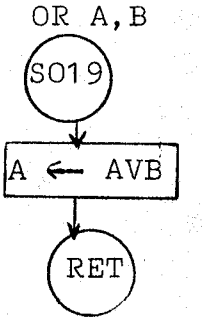
XOR A,(n)



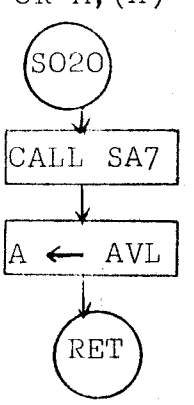
XOR A,*n

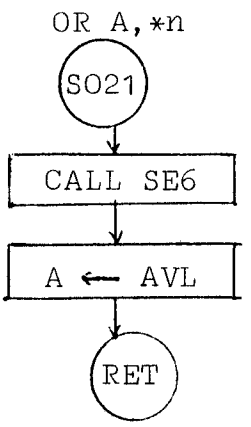


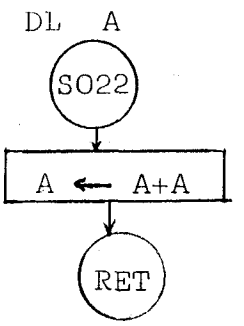
OR A,B

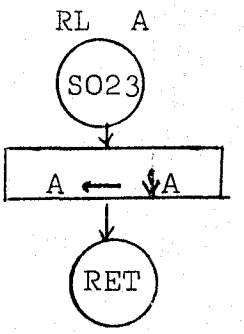


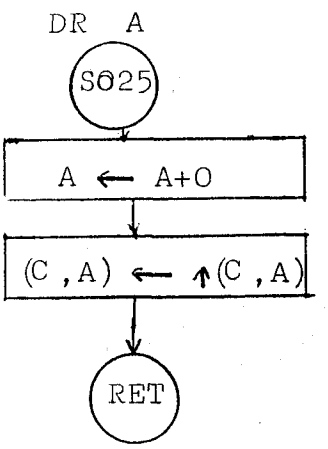
OR A,(n)

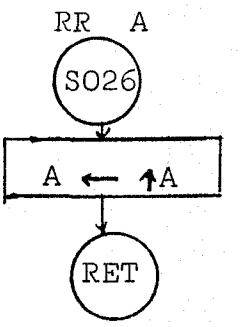


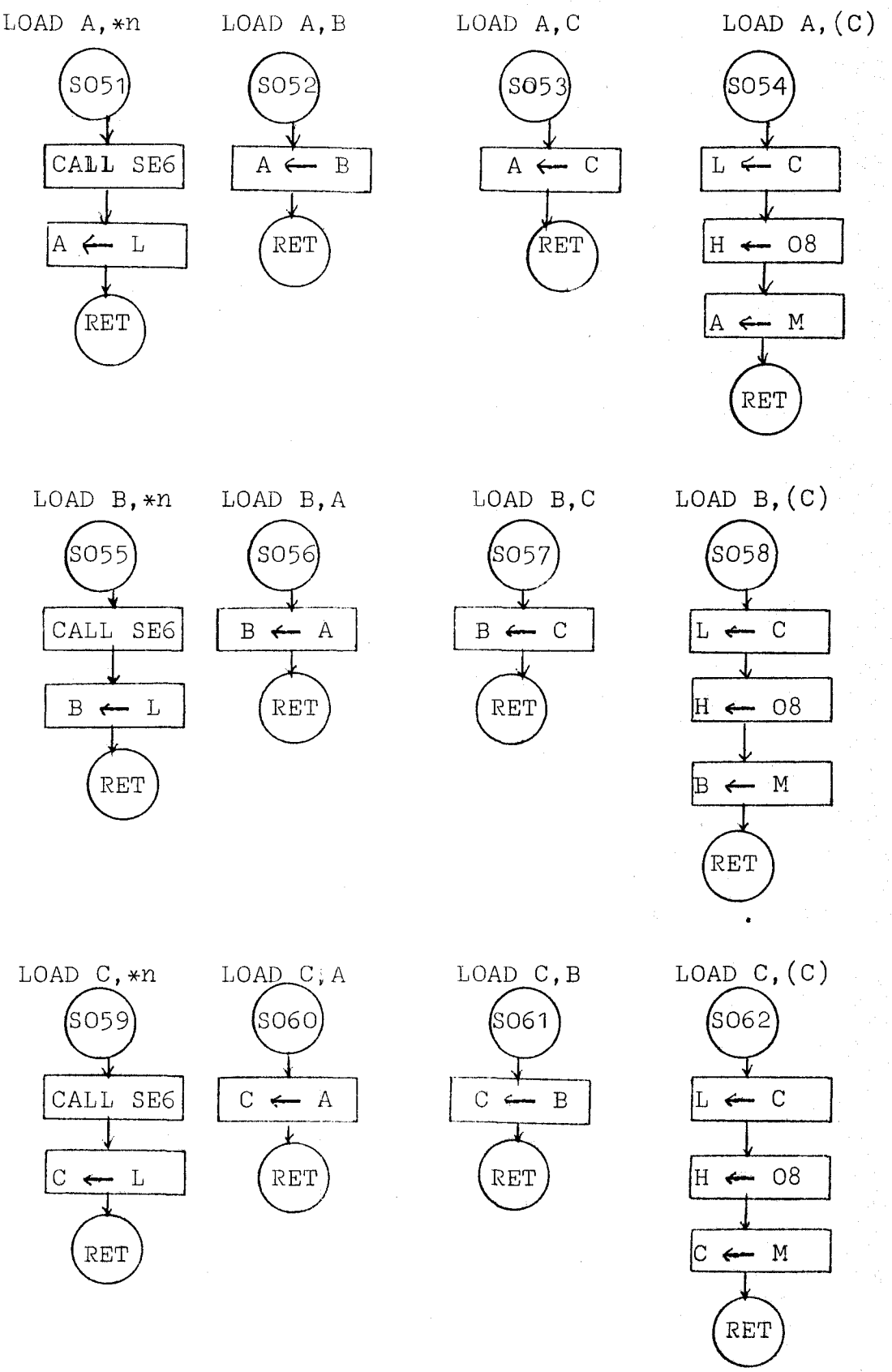

OR A,*n
S021
CALL SE6
A ← AVL
RET


DL A
S022
A ← A+A
RET

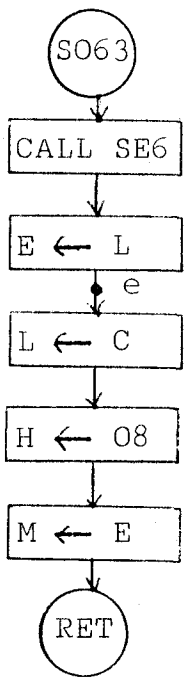

RL A
S023
A ← ↓A
RET


DR A
S025
A ← A+0
(C ,A) ← ↑(C ,A)
RET

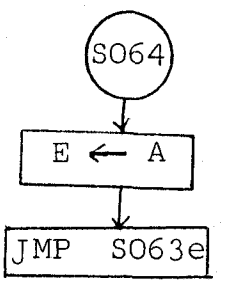

RR A
S026
A ← ↑A
RET


LOAD A,*n
S051
CALL SE6
A ← L
RET
LOAD A,B
S052
A ← B
RET
LOAD A,C
S053
A ← C
RET
LOAD A,(C)
S054
L ← C
H ← 08
A ← M
RET
LOAD B,*n
S055
CALL SE6
B ← L
RET
LOAD B,A
S056
B ← A
RET
LOAD B,C
S057
B ← C
RET
LOAD B,(C)
S058
L ← C
H ← 08
B ← M
RET
LOAD C,*n
S059
CALL SE6
C ← L
RET
LOAD C,A
S060
C ← A
RET
LOAD C,B
S061
C ← B
RET
LOAD C,(C)
S062
L ← C
H ← 08
C ← M
RET

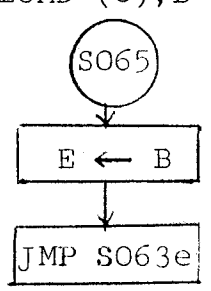
LOAD (C),*n



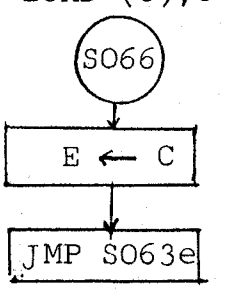
LOAD (C),A



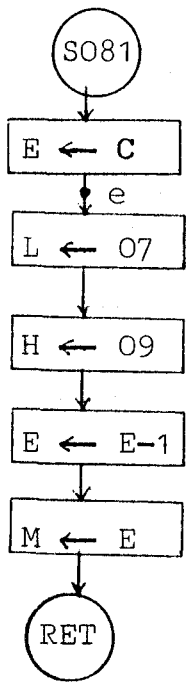
LOAD (C),B



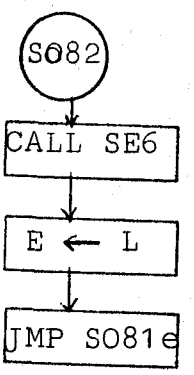
LOAD (C),C



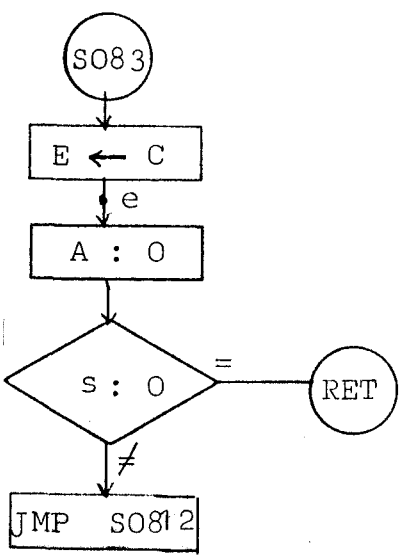
JUMP C



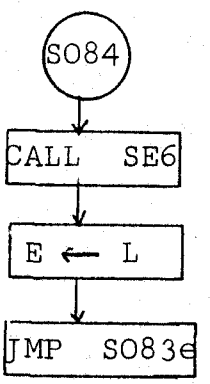
JUMP n



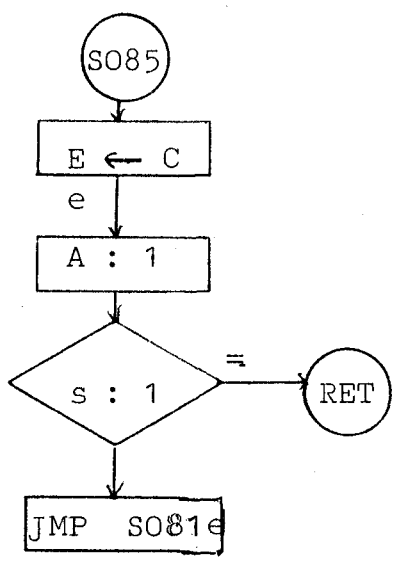
JUMP AN,C



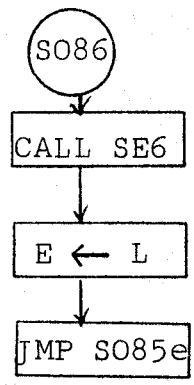
JUMP AN,n



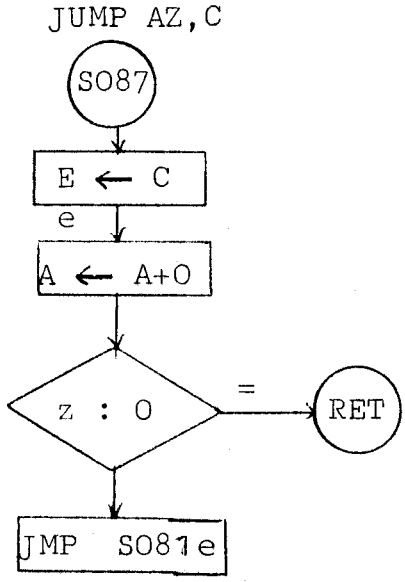
JUMP AP,C



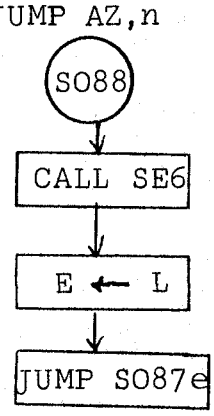
JUMP AP,n



JUMP AZ,C



JUMP AZ,n



NOMBRE: SO91 y SO92

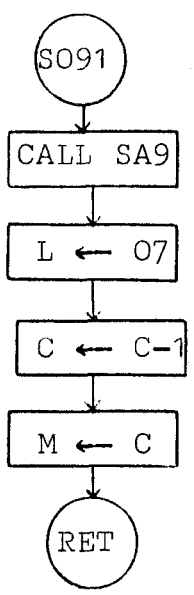
OBJETIVO: - llamada a una subrutina,C.
- llamada a una subrutina,n

SUBRUTINAS QUE UTILIZA: SA9; SSA9; SO81e

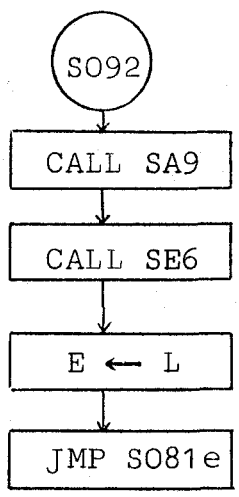
CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 4/72; 4/104

CALL C



CALL *n



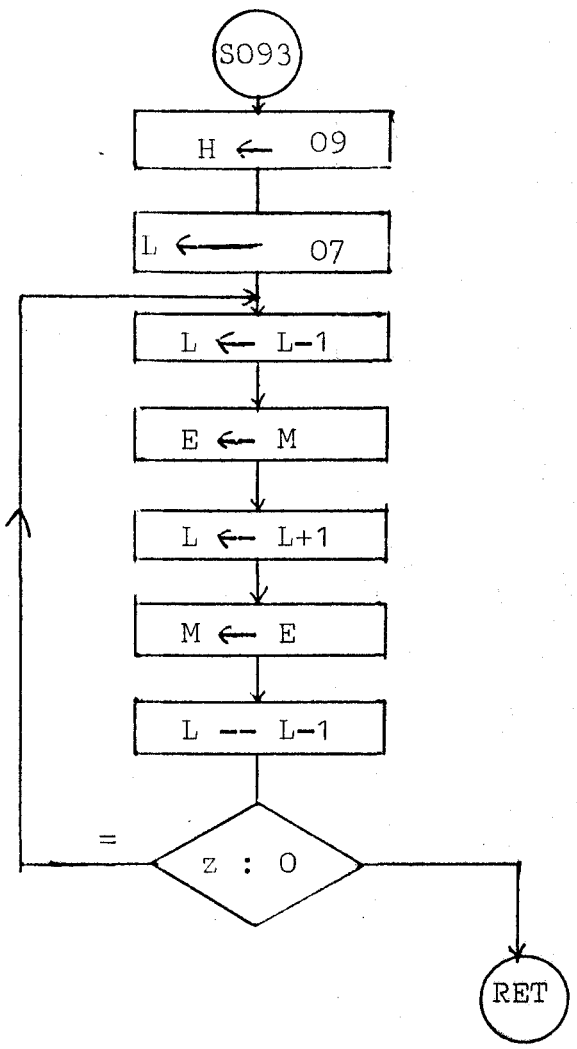
NOMBRE: S093

OBJETIVO: retorno de rutina

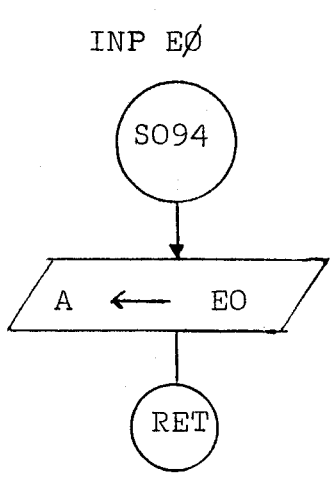
SUBRUTINAS QUE UTILIZA: ninguna

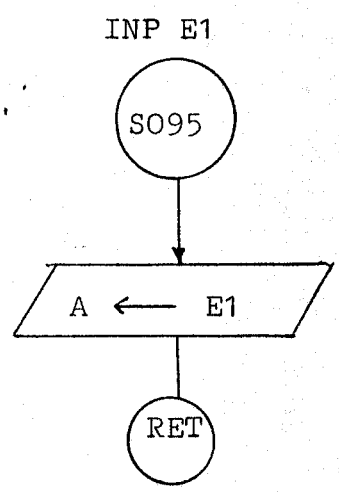
CODIGO: Nemotécnico:
Binario:

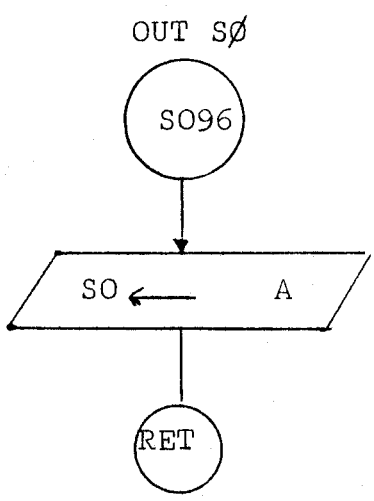
DIRECCION DE ENTRADA: 4/80

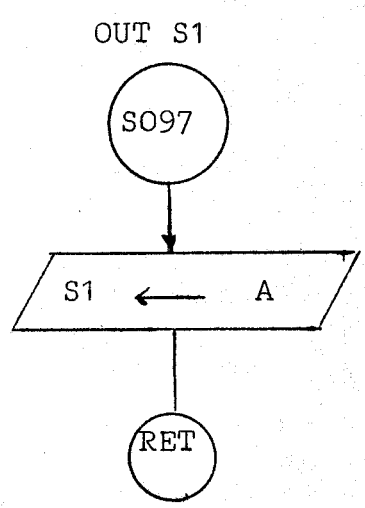


ver sección 3.5.1


INP EØ
SO94
A ← EO
RET


INP E1
SO95
A ← E1
RET


OUT SØ
SO96
SO ← A
RET


OUT S1
SO97
S1 ← A
RET

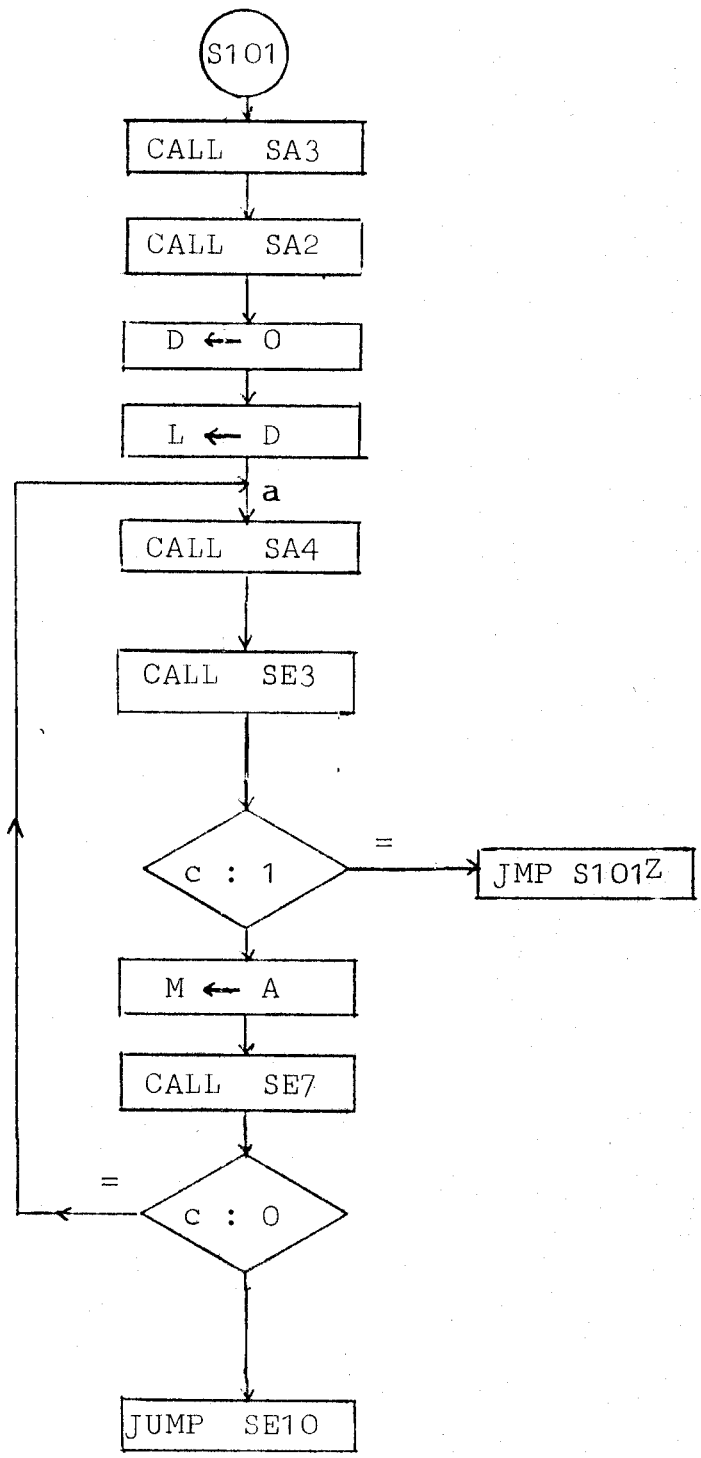
NOMBRE: S101

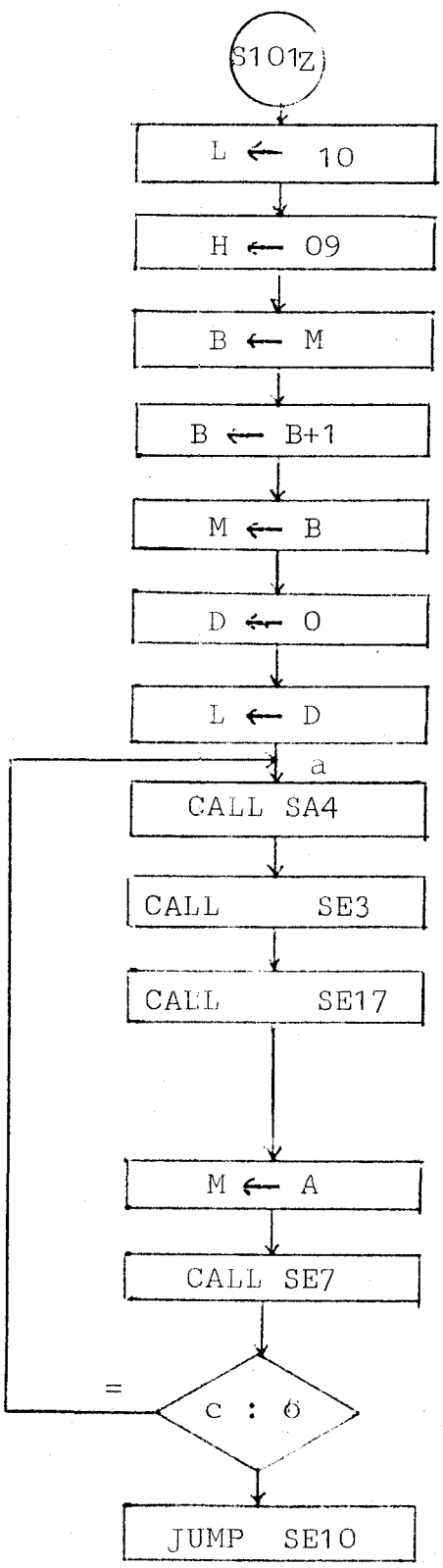
OBJETIVO: suma de funciones.

SUBRUTINAS QUE UTILIZA:SA3, SA2,SA4,SE7,SE10, S101Z,SE3,SE17

CODIGO: Nemotécnico: ADD F FO,Fn
Binario:

DIRECCION DE ENTRADA 1/Ø




S101Z
L ← 10
H ← 09
B ← M
B ← B+1
M ← B
D ← 0
L ← D
a
CALL SA4
CALL SE3
CALL SE17
M ← A
CALL SE7
=
C : 6
JUMP SE10

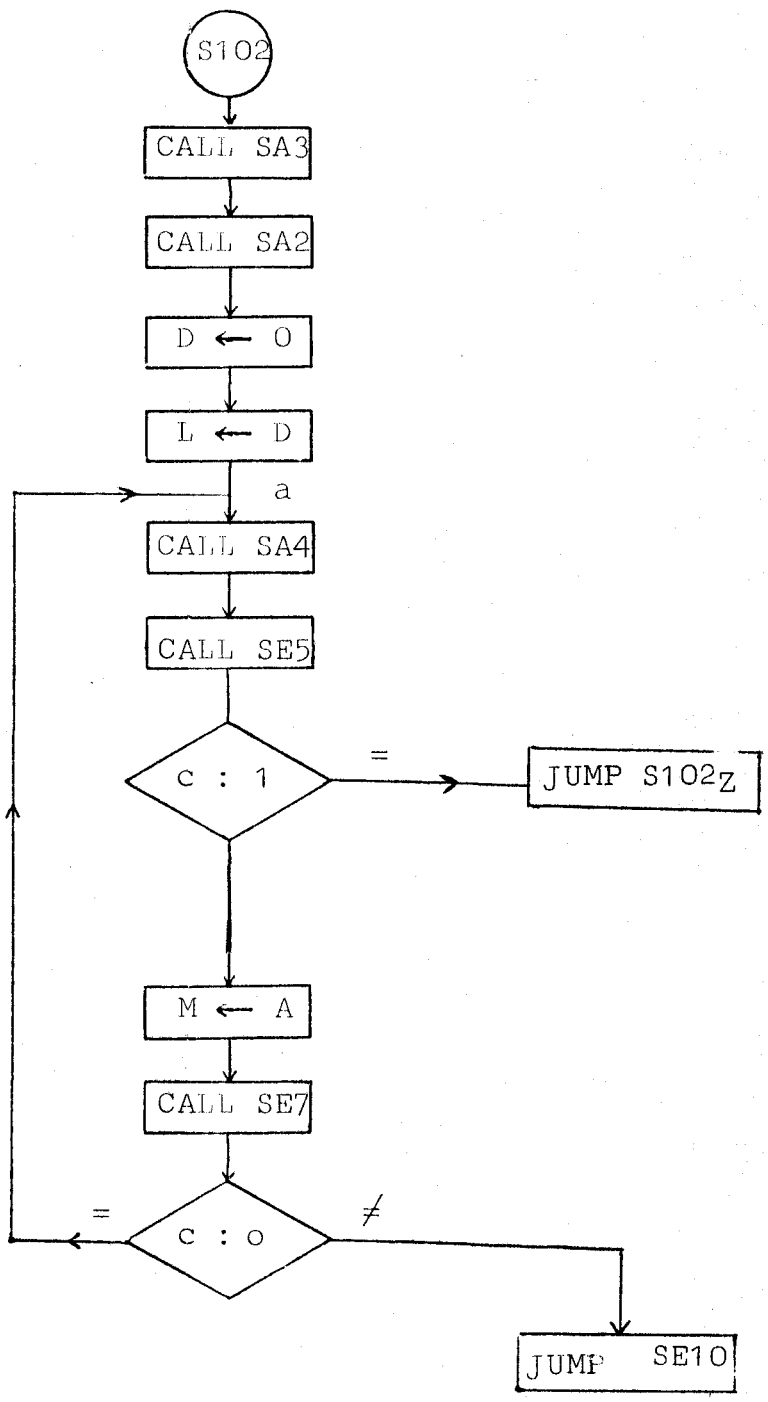
NOMBRE: S102

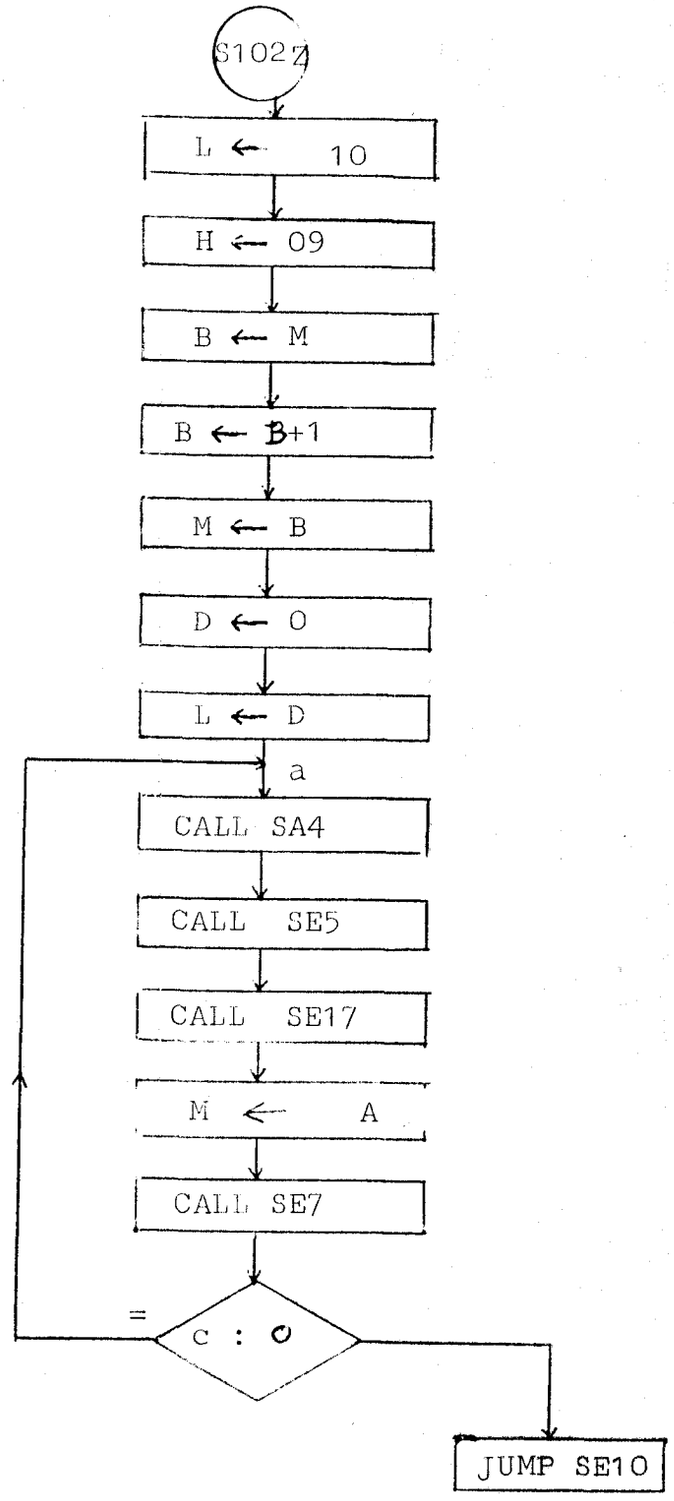
OBJETIVO: resta de funciones

SUBRUTINAS QUE UTILIZA: SA2, SA3, SA4, SE7, SE10, S102Z, SE17, SE5,

CODIGO: Nemotécnico: SUBF FO,Fn
Binario:

DIRECCION DE ENTRADA: 1/64




S102Z
L ← 10
H ← 09
B ← M
B ← B+1
M ← B
D ← 0
L ← D
a
CALL SA4
CALL SE5
CALL SE17
M ← A
CALL SE7
=
c : 0
JUMP SE10

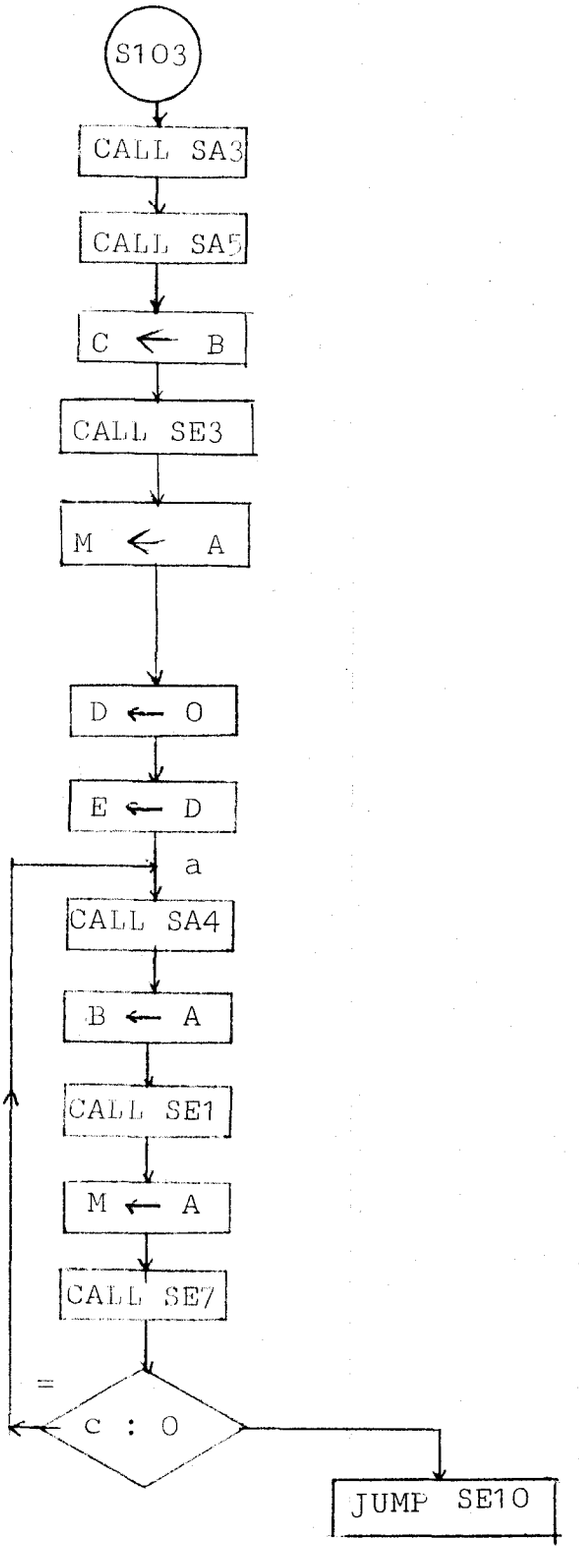
NOMBRE: S103

OBJETIVO: multiplicación de funciones

SUBRUTINAS QUE UTILIZA: SA3, SA5, SA4, SE1, SE7, SE13, SE3, SE10,

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 1/120



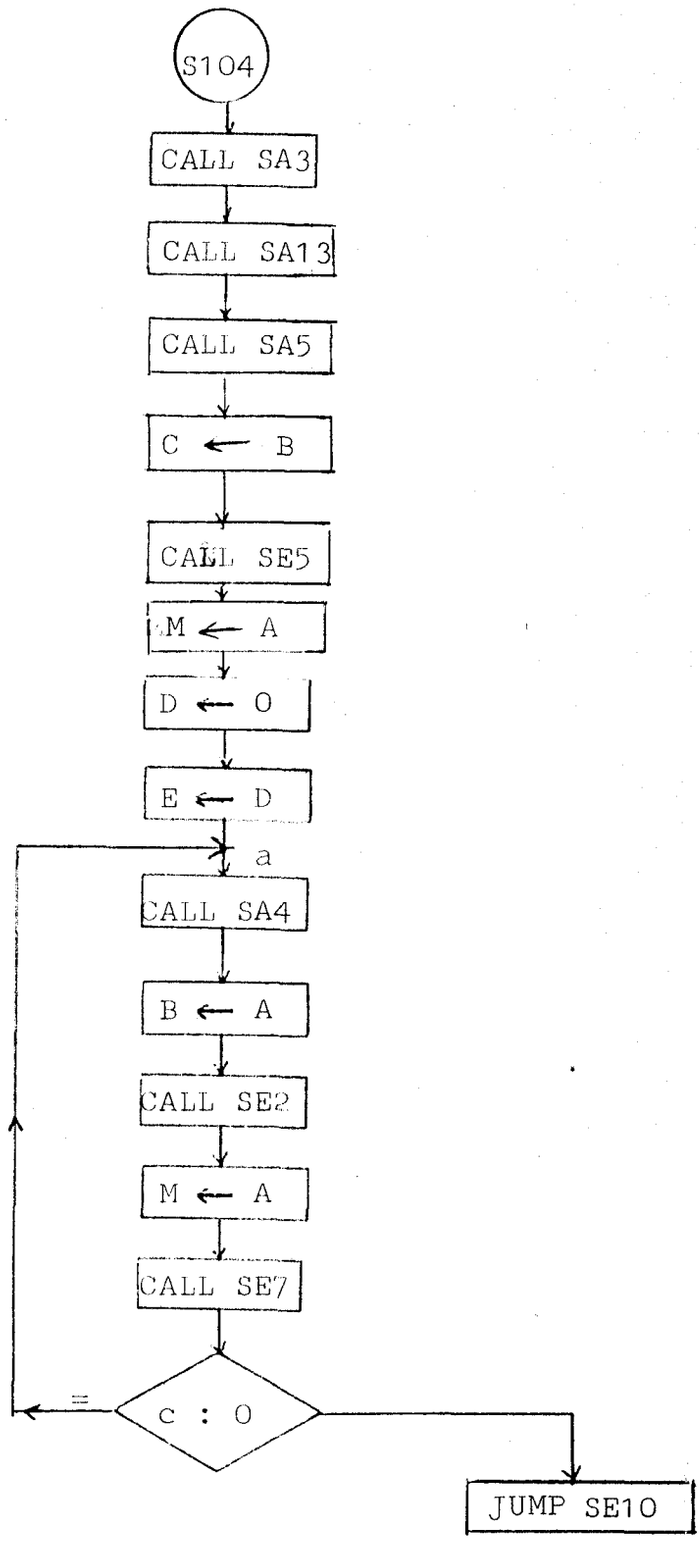
NOMBRE: S104

OBJETIVO: división de funciones

SUBRUTINAS QUE UTILIZA: SA3, SA13, SA5, Sb5, SA4, SE7, SE10,

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 1/152



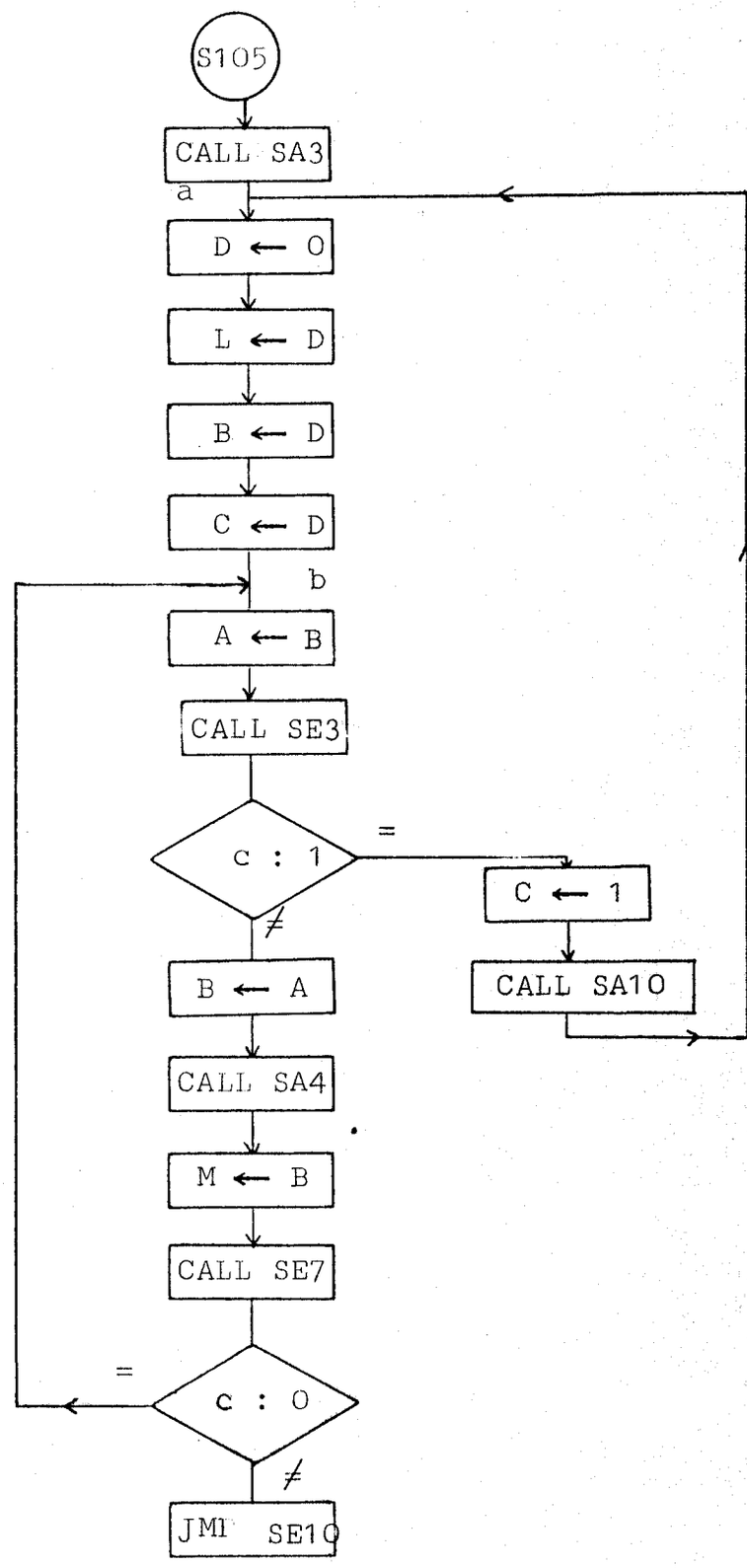
NOMBRE: S105

OBJETIVO: integral de una función.

SUBRUTINAS QUE UTILIZA: SA3, SE3, SA4, SE7, SE10, SA10,

CODIGO: Nemotécnico
Binario:

DIRECCION DE ENTRADA: 1/184


S105
CALL SA3
a
D ← 0
L ← D
B ← D
C ← D
b
A ← B
CALL SE3
c : 1
=
C ← 1
CALL SA10
≠
B ← A
CALL SA4
M ← B
CALL SE7
=
c : 0
≠
JMI SE10

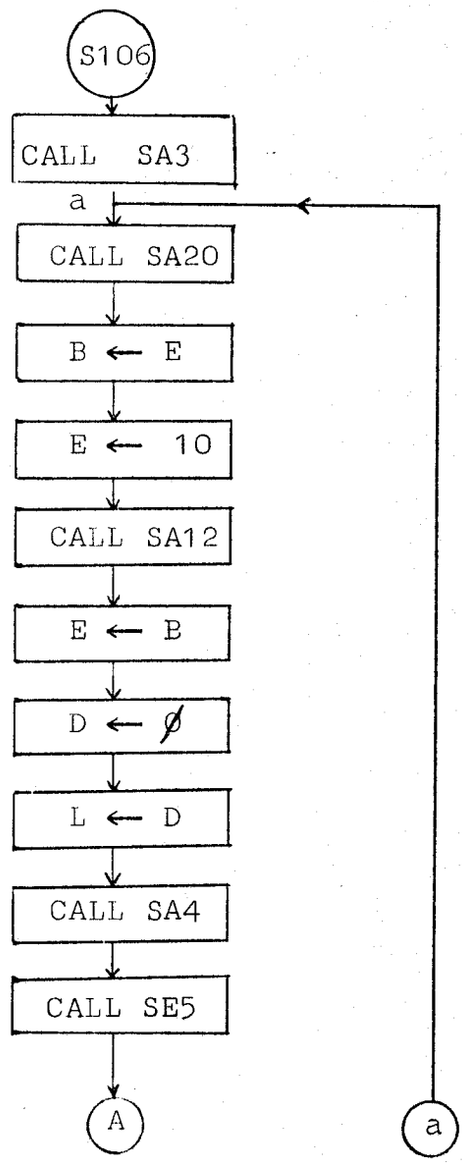
NOMBRE: S106

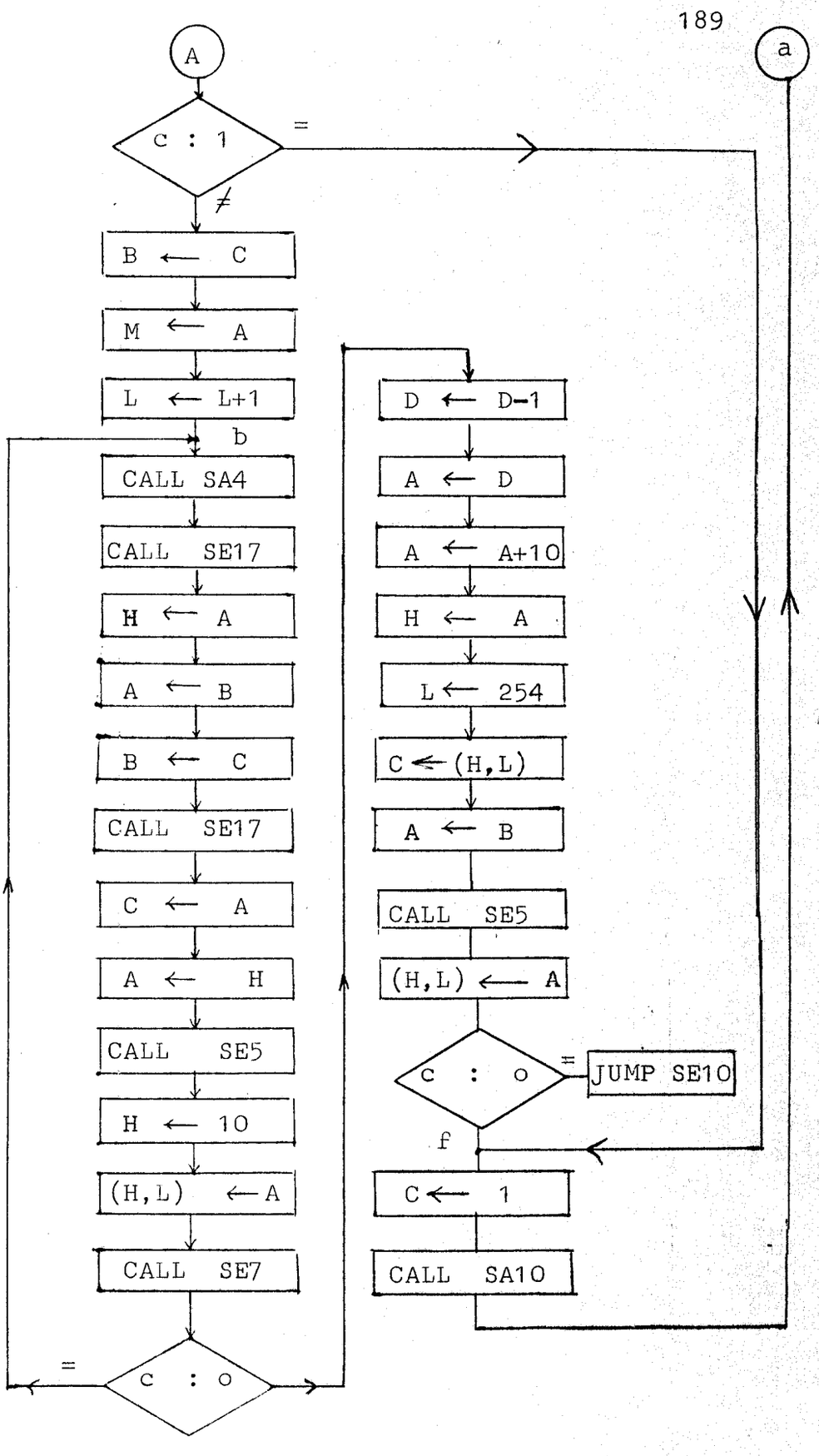
OBJETIVO: derivar una función

SUBRUTINAS QUE UTILIZA: SA10, SA3, SA20, SA12, SE10, SA4, SE17, SE5, SE7,

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 1/224




A
a
C : 1
=
≠
B ← C
M ← A
L ← L+1
b
CALL SA4
CALL SE17
H ← A
A ← B
B ← C
CALL SE17
C ← A
A ← H
CALL SE5
H ← 10
(H,L) ← A
CALL SE7
=
C : 0
D ← D-1
A ← D
A ← A+10
H ← A
L ← 254
C ← (H,L)
A ← B
CALL SE5
(H,L) ← A
C : 0
=
JUMP SE10
f
C ← 1
CALL SA10

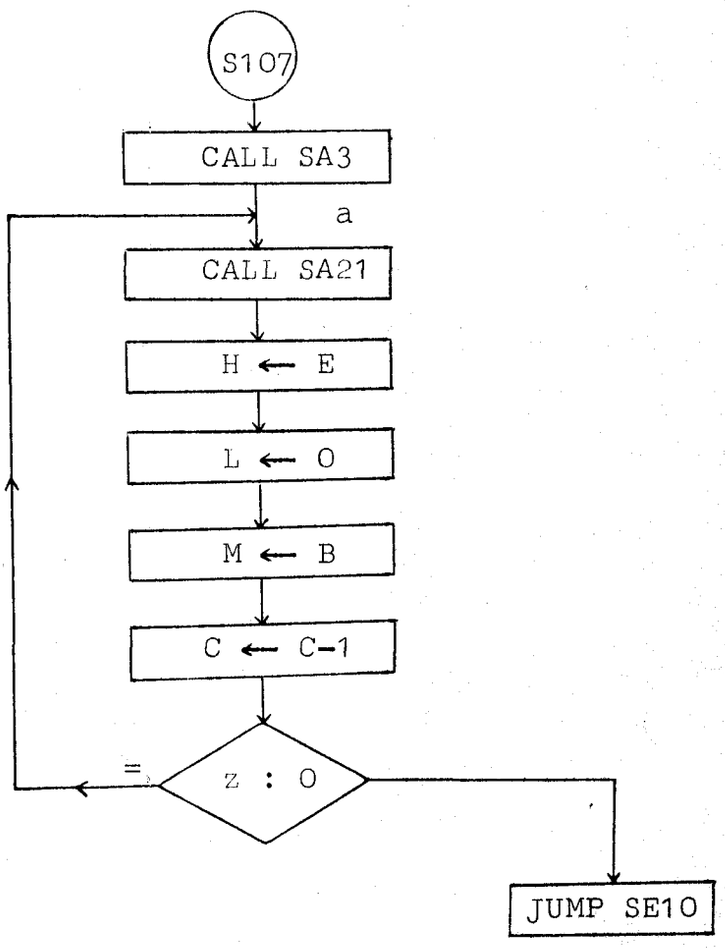
NOMBRE: S107

OBJETIVO: rotación positiva de una función en el tiempo

SUBRUTINAS QUE UTILIZA: SA3, SA21, SE10.

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 2/48



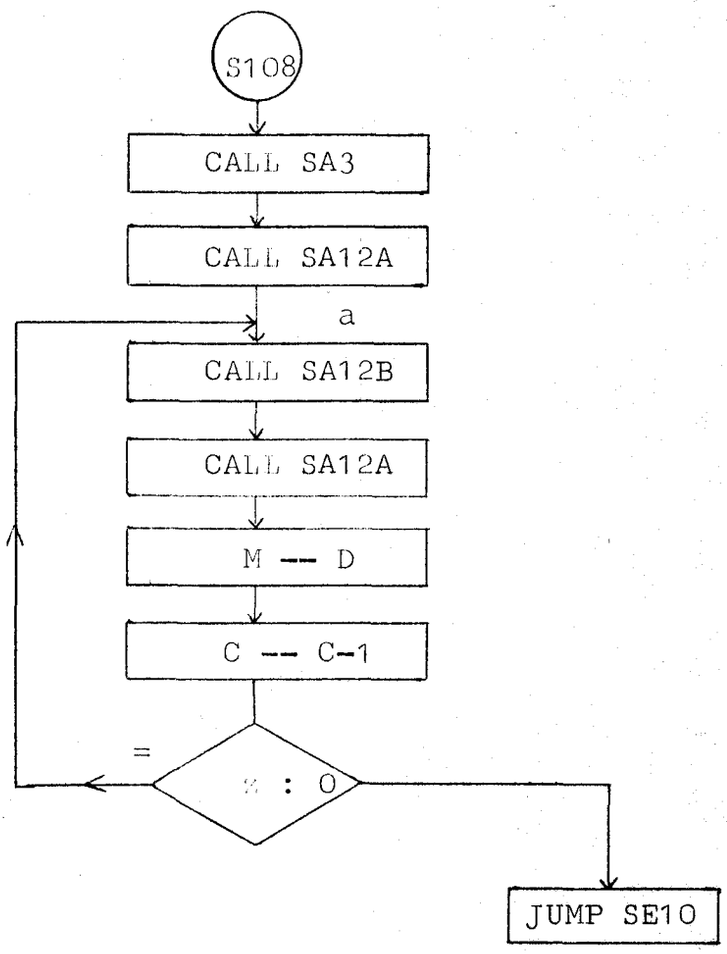
NOMBRE: S108

OBJETIVO: rotación negativa de una función en el tiempo

SUBRUTINAS QUE UTILIZA: SA3, SA12A, SA12B, SE10.

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 2/72



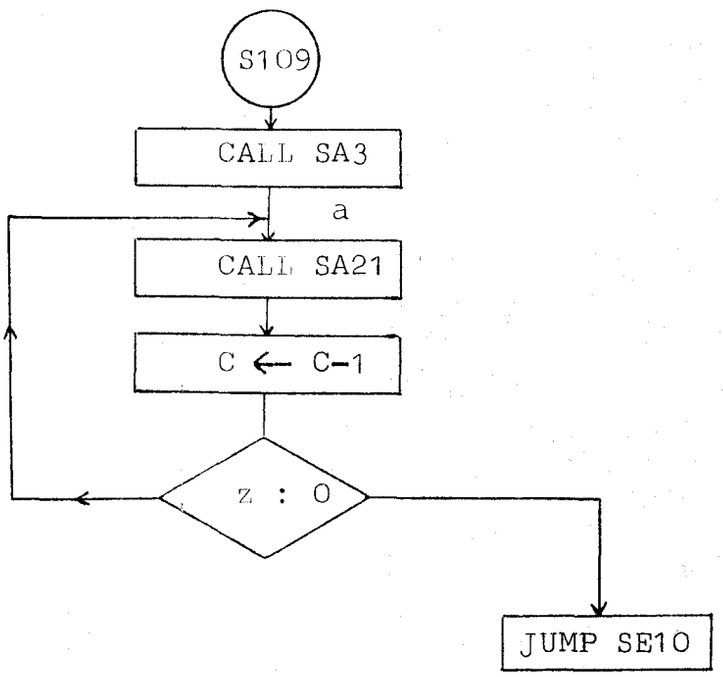
NOMBRE: S109

OBJETIVO: traslación positiva de una función.

SUBRUTINAS QUE UTILIZA: SA3, SA21, SE10,

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 2/96



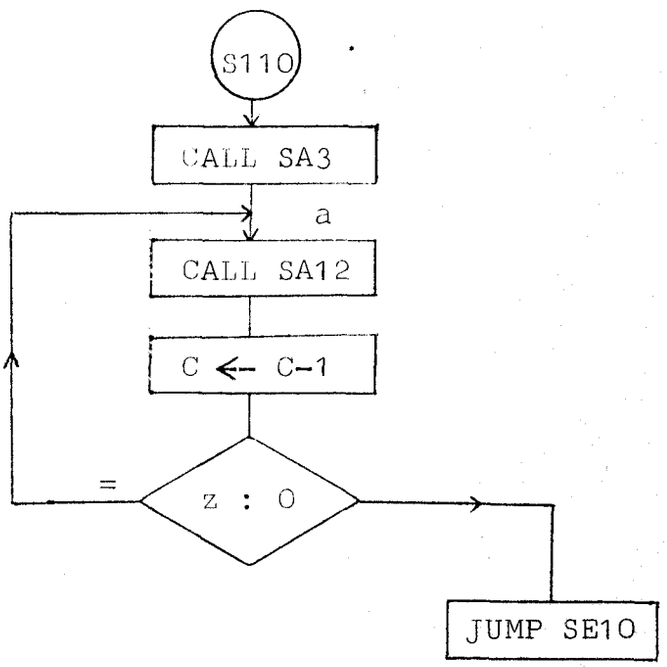
NOMBRE: S110

OBJETIVO: traslación negativa de una función en el tiempo

SUBRUTINAS QUE UTILIZA: SA3, SA12, SE10,

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 2/112



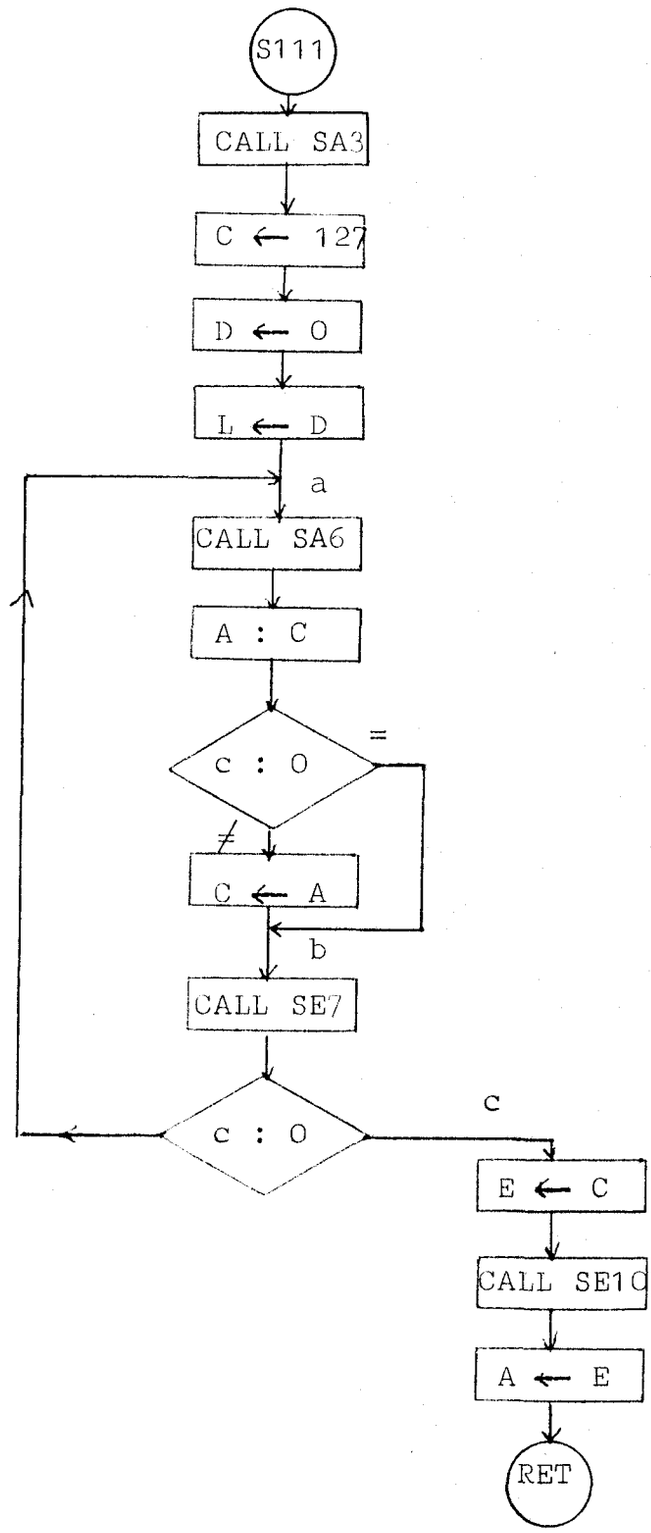
NOMBRE: S111

OBJETIVO: mínimo de una función

SUBRUTINAS QUE UTILIZA: SA3, SA6, SE7, SE10

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 2/128



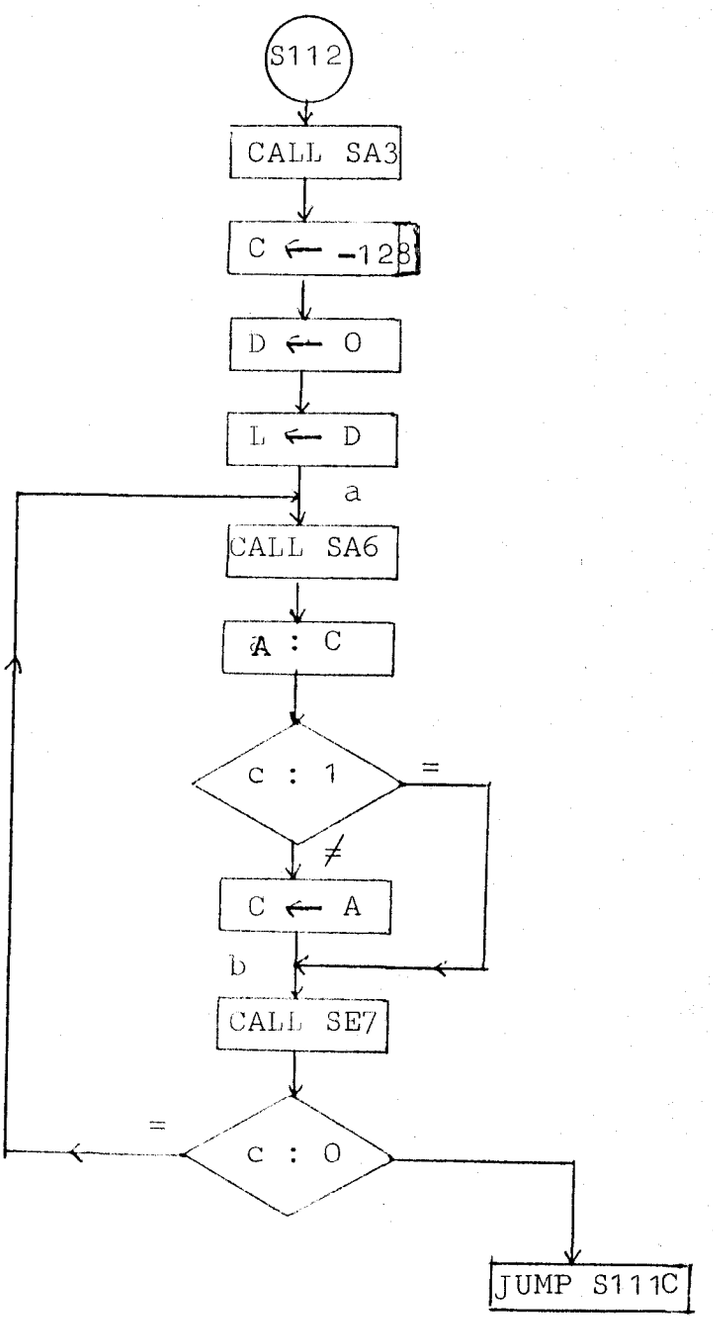
NOMBRE: S112

OBJETIVO: máximo de una función

SUBRUTINAS QUE UTILIZA: SA3, SA6, SE7, S111C

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 2/160



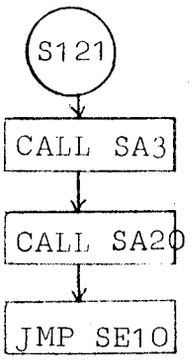
NOMBRE: S121

OBJETIVO: trasferencia de una función, Fn, a la función FØ.

SUBRUTINAS QUE UTILIZA: SA3, SA20, SE10.

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 2/192



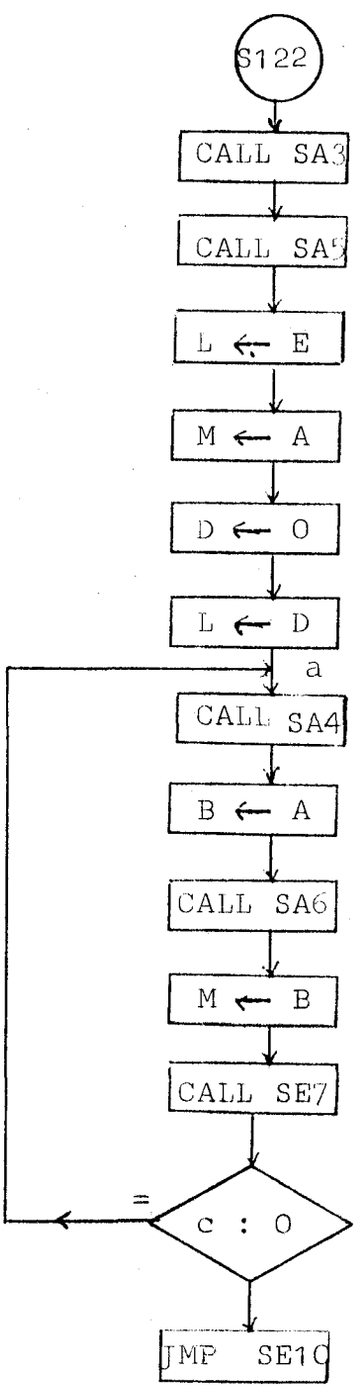
NOMBRE: S122

OBJETIVO: trasferencia de la función FØ a otra, Fn.

SUBRUTINAS QUE UTILIZA: SA3, SA5, SA4, SE7, SE10, SE6

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 2/208



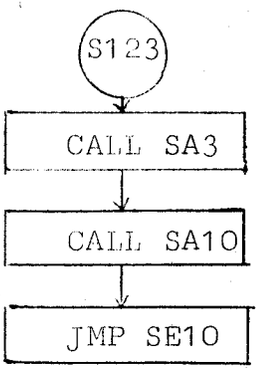
NOMBRE: S123

OBJETIVO: desplazamiento de todos los puntos de una función un lugar a la derecha con modificación de exponentes.

SUBRUTINAS QUE UTILIZA: SA3, SA10, SE10,

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 2/240



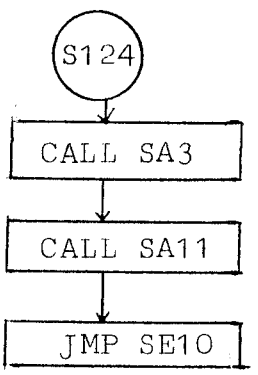
NOMBRE: S124

OBJETIVO: desplazamiento de los puntos de una función a la izquierda con modificación de exponente.

SUBRUTINAS QUE UTILIZA: SA3, SA11, SE10.

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 3/Ø



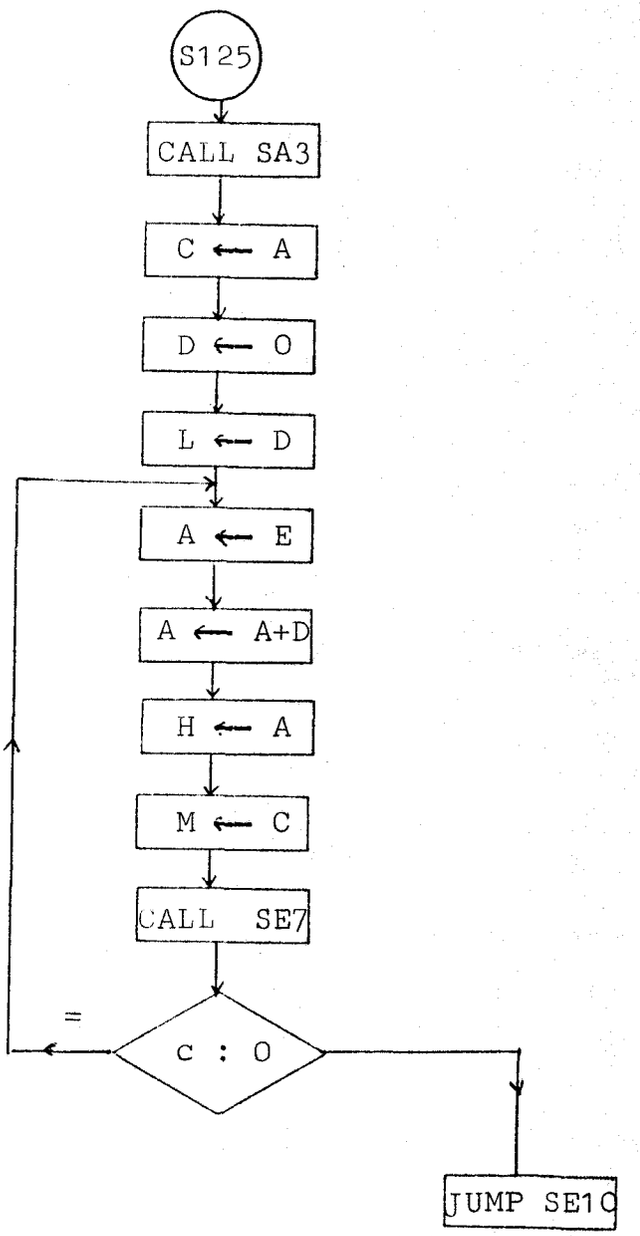
NOMBRE: S125

OBJETIVO: generación de una función constante

SUBRUTINAS QUE UTILIZA : SA3, SE7, SE10

CODIGO: Nemotécnico: LOADF Fn,A
Binario:

DIRECCION DE ENTRADA: 3/16



NOMBRE: S126 y S127

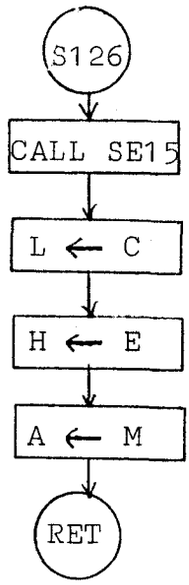
OBJETIVO: -leer un octeto de una página de una función

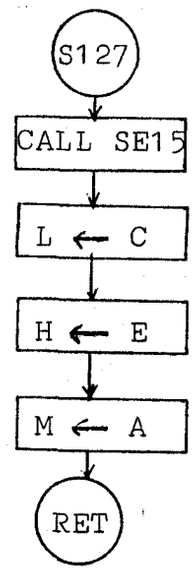
- escribir en memoria un octeto de una página de una función.

SUBRUTINAS QUE UTILIZA: SE15

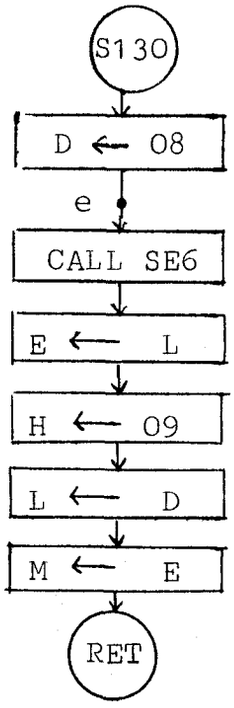
CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 3/40: 3/48

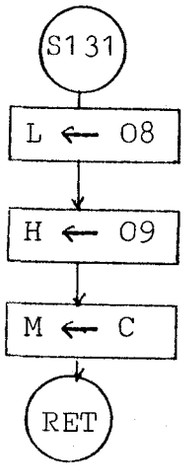




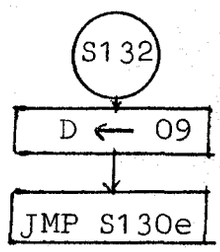
LOADF PNP,*n



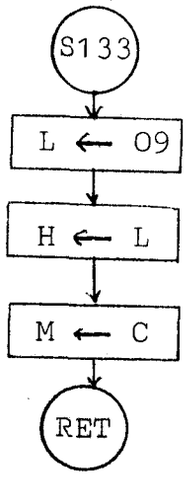
LOADF PNP,C



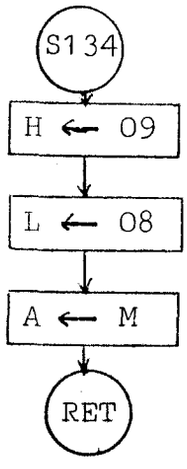
LOADF PR,*n



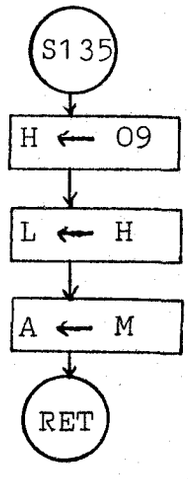
LOADF PR,C



LOADF A,PNP



LOADF A,PR



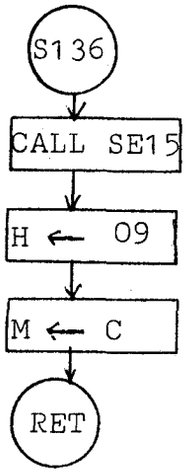
NOMBRE: S136 y S137

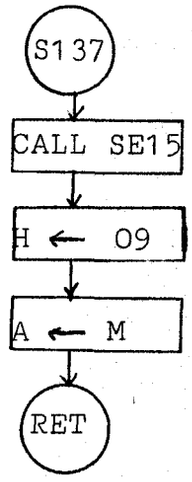
OBJETIVO: - Definición de un exponente, xFn, de una función Fn,
- Obtención en el acumulador de un exponente de una función, Fn.

SUBRUTINAS QUE UTILIZA: SE15,

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 3/112 3/120





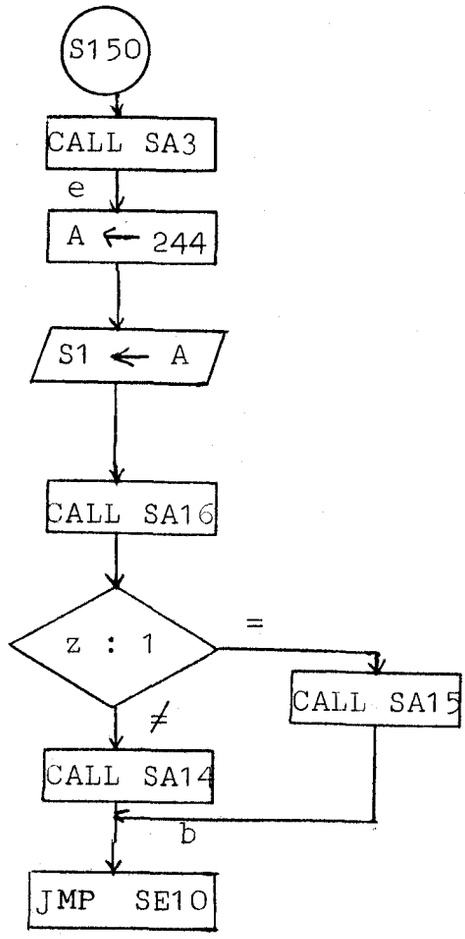
NOMBRE: S150 y S151

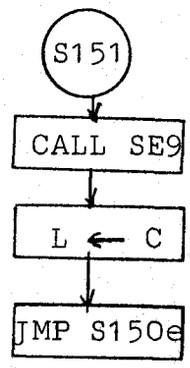
OBJETIVO: - entrada y memorización de una función, Fn
- entrada y memorización de una función, FC.

SUBRUTINAS QUE UTILIZA: SA3, SA16, SA14, SA15, SE10, SE9, S150E

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 3/128; 3/152





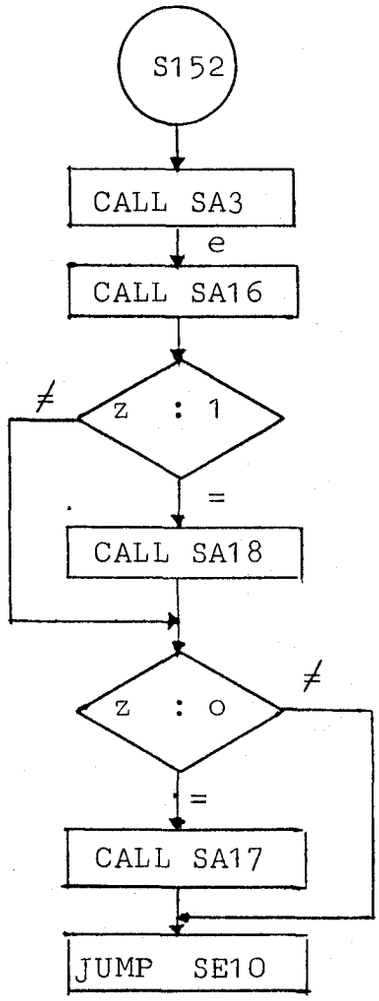
NOMBRE: S152 y S153

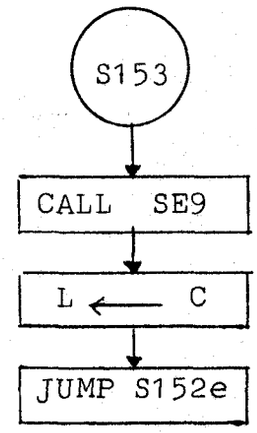
OBJETIVO: - Salida de una función Fn, memorizada.
- Salida de una función FC, almacenada en memoria.

SUBRUTINAS QUE UTILIZA: SA3, SA16, SA18, SA17, SE10, SE9, S152E

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 3/160; 3/184





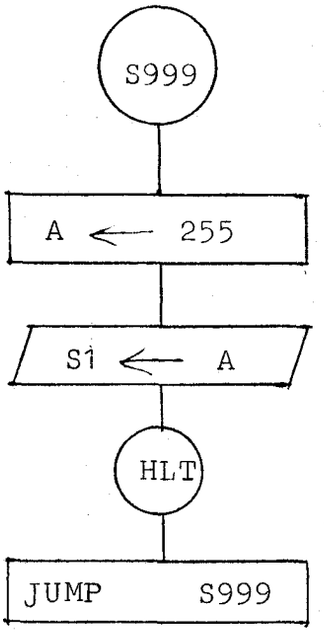
NOMBRE: S999

OBJETIVO: Final del programa.

SUBRUTINAS QUE UTILIZA: ninguna.

CODIGO: Nemotécnico:
Binario:

DIRECCION DE ENTRADA: 7/248



3.6 PROGRAMAS, IMBRICACION, DIRECTORIO Y GRABACION DEL SISTEMA OPERATIVO:

En el apéndice III pueden verse la totalidad de programas que constituyen el S.O..

Estos programas están grabados en 8 pastillas de memoria REPROM de 256 x 8 bits cada una; es decir, cada pastilla corresponde a una página.

La grabación se ha efectuado con un "Programador de memorias muertas" de "Data I/O Corporation" módelo 5H. En este dispositivo la información a grabar se da en notación hexadecimal.

En el apéndice se presentan los programas tal y como han sido grabados en la memoria. La primera columna representa la dirección de memoria en decimal y la segunda la información memorizada en hexadecimal. Las otras columnas contienen el ensamblador y la dirección de la instrucción en hexadecimal.

Los programas y subrutinas se pueden clasificar en externos e internos, según que se puedan acceder o no a elbsdirectamente desde los lenguajes LC o LPF.

Para fijar las posiciones que ocupa cada programa dentro de la memoria se han seguido los siguientes criterior:

(1) los programas monitores, secc. 1.5.2.2., comienzan en las direcciones 0, 8, 16, 24, 32, 40, 48 y 56 de la primera página (pág. 0).

(2) todos los demás programas externos pueden comenzar (secc. 1.4.3) en una dirección arbitraria con la única condición de que sea la primera de una de las 256 páginas de 8 octetos en que se divide la totalidad de memoria REPROM.

(3) para la mayor parte de las instrucciones del lenguaje LPF en las que no varía, en su lenguaje nemotécnico, el código sino sus operadores (instrucciones de transferencia de información entre registros, por ejemplo), se ha procurado, para facilitar el trabajo de programación, que exista un método sistemático en la formación de su código binario con lo que la dirección de comienzo de las macros correspondientes queda determinada unívocamente por dicho código (secc. 1.4.2.3). Así, por ejemplo: LOAD d,s tiene de código binario: 1001 DDSS (secc. 2.3.2)
esta codificación implica la fijación de asignación de memoria para las macros correspondientes.

(4) para las subrutinas internas no existe ninguna ligadura en cuanto a las posiciones donde deben ir grabadas.

(5) para lograr un aprovechamiento más óptimo se han utilizado instrucciones de salto (JUMP) . cuando un programa debía, por razones de imbricación, fragmentarse en varias partes. No obstante estas instrucciones de salto se han utilizado lo menos posible, ya que según se dijo en la secc. 3.2.2, implican una pérdida de tres octetos de memoria y un tiempo adicional de 55 microsegundos de procesamiento.

De estos cinco puntos se deduce el orden que se ha seguido para la asiganción de memoria a los programas; el orden es el siguiente:

1º) programas monitores.
2º) programas externos cuyo código binario sigue un método de formación.
3º) resto de programas externos.
4º) programas internos de mayor número de instrucciones, asignandoles las mayores zonas de memoria libres.
5º) programas internos restantes ocupando las zonas de memoria libres.

En el apéndice IV se da el directorio de puntos de entrada. Las líneas señaladas con un asterisco indican el comienzo de un programa o un punto de entrada.

Algunos datos de interés que se deducen de los apéndices citados son:

- número total de programas grabados: 138
- capacidad de memoria ocupada: 2030
- capacidad de memoria utilizada realmente: 1852
- nº de instrucciones de salto de imbricación: 5
- nº de octetos de saltos de imbricación: 15
- número medio de octetos por rutina: 13,3
- rendimiento de la capacidad de memoria:

$$\frac{\text{capacidad de memoria ocupada}}{\text{capacidad de memoria total}} = 0.91$$

- rendimiento de imbricación de programas:

$$\frac{\text{nº de octetos de instrucciones de programas}}{\text{capacidad de memoria ocupada}} = \frac{1837}{1852} = 0.99$$

3.7 CONSIDERACIONES LOGICAS SOBRE LLAMADAS Y ANIDAMIENTO ENTRE SUBRUTINAS:

Según se ha dicho en la sección anterior el número de rutinas que constituyen el S.O. es de 138. En principio, para ver las conexiones y llamadas entre ellas, parecería lógico representarlas por medio de un grafo orientado; ahora bien, en el presente caso esto daría lugar a una labor tediosa y el grafo resultante no sería de utilidad por su gran complejidad. En esta sección se pretende esbozar un método sistemático para el estudio de conexiones entre subrutinas.

Sea S el conjunto de todos los programas y subrutinas que constituyen el sistema operativo.

En el conjunto S se define la relación R:

$$R = \{(Ai\ ,\ Aj) \mid \quad Ai \quad \text{"llama a"} \quad Aj\}$$

Esta relación no es total, ya que pueden existir elementos del conjunto que no estén relacionados.

DEFINICION 1: MATRIZ RELACION (MR):

Se denomina "matriz relación" a una matriz tal que uno de sus elementos (i,j) es "1" si AiRAj, y "0" en caso contrario.

La matriz de relación es cuadrada.

DEFINICION 2: MATRIZ DE RELACION REDUCIDA (MRR):

Es la matriz que se obtiene a partir de la matriz de relación eliminando las filas y columnas tales que todos sus elementos sean cero.

DEFINICION 3: CIERRE TRANSITIVO (T) DE LA RELACION R:

Es la mínima relación transitiva que contiene a la relación dada.

Se demuestra que:

$(Ai,Aj) \in T$, si y sólo si:

$\exists$ k1, k2, k3,..., kp $\mid$ $(Ai,Ak1) \in R$, $(Ak1,Ak2) \in R$,... $(Akp,Aj) \in R$.

Observese que de la expresión anterior se deduce que:

(Ai,Ak2), (Ai,Ak3), (Ai,Ak4),....,(Ak1,Ak3),....

pertenecen también al cierre transitivo.

El cierre transitivo contiene a todas las parejas de las rutinas que están relacionadas entre sí directa o indirectamente (a través de otras).

De forma análoga a la definición 1 se puede definir la "matriz cierre transitivo de la relación (MT).

Un algoritmo para calcular la matriz cierre transitivo es:

$$MT = MR + MR^2 + ... + MR^n$$

donde n es tal que $MR^{n+1} = 0$; las operaciones están referidas a la lógica booleana (+: unión, . : intersección), y las operaciones con matrices se realizan de acuerdo con el álgebra de matrices. Siempre n es menor o igual que el número de filas de MR.

La matriz cierre transitivo tiene un gran interés práctico (detección de posibles errores,...), entre otros por los siguientes motivos:

(1) n da el nivel máximo de anidamiento (en el presente caso n ha de ser menor o igual a 7).
(2) caso de que un elemento de la diagonal principal sea 1 implica un lazo entre rutinas; y por tanto hay que estudiar la posibilidad de recursividad.
(3) con la matriz MT se puede localizar con gran facilidad todas las rutinas a las que puede afectar un error en una de ellas.

...

Con objeto de obtener fácilmente con ordenador el cierre transitivo de una matriz, existe el algoritmo de Warshall (51). Este calcula:

$$A = MR + MR^2 + \ldots + MR^n$$

siendo MR (nxn), de acuerdo con el siguiente proceso iterativo:

1: hacer a A <--- MR

2: hacer i <--- 1

3: para cada j, si A(i,j) = 1, entonces
para k = 1,...,n hacer (i,j) <---A(j,k)+A(i,k)

4: hacer i<--- i + 1

5: si i menor o igual que n ir a 3, en otro caso fin.

3.8 SINOPSIS:

En este capítulo se han desarrollado todos los aspectos relacionados con la construcción del Sistema Operativo. o software del ordenador.

En una primera etapa se han descrito y delimitado la aritmética utilizada y los algoritmos numéricos para operar con funciones.

Posteriormente se han formulado los organigramas y programas del sistema, llevandose a la práctica las ideas expuestas en las secciones anteriores de la memoria.

Finalmente se han expuesto los criterios que se han seguido en la asignación de memoria así como en la imbricación de los programas, y se han dado unos datos numéricos sobre la implementación del sistema en la memoria REPROM.

CAPITULO IV

DISEÑO ELECTRONICO DEL DISPOSITIVO:

4.1 INTRODUCCION:

En este capítulo se presenta el diseño del sistema desde el punto de vista físico.

En primer lugar se indica la organización y diseño de la memoria. Posteriormente se estudia la estructura y diseño de los circuitos básicos del ordenador, deteniéndose especialmente en la generación de las señales de control.

Finalmente se describe la realización y montaje del sistema.

4.2 DISEÑO DE LA MEMORIA:

Los componentes utilizados para la memoria (secc. 1.4.3) son circuitos MOS estáticos:

- La memoria REPROM está constituida por 8 pastillas organizadas en palabras de 8 bits. Cada pastilla contiene 256 octetos. Los terminales lógicos de cada pastilla son:
  - 8 de entrada para direccionamiento del octeto a leer (AROM).
  - 1 de selección de pastilla ("chip select") (CSO) y
  - 8 de salida del octeto leido (DO)

- La memoria RAM estática, a su vez, está constituida por 24 pastillas organizadas en 1024 x 1 bit cada una. Los terminales de una pastilla son:
  - 1 terminal de entrada del bit a memorizar (EM)
  - 1 terminal de salida del bit de información leido (DA)
  - 1 terminal de selección de pastilla ("chip enable") (CEA).
  - 10 terminales para direccionamiento del bit a leer o escribir (ARAM)
  - 1 terminal de control de lectura/escritura (R/W).

Para formar con las pastillas de memoria RAM un octeto, es necesario conectar 8 de ellas tal como indica la figura 4.1. De esta forma se tiene, para los datos de salida, un total de 11 buses de 8 bits cada uno:

LO0,....LO7, LA0, LA1 y LA2

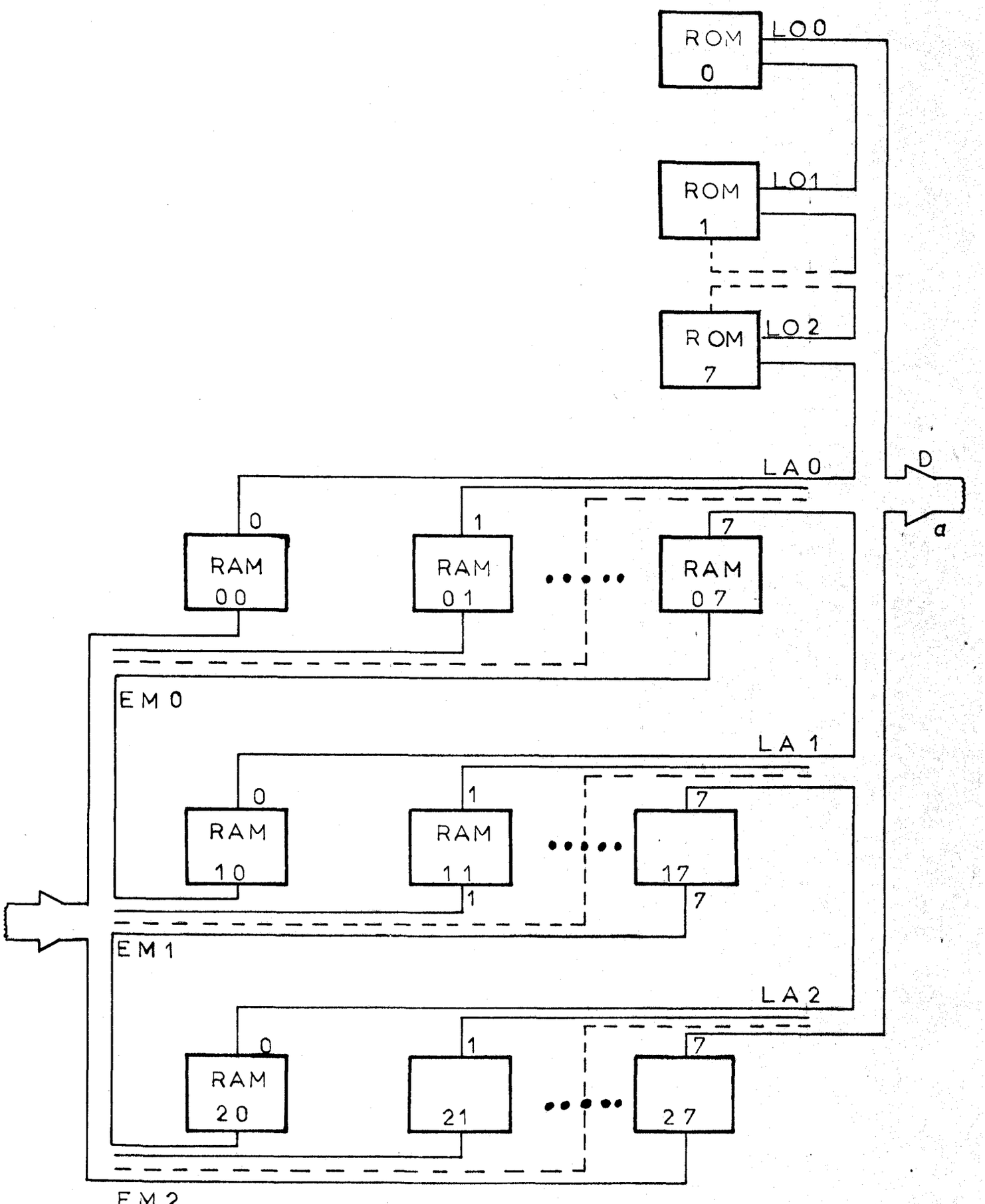


FIG 4-1

ORGANIZACION DE LA MEMORIA: E/S DE DATOS

La lógica de salida de las memorias es de "tres estados" (three state). Cuando la señal de selección de pastilla (CSO o CEA) es alta, la salida o salidas correspondientes están en el estado de alta impedancia; esto quiere decir que los 11 buses antes citados se pueden conectar entre si directamente, figura 4.1 En el punto de conexión a se realiza la operación lógica OR.

Como se vio en la secc. 1.4.1.1, el microprocesador facilita la dirección de memoria en los estados T1 y T2; esta se memoriza en los registros CL y CH, respectivamente. El direccionamiento de memoria se hace (figura 4.2):

- Memoria ROM:
  - ARON: CL7 a CLO (8 bits)
  - CSO: se obtiene a partir de una decodificación de los bits CH5 o CHO.
- Memoria RAM:
  - ARAM: CH1, CHO, CL7,...., CLO (10 bits)
  - CEA: se obtiene a partir de una decodificación de los bits CH5 y CH2.

En la figura 4.2 puede verse la organización de direccionamiento de la memoria, y en las figuras 4.3. y 4.4 los esquemas de montaje.

En la secc. 4.4.2 se expone el diseño de los circuitos de selección de memoria.

Sobre características técnicas y funcionamiento de los circuitos MOS estáticos utilizados, así como de tecnología en general sobre memorias integradas pueden consultarse las referencias (91), (64) y (67).

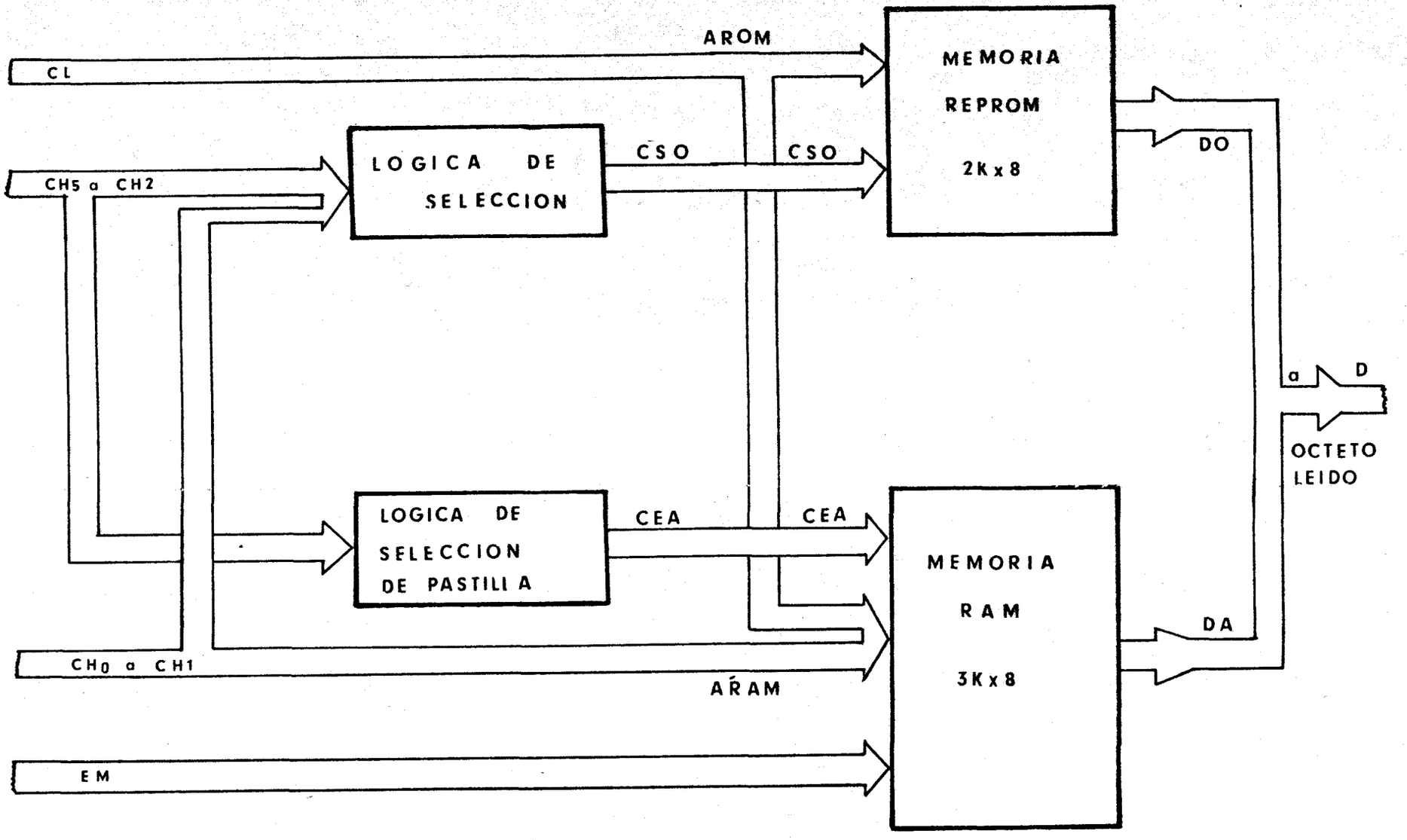


FIG. 4-2

ORGANIZACION DE LA MEMORIA

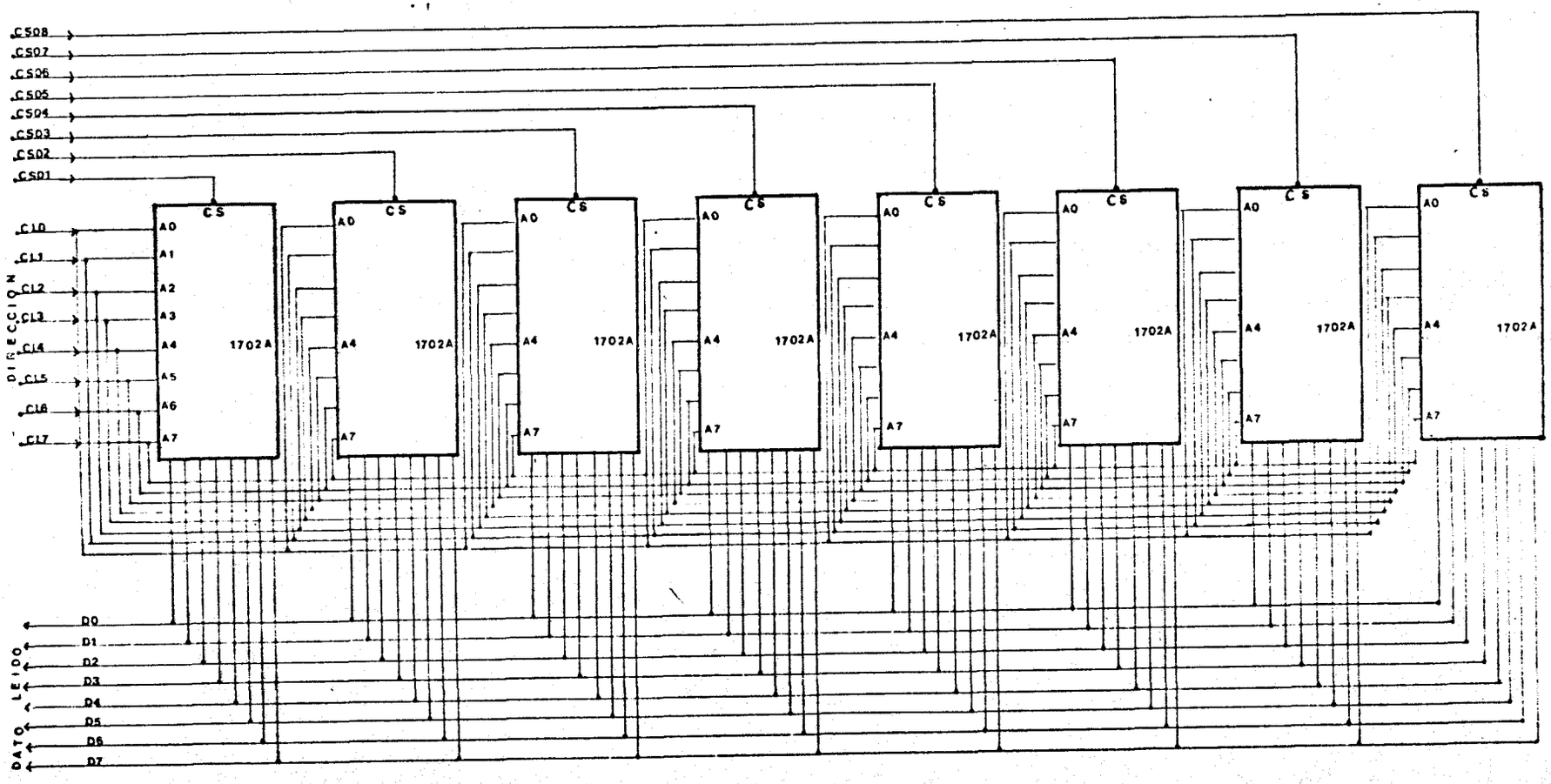


FIG. 4·3

ESQUEMA DE MONTAJE DE LA MEMORIA REPROM

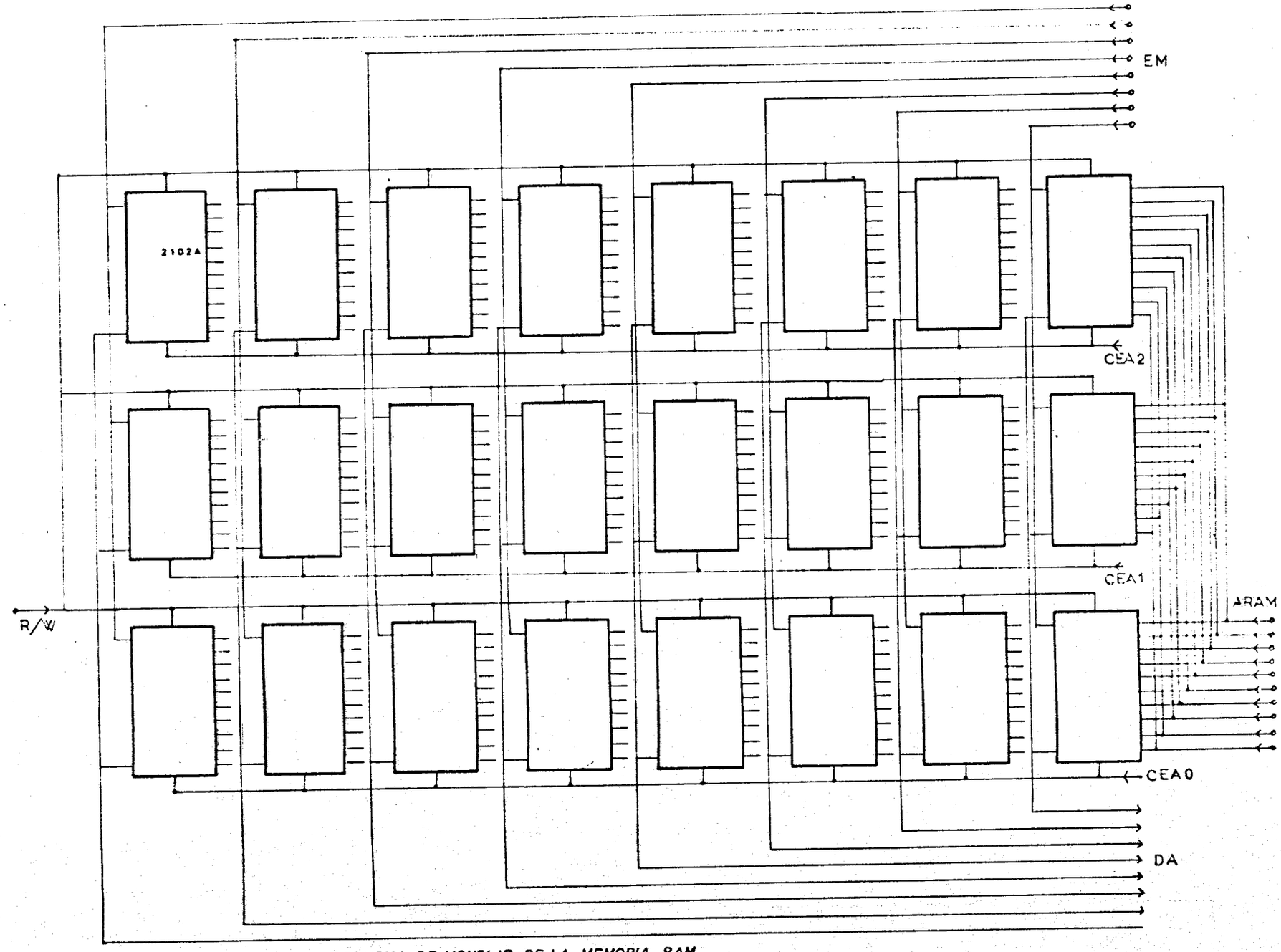


FIG. 4-4 ESQUEMA DE MONTAJE DE LA MEMORIA RAM

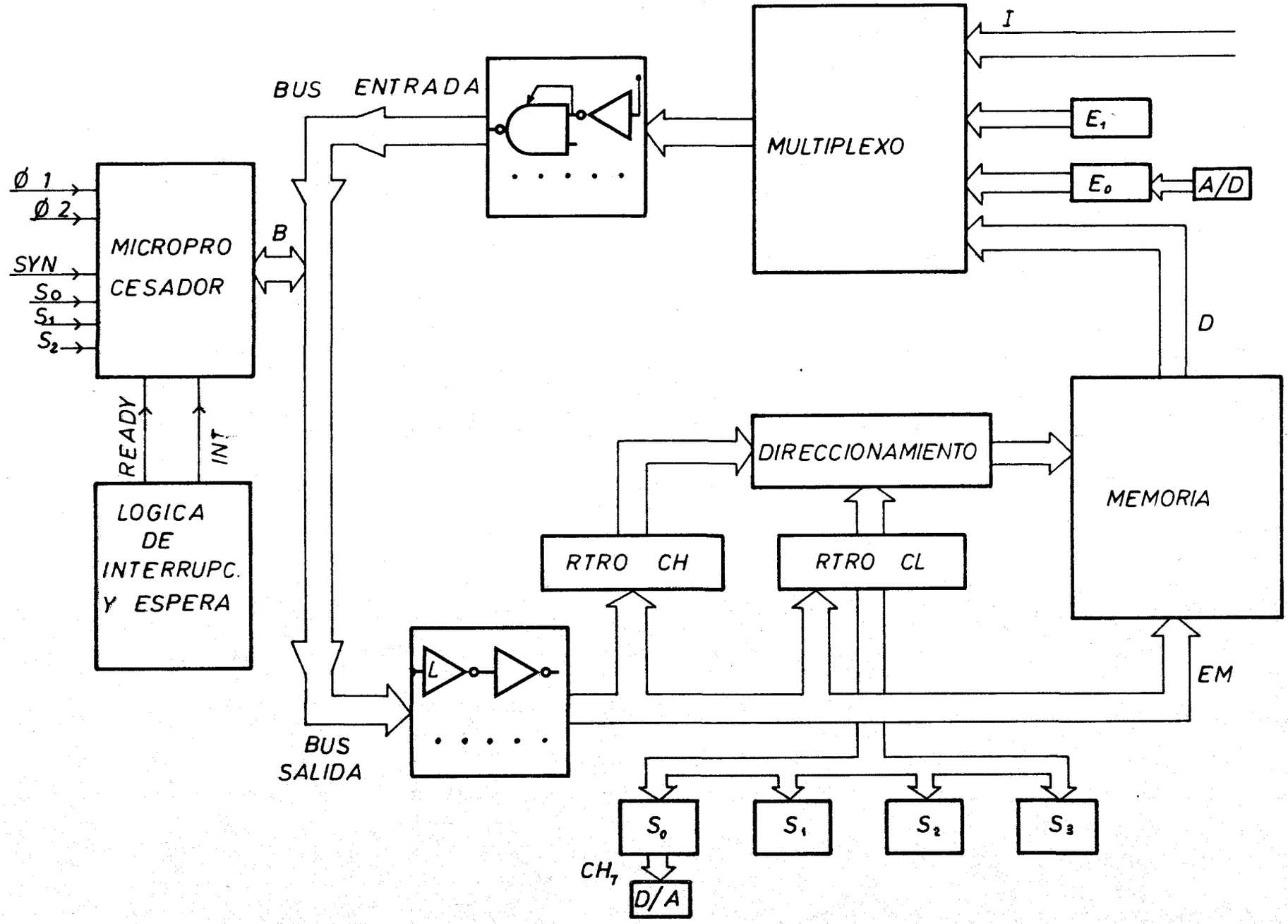


FIG. 4-5

ESTRUCTURA BASICA DEL MICROORDENADOR

4.3 ESTRUCTURACION Y DISEÑO GENERAL DEL MICROORDENADOR:

En la secc. 1.4.1 se presentó y justificó la estructura básica del microordenador. En la figura 4.5 se muestra un diagrama general detallado de la constitución del microordenador. Las partes esenciales del dispositivo son, además de la memoria, el bus de entrada y el bus de salida con sus circuitos asociados. En esta sección se describe el diseño de cada una de estas partes.

4.3.1 BUS DE ENTRADA:

Según se dijo en la secc. 1.4.1.2, el microprocesador se comunica con el exterior a través de un bus bidireccional. Este bus se bifurca en dos, uno de entrada y otro de salida, tal y como se indica en la figura 4.5. Para acoplar estos buses, el de entrada (figura 4.6) se conecta al microprocesador con puertas Y (tipo "buffer") de tres estados (2 pastillas SN74125). Cuando la señal de control de estos circuitos, a, es alta, la salida está en el estado de alta impedancia; es decir, siempre que sea necesario entrar al microprocesador debe hacerse a=0 y, en caso contrario, a=1.

La información de entrada al microprocesador puede provenir de la memoria, del dispositivo de entrada E1, de los interruptores del panel de control o del conversor A/D, E0. Para la conexión de estos dispositivos al bus del microprocesador se utiliza un multiplexo de 4 canales a 1. Esta multiplexación se realiza con cuatro circuitos integrados SN74153 (multiplexos 4:1), según puede verse en la figura 4.6 Cuando la señal a ("strobe") es baja hay transmisión de información, y cuando es alta todas las salidas son 0. Los terminales b y c se utilizan para seleccionar cuál de los canales de entrada debe aparecer en la salida. Este direccionamiento se hace de acuerdo con la siguiente tabla (89p. 1-75):

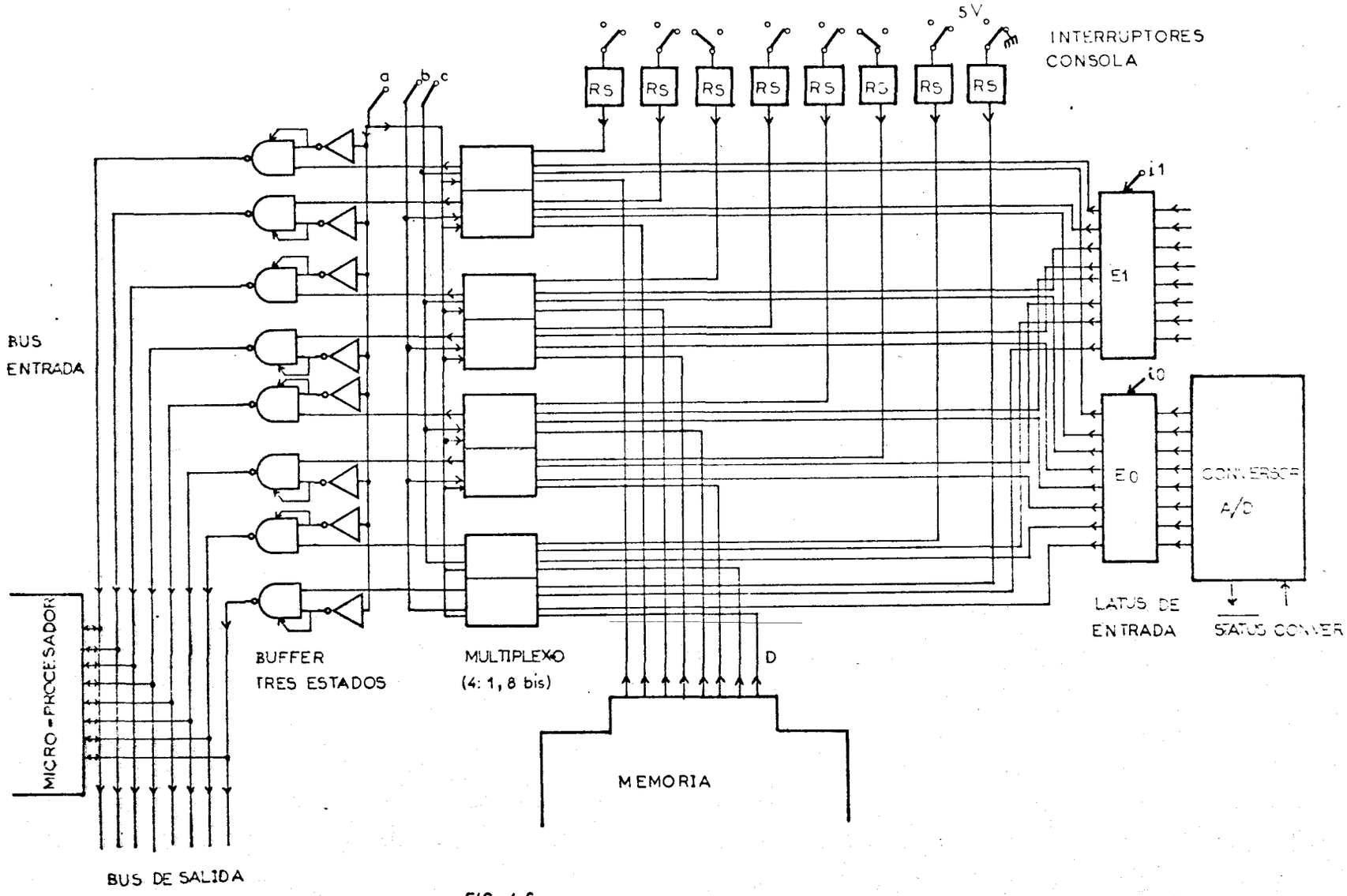


FIG. 4-6

CONEXIONES REALIZADAS EN EL BUS DE ENTRADA

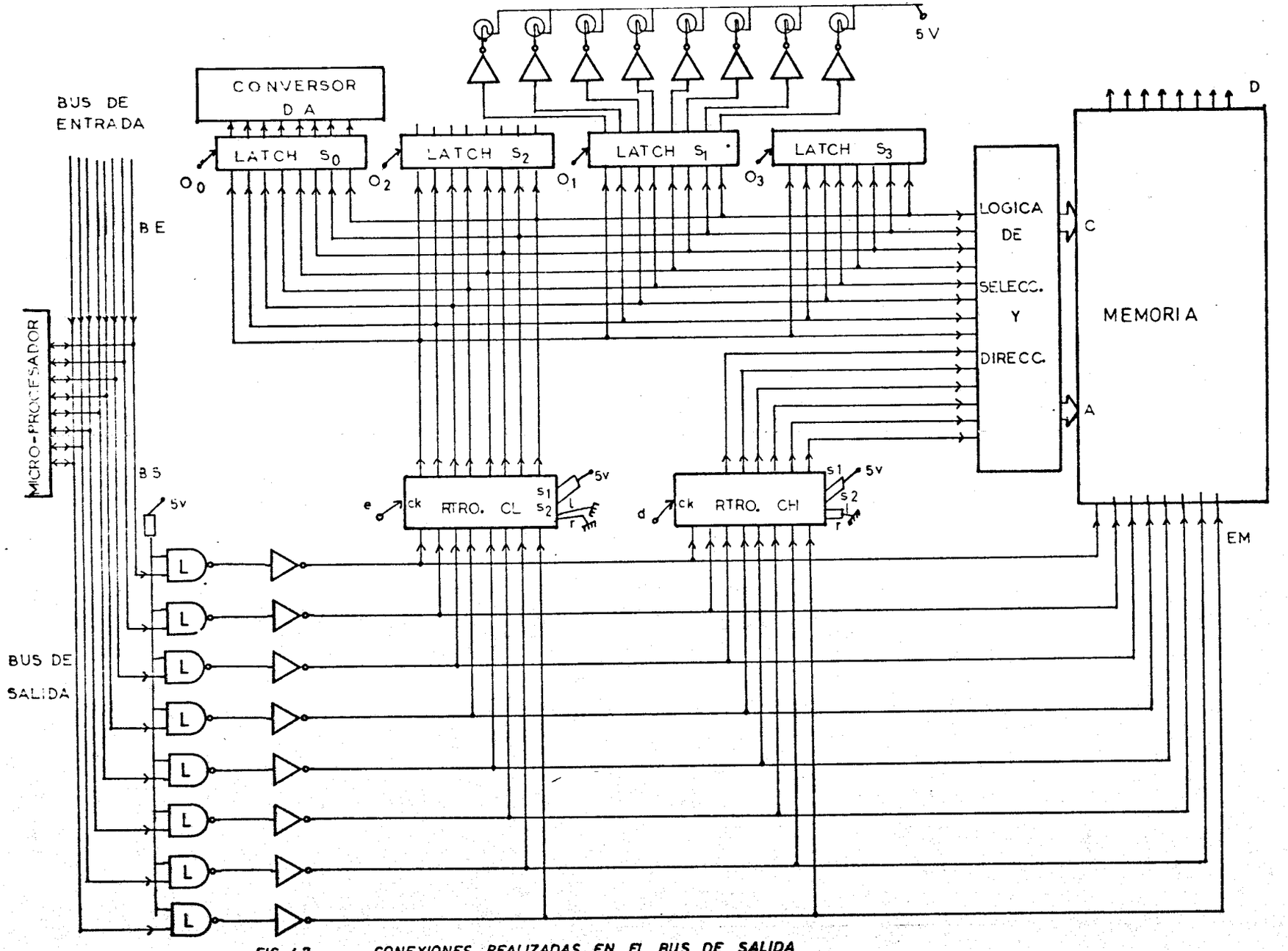


FIG 4-7 **CONEXIONES REALIZADAS EN EL BUS DE SALIDA**

c	b	entrada que se transmite a la salida
0	0	memoria
0	1	interruptores del panel
1	0	entrada E0 (conversor A/D)
1	1	entrada E1

Tabla 4.1
direccionamiento del multiplexo.

4.3.2 BUS DE SALIDA:

En la figura 4.7 puede verse un esquema de los circuitos que van conectados al bus de salida.

Debido al pequeño valor del "fan-out" del microprocesador, a la entrada del bus se han colocado 8 inversores con lo que este aumenta a 10 circuitos TTL. Como la salida del microprocesador es de baja potencia (LOW POWER) antes de los inversores se conectan 8 circuitos NAND de baja potencia (SN74L00).

Los registros utilizados para memorizar la información que facilita el microprocesador en los estados T1 y T2 (registros CL y CH, respectivamente) son registros de desplazamiento de 8 bits cada uno: SN74198.

La carga de los registros se hace después de la transición positiva del pulso de reloj; estas señales de control de carga se denominan en este trabajo e y d.

Las salidas CL0 a CL7 y CH5 a CH0 son entradas de la lógica de selección y direccionamiento de memoria (secc. 4.2). El registro CL también está conectado a los dispositivos de salida (S0, S1, S2 y S3).

El bus de salida termina en la entrada de la memoria RAM (EM, figura 4.7).

4.3.3 DISPOSITIVOS DE ENTRADA Y SALIDA:

Los dispositivos de entrada y salida están constituidos por básculas D (tipo "latch"). Cada salida está formada por dos circuitos integrados SN7475 (89 p1-46); es decir, por 8 básculas.

Las señales de carga o reloj de las básculas se denominan (figura 1.6 y 1.7):

básculas S0: señal O0
básculas S1: señal O1
básculas S2: señal O2
básculas S3: señal O3
básculas E0: señal i0
básculas E1: señal i1

Cuando la entrada de control de carga esta en el estado lógico "1", las salidas de las básculas siguen a las entradas, quedando memorizadas estas últimas en la transición de "1" a "0".

En las figuras 4.6 y 4.7 puede verse como están conectadas las básculas al resto del sistema.

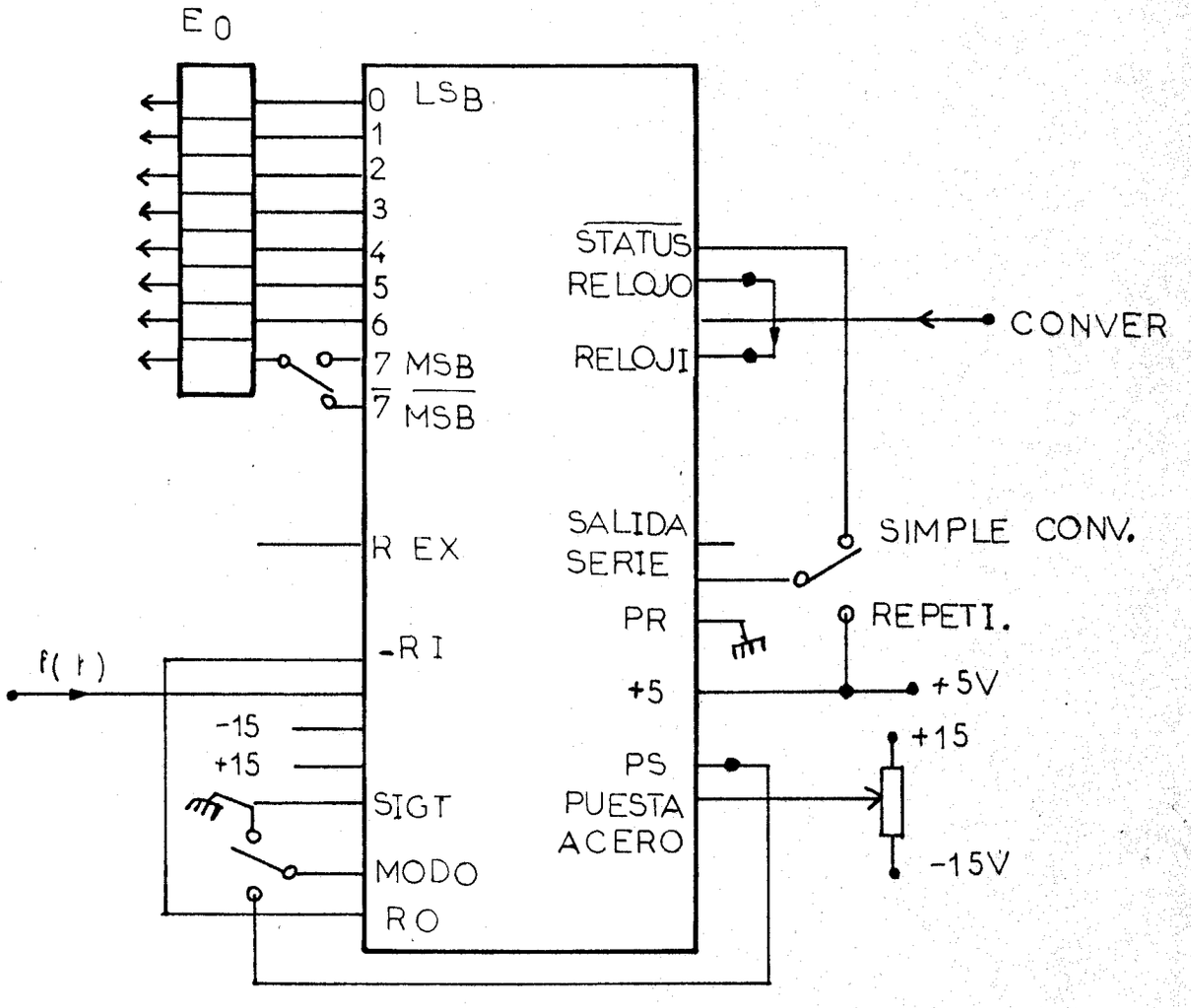


FIG. 4-9

CONEXIONES QUE SE REALIZAN EN EL CONVERSOR A/D

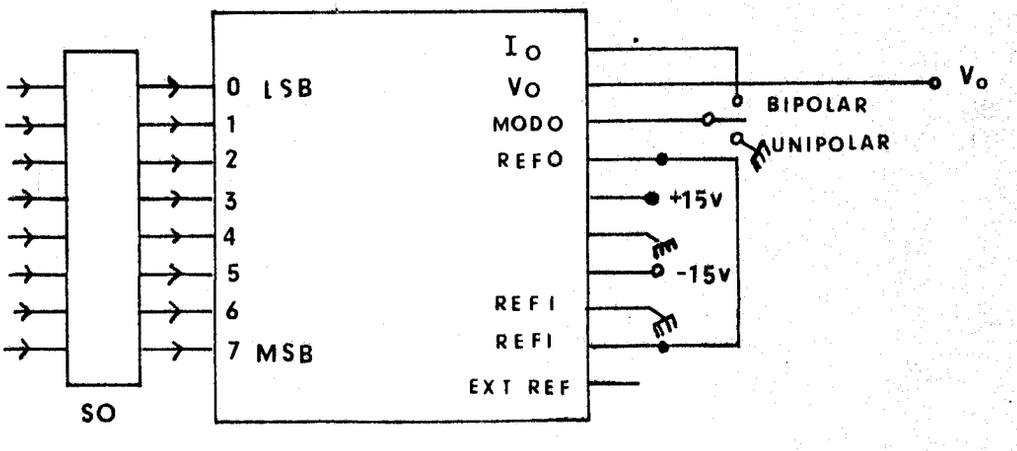


FIG. 4-8

CONEXIONES QUE SE REALIZAN EN EL CONVERSOR D/A

4.3.4 CONVERSORES:

Los conversores A/D y D/A utilizados, Teledyne Philbrick modelos 4114 y 4021, respectivamente, son bloques compactos construidos utilizando técnicas de microcircuitos.

4.3.4.1 CONVERSOR DIGITAL / ANALOGICO DE SALIDA:

El conversor puede trabajar en forma unipolar o bipolar:

- unipolar: el intervalo de amplitudes de la señal de entrada debe estar comprendido entre -10 y 0 voltios.
- bipolar: el intervalo de amplitudes de entrada está comprendido entre -5 y +5V.

En la figura 4.8 se muestran las conexiones que se realizan en el conversor para acoplarlo al resto del dispositivo la entrada digital está constituida por las salidas de las básculas SO. En la referencia (113 p38) pueden verse otras características técnicas del conversor utilizado.

4.3.4.2 CONVERSOR ANALOGICO / DIGITAL DE ENTRADA:

El conversor A/D es del tipo de "aproximaciones sucesivas" (113), (108), (56).

En la figura 4.9 se indican las conexiones que se realizan en el conversor.

Por medio de interruptores puede hacerse trabajar al conversor de los siguientes modos:

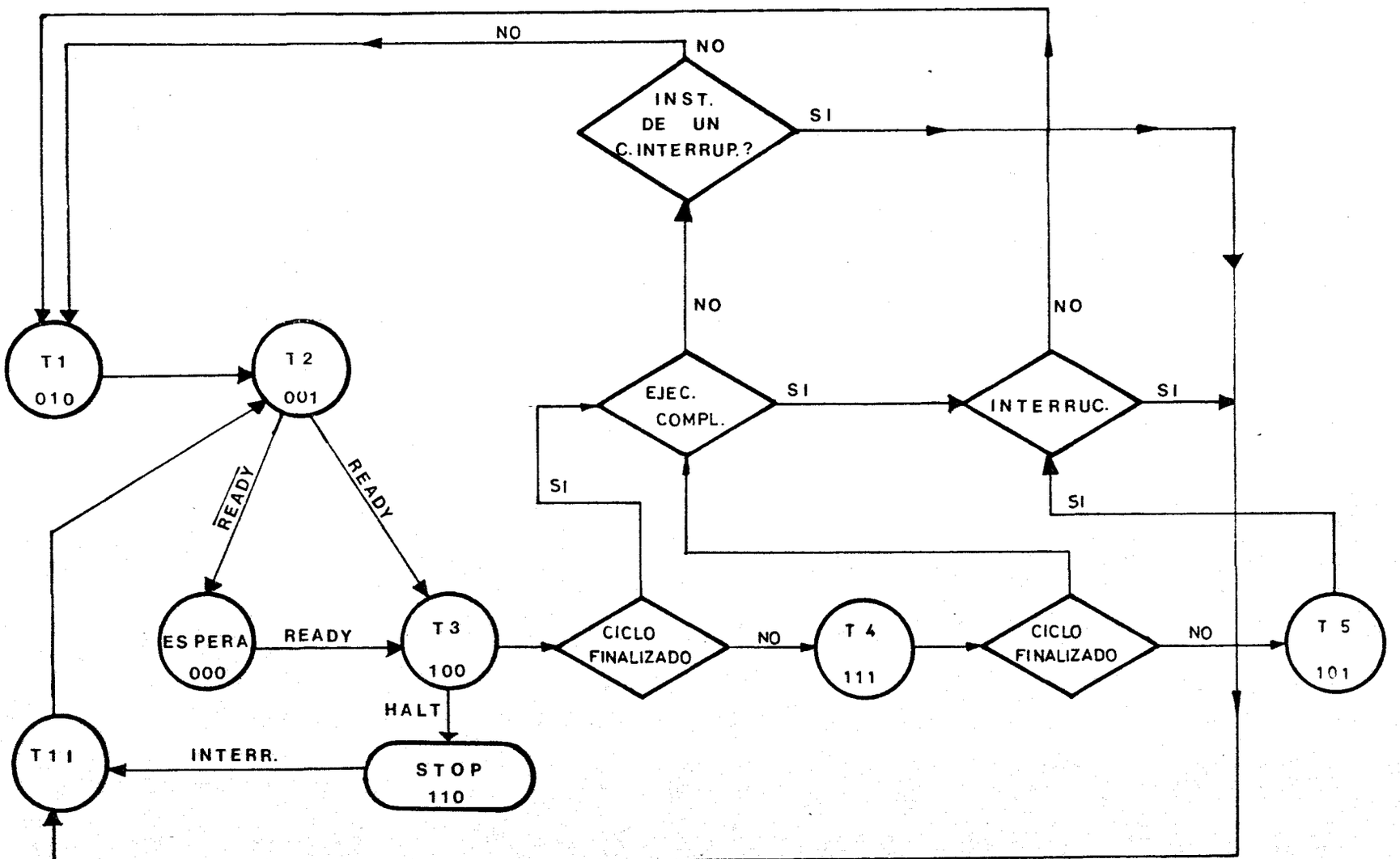


FIG. 4-10

DIAGRAMA DE TRANSICION DE ESTADOS DEL MICROORDENADOR

(1) En modo repetitivo o en simple conversión. En el primer modo las conversiones se realizan iterativamente sin necesidad de dar externamente la señal de control CONVER.

(2) En modo unipolar o bipolar; la señal de entrada debe estar comprendida entre los limites de 0 a -10 V o de -5 a +5V, respectivamente.

(3) Con código de salida en binario o binario complemento a 2.

Otras características técnicas del conversor pueden verse en la referencia (113 p42).

4.4 GENERACION DE LAS SEÑALES DE CONTROL:

4.4.1 CARACTERISTICAS ELECTRICAS Y FUNCIONAMIENTO DINAMICO DEL MICROPROCESADOR:

En esta sección se presenta el funcionamiento dinámico (transición de estados y diagrama de tiempo) y principales características eléctricas del microprocesador.

En la figura 4.10 puede verse un diagrama de transición de estados. Un ciclo normalmente está constituido por los estados T1, T2, T3, T4 y T5. Los estados T4 y T5 no tienen por que darse necesariamente. En los ciclos de interrupción el microprocesador pasa por el estado T1I en lugar de T1.

Cuando la señal READY se hace "0" el microprocesador entra, después de pasar por T1 o T1I y T2, en el estado de ESPERA; cuando sale de él continua el ciclo normalmente (T3,...). Por medio

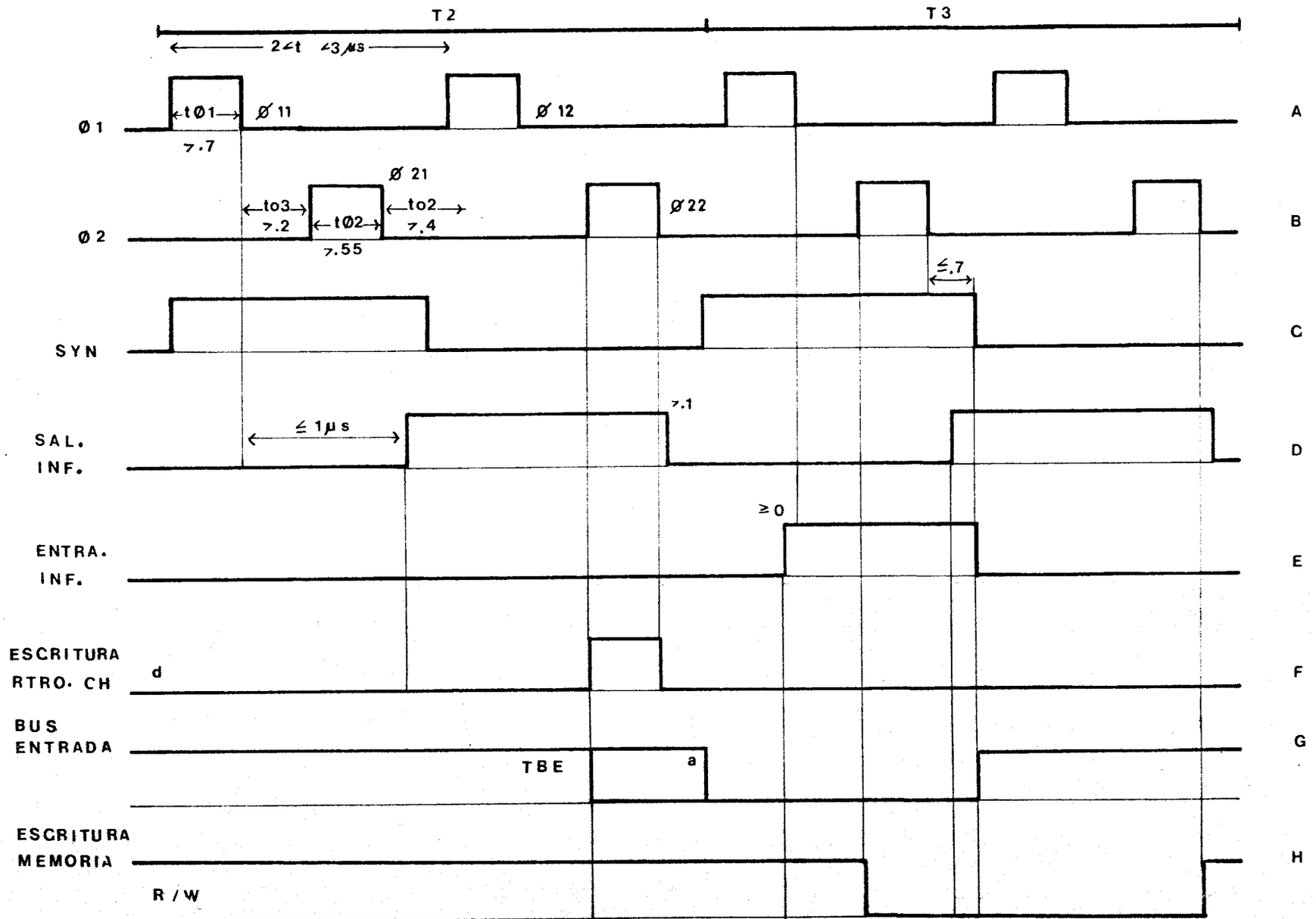


FIG. 4-11

DIAGRAMA DE TIEMPOS DEL MICROORDENADOR

de la instrucción HALT el microprocesador, después de pasar por T3, se situa en el estado STOP. Para salir de STOP es necesario provocar una interrupción (INTERRUPT = "1").

En la figura 4.11 se representa un diagrama de tiempos. Las señales de reloj (líneas A y B de la figura) deben cumplir las siguientes especificaciones:

		Mín.	Máxi.	
tØ1	duración del impulso Ø1	0,7		microseg.
tØ2	duración del impulso Ø2	0,55		microseg.
t12	tiempo transcurrido entre el final de Ø1 y comienzo de Ø2	0,2		microseg.
t21	tiempo transcurrido entre el final de Ø2 y comienzo de Ø1	0,4		microseg.
t	periodo de reloj	2,0	3,0	microseg.
	frecuencia de reloj	0,33	0,5	megaciclos

Un estado comprende siempre cuatro pulsos de reloj, que se denominan Ø11, Ø21, Ø12, Ø22 y que vienen dados por las ecuaciones lógicas (fig. 4.11 A,B y C):

$$Ø11 = SYN \cdot Ø1$$
$$Ø21 = SYN \cdot Ø2$$
$$Ø12 = \overline{SYN} \cdot Ø1$$
$$Ø22 = \overline{SYN} \cdot Ø2$$

El microprocesador da la señal de sincronismo (SYN, figura 4.11C) coincidiendo con la bajada del pulso Ø2, pero puede aparecer con un retraso de hasta 0,7 microsegundos como máximo.

La información de salida del bus del microprocesador (figura 4.11C) está sincronizada con el final del impulso Ø11, pudiendo tener un retraso de hasta un microsegundo. Esta información, permanece en el bus de salida por lo menos hasta 0.1 microsegundos después del final del pulso Ø22.

Cuando el microprocesador capta información a través del bus de entrada, lo hace, tal como se indica en la figura 4.11E, en el intervalo de tiempo comprendido entre el final de Ø11 hasta la bajada del pulso SYN.

Otra consideración importante es que las señales indicativas del estado (s0, s1 y s2) pueden ser dadas por el microprocesador con un retraso de hasta 1 microsegundo respecto al final del último impulso Ø22 del estado anterior.

Para entrar en el estado ESPERA la señal READY debe hacerse '0 'antes de (T2.Ø22) y volver a '1'con un retraso mayor de 0,20 microsegundos respecto a la bajada de (T2.Ø22). Para salir de este estado en el terminal READY hay que dar un impulso antes del final de Ø22, de anchura mayor de 0,35 microsegundos. La sincronización de la señal de interrupción tiene que hacerse de tal forma que el cambio de nivel a "1" de INTERRUPT no caiga dentro de los 200 ns anteriores al final del pulso Ø1.

En cuanto a las características eléctricas se tiene para las señales de entrada y salida:

tensión de entrada nivel "0" (VIL) mín.:-9V máx.:0,8V
tensión de entrada nivel "1" (VIH) mín.: 3,5V máx.: 5.3V
tensión de salida nivel "0" (VOL) máx.: 0,4V
corriente de salida nivel "0" (IOL) típ.: 0,44 mA
tensión de salida nivel "1" (VOH) mín.: 3,5 V
corriente de salida nivel "1" (IOH) típ.: 0,2 mA.

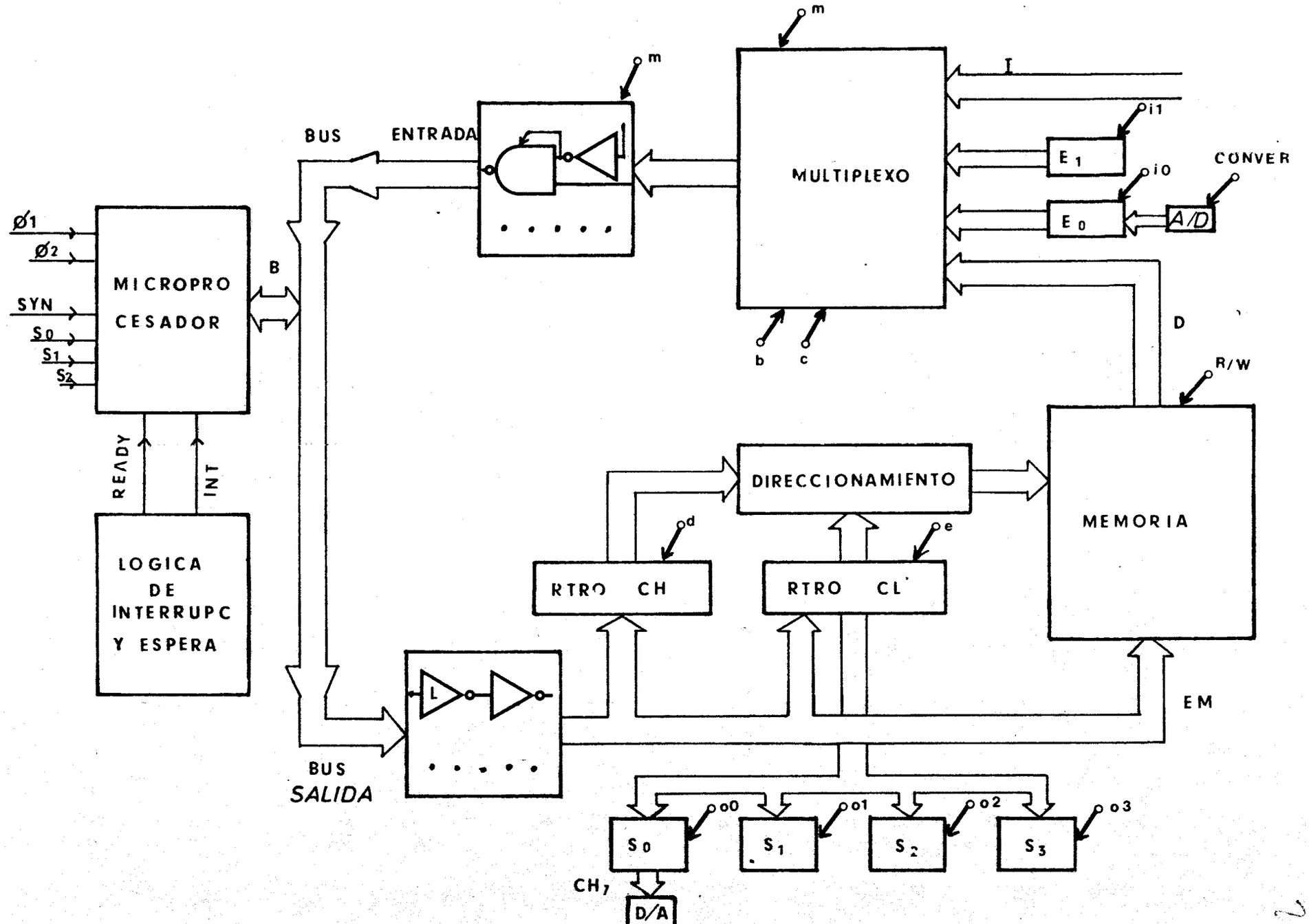


FIG. 4-12

ESTRUCTURA BASICA DEL MICROORDENADOR

El microprocesador, por lo tanto, es compatible con circuitos TTL de baja potencia. Para asegurar los niveles altos de entrada es conveniente utilizar resistencias "pull-up".

4.4.2 SEÑALES DE CONTROL BASICAS:

En esta sección se realiza el diseño de los circuitos que generan las señales de control básicas. En las secciones 4.4.3 y 4.4.4 se estudian las señales de control para la interfase con los conversores y la lógica de interrupción y espera, respectivamente.

En la figura 4.12 se muestra un esquema con las señales de control necesarias para el funcionamiento del dispositivo. Las funciones de control básicas se generan a partir de señales que da el microprocesador, estas son:

(1) señal de sincronismo, (SYN)
(2) señales identificativas de estados (s0, s1 y s2)
(3) señales identificativas de ciclos (CH7, CH6)
(4) señales de control de entradas/salidas (CH5, CH4, CH3, CH2 y CH1), y
(5) direccionamiento de memoria (CH5 a CHO y CL7 a CLO).

La información proporcionada por los terminales s0, s1 y s2 del microprocesador (estado) es tratada por un decodificador 3x8, tal como se indica en la figura 4.13 y en la tabla 4.2. De esta forma se tienen directamente las señales indicativas de estado (T1, T2, T3, T4, T5, T1I, W, STOP), las cuales intervienen en la mayor parte de las funciones de control a generar.

Las señales de control a generar son:

- señal de identificación del ciclo de interrupción.

CIN

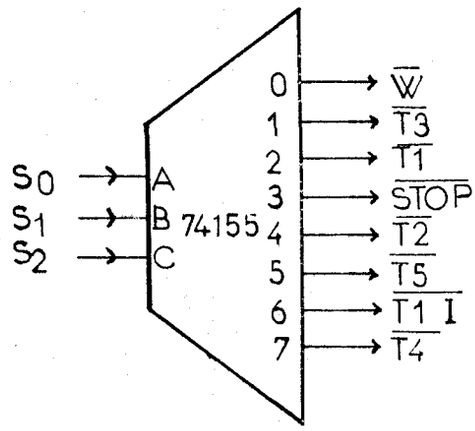


FIG. 4-13

DECODIFICADOR DE ESTADOS

C B A	7 6 5 4 3 2 1 0
o o o	| | | | | | | o
o o |	| | | | | | o |
o | o	| | | | | o | |
o | |	| | | | o | | |
| o o	| | | o | | | |
| o |	| | o | | | | |
| | o	| o | | | | | |
| | |	o | | | | | | |

TABLA 4-2

DECODIFICACION DE ESTADOS

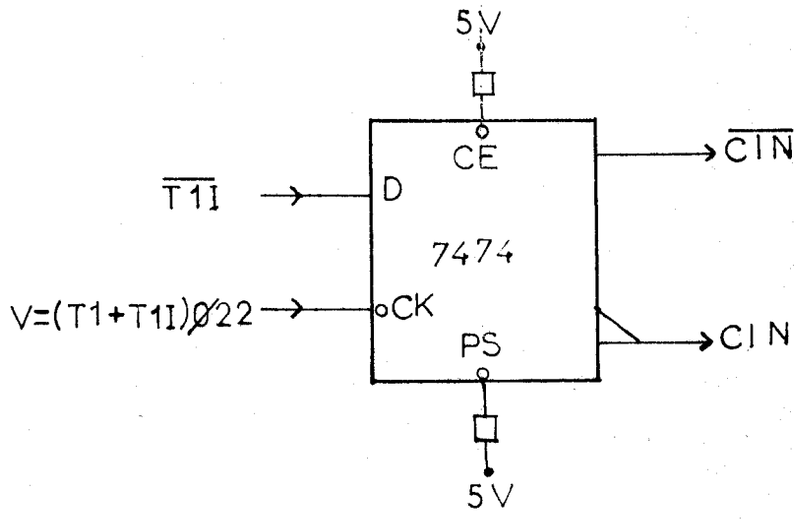


FIG. 4-14

SEÑAL DE CICLOS DE INTERRUPCION

- señales de carga de registros:
  - .RTRO CL: señal e
  - .RTRO CH: señal d

- selección de pastilla de memoria:
  - . memoria ROM:
    CSO0, CSO1, CSO2, CSO3, CSO4, CSO5, CSO6, CSO7
  - . memoria RAM:
    CEA0, CEA1, CEA2,....., CEA15.

- lectura/escritura de memoria:
  - R/W

- señales de carga de RTROS de salida:
  - . registro O0: señal o0
  - . registro O1: señal o1
  - . registro O2: señal o2
  - . registro O3: señal o3.

- señales de identificación de entrada:
  - . registro E0: señal e0
  - . registro E1: señal e1.

- señales de control del bus de ENTRADA:
  - . entrada de información al microprocesador a
  - . multiplexación b,c.

4.4.2.1 CICLOS DE INTERRUPCION:

El microprocesador indica que reconoce una interrupción si pasa por el estado T1I en lugar de T1 (figura 4.10). Es necesario memorizar durante los próximos estados el hecho de que el

microprocesador se encuentra en un "ciclo de interrupción" (CIN=1). Esta memorización se efectua por medio de una báscula D tal y como se indica en la figura 4.14. Al pasar por un estado T1I, la salida $\overline{Q}$ se hace '1', hasta el próximo ciclo en el que se de un estado T1.

Las señales de identificación de estado (T1 o T1I) pueden llegar a tener un gran retraso respecto a las señales de reloj y sincronismo (secc. 4.4.1), por lo que la señal de reloj de la báscula se da al finalizar el estado (Ø22):

$$v = (T1 + T1I) \cdot Ø22.$$

4.4.2.2 REGISTROS:

Según se indicó en la secc. 4.3.2, en los registros CL y CH se memoriza la información dada por el bus del microprocesador en los estados T1 y T2. Las ordenes de carga, e y d, de los registros (figura 4.11F) hay que darlas dentro del intervalo de tiempo en que el microprocesador da la información de dalida (figura 4.11D).

memorización en CL: $e = (T1.+T1I).Ø22$

memorización en CH: $d = T2.Ø22$.

4.4.2.3 MEMORIA:

Para implementar la lógica de selección de memoria expuesta en la sección 4.2, se utiliza un decodificador 4x8 (figura 4.15).

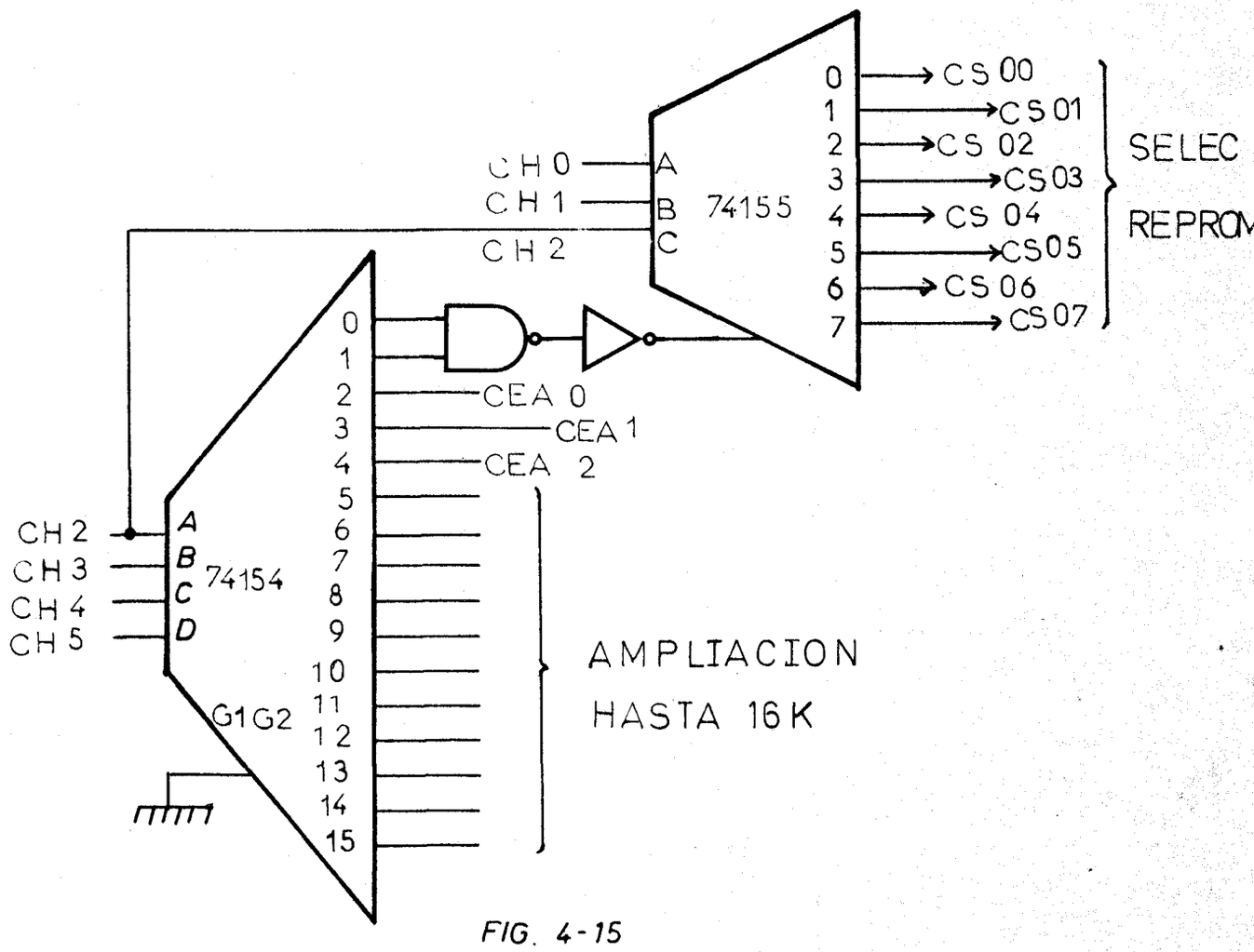


FIG. 4-15

LOGICA DE SELECCION DE MEMORIA

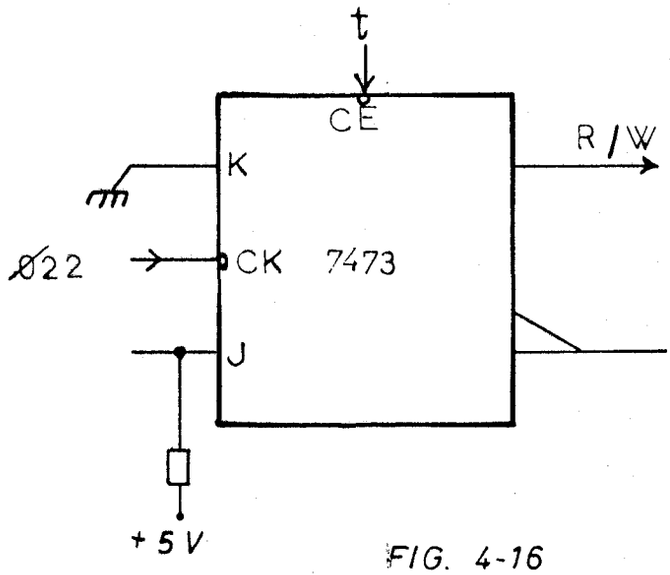


FIG. 4-16

GENERACION DE LA SEÑAL R/W

A cada combinación de entrada: CH2, CH3, CH4, CH5, le corresponde una selección de un bloque de 1k de memoria. En el caso de que la selección corresponda a los dos primeros k de memoria, por lo tanto memoria ROM, se realiza una nueva decodificación (3x8) con los bits CH2, CH3 y CHO para la selección de una de las 8 pastillas de estos dos primeros K.

En la figura 4.15 puede verse un esquema de la realización práctica de la lógica de la selección de memoria y en la tabla 4.3 la tabla de verdad de las funciones realizadas.

Obsérvese que las salidas de los decodificadores son altas ('1') mientras no son seleccionadas, en caso contrario descienden al nivel bajo ('0'); éste es precisamente el requisito de las especificaciones de las señales de selección de memoria los componentes elegidos: Intel 1702A y 2102A (64).

La selección de memoria se realiza permanentemente; es decir, dentro de un ciclo se efectua a partir del momento en que quedan grabados en el registro CH los bits más significativos de direccionamiento (intervalo de tiempo (T2.Ø22)). Se deduce, por tanto, que la memoria utilizada puede tener un tiempo de acceso máximo de hasta 1.65 microsegundos, suponiendo que los relojes funcionen a una frecuencia de 2,5 microsegundos.

La señal de control de lectura/escritura de memoria, R/W, debe ser normalmente 1 y hacerse 0 al efectuar una escritura de información en memoria; es decir, dentro de los ciclos PCW (figura 4.11 H). Como se observa la orden de escritura se da con una cierta antelación respecto al instante en que el microprocesador da la información a memorizar (figura 4.11D).

							MEMORIA RAM			MEMORIA ROM							
CH5	CH4	CH3	CH2	CH1	CHO	....	CEA3	CEA2	CEA1	CSO8	CSO7	CSO6	CSO5	CSO4	CSO3	CSO2	CSO1
0	0	0	0	0	0		1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	1		1	1	1	1	1	1	1	1	1	0	1
0	0	0	0	1	0		1	1	1	1	1	1	1	1	0	1	1
0	0	0	0	1	1		1	1	1	1	1	1	1	0	1	1	1
0	0	0	1	0	0		1	1	1	1	1	1	0	1	1	1	1
0	0	0	1	0	1		1	1	1	1	1	0	1	1	1	1	1
0	0	0	1	1	0		1	1	1	1	0	1	1	1	1	1	1
0	0	0	1	1	1		1	1	1	0	1	1	1	1	1	1	1
0	0	1	0	–	–		1	1	0	1	1	1	1	1	1	1	1
0	0	1	1	–	–		1	0	1	1	1	1	1	1	1	1	1
0	0	1	0	–	–		0	1	1	1	1	1	1	1	1	1	1
0	0	1	1	–	–		1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	–	–		1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	–	–		1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	–	–		1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	–	–		1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	–	–		1	1	1	1	1	1	1	1	1	1	1
1	0	0	1	–	–		1	1	1	1	1	1	1	1	1	1	1

2 PRIMEROS k

MEMORIA ROM

DIRECCION:

CL7 a CLO (8 bit)

14 ULTIMOS k

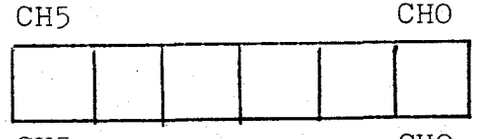
MEMORIA RAM

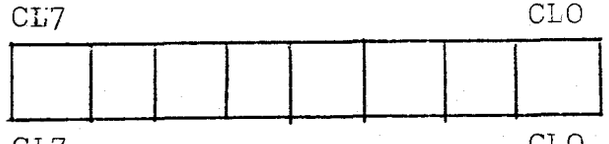
DIRECCION:

CH1 CHO

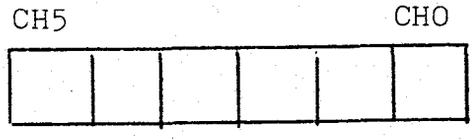
CL7 ------ CLO(8bit)

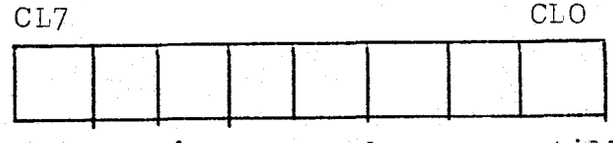
DIRECCION:14 bit (16k de memoria)





ROM (8 pastillas de 256 palabras)

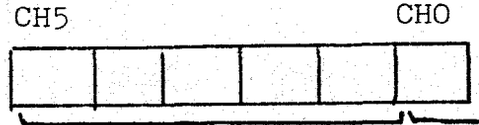


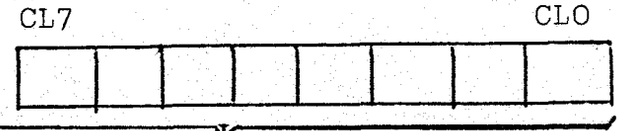


selección de pastilla

dirección dentro de una pastilla

RAM (8 bloques de 1 k palabras)





selección de bloque

dirección dentro de bloque

La función R/W se genera con una báscula JK, y como se indica en la figura 4.16. La señal de puesta a cero es:

$$t = \overline{PCW.\ T3.\emptyset 21} = \overline{CH6.CH7.T3.\emptyset 21}$$

El biestable vuelve al estado '1' en el próximo impulso de reloj; es decir, de acuerdo con (figuras 4.11B y H):

$$u = \emptyset 22.$$

De la figura 4.11D se deduce que se pueden utilizar memorias con un tiempo de escritura de hasta 1.25 microsegundos.

4.4.2.4 ENTRADAS/SALIDAS:

Según se dijo en la sección 1.4.1.1 durante los ciclos de entrada/salida en CH1, CH2, CH3, CH4 y CH5 se encuentra codificada la dirección del dispositivo de E/S. En el caso de que los dos últimos bits sean '0' la operación es de entrada y en caso contrario de salida.
Se tiene que:

$$PCC = \overline{CH7}.CH6$$

entrada: $E = \overline{CH5}.\overline{CH4}$

salida: $E = \overline{CH5}.\overline{CH4}$

se denominan: $F = PCC.\overline{CH3}$ y

$G = E.\overline{CH2}$

Las señales o0, o1, o2 y o3, que controlan la carga del contenido de CL en los registros de salida O0, O1, O2 y O3, se dan en el tiempo:

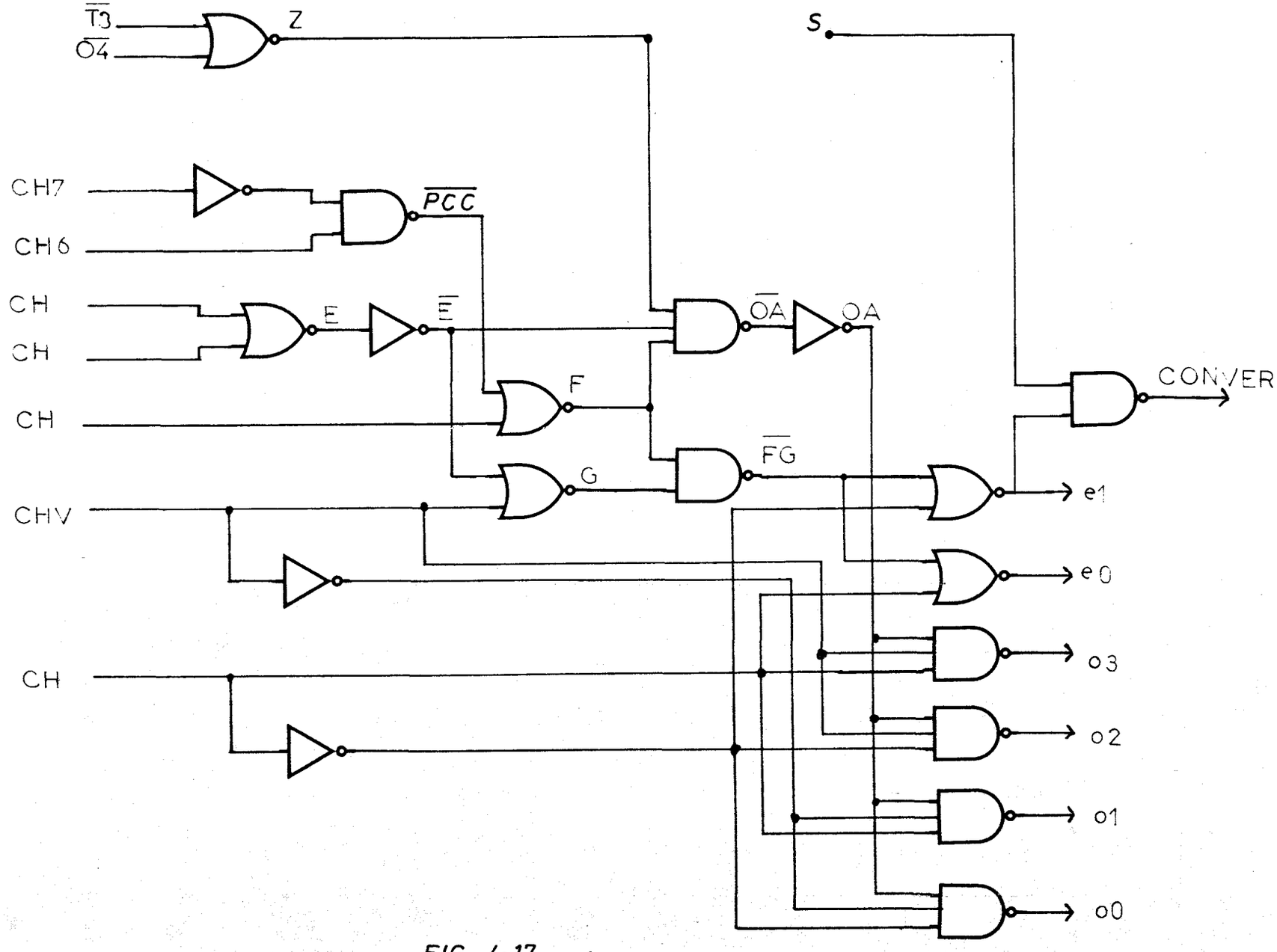


FIG. 4-17

SEÑALES DE CONTROL DE ENTRADA/SALIDA

llamando : $z = T3.\emptyset21.$

$$OA = PCC.\overline{E}.\overline{CH3}.z = F.\overline{E}.z$$

se tiene (figura 4.17):

$o0 = OA.\ \overline{CH2}.\overline{CH1}$

$o1 = OA.\ \overline{CH2}.CH1$

$o2 = OA.\ CH2.\overline{CH1}$

$o3 = OA.\ CH2.CH1$

También es necesario generar, como más adelante se verá, unas señales de control de entradas:

entrada del dispositivo E0: $e0 = F.G.\overline{CH1}$

entrada del dispositivo E1: $e1 = F.G.CH1$

4.4.2.5 BUS DE ENTRADA:

La conexión del bus de entrada al microprocesador debe realizarse de acuerdo con lo indicado en la figura 4.11E; esta figura da el intervalo de tiempo en que el microprocesador debe captar la información de entrada. La señal de control del bus, a, se da cuando SYN es '1' en el estado T3. Como la señal de estado, T3, puede venir retrasada hasta 1.1 microsegundos respecto al final de $\emptyset22$ de T2, se genera la señal TBE (figura 4.11G) y a partir de ella, por intersección con SYN, la función a.

Para obtener la señal TBE se utiliza una báscula D, como indica la figura 4.18. La salida Q es siempre 1 a no ser cuando la señal r("clear") es baja, volviendo a ser Q = 1, cuando llega un nuevo pulso de reloj (función s):

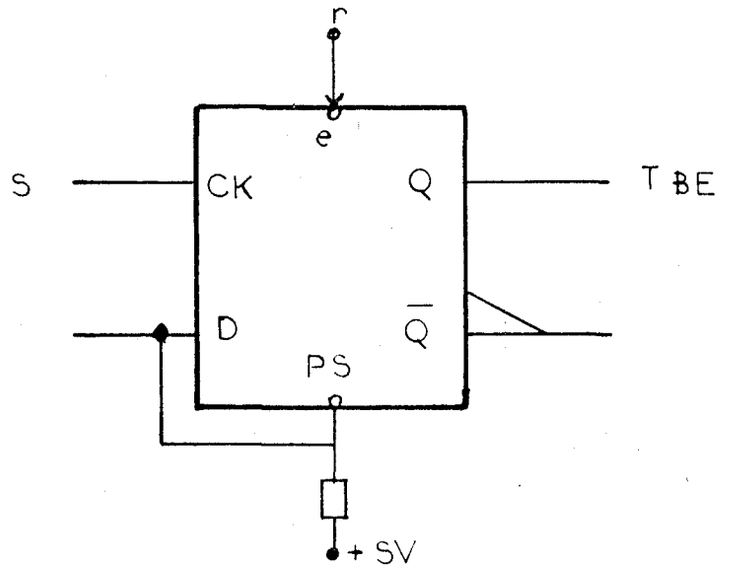


FIG. 4-18

GENERACION DE TBE

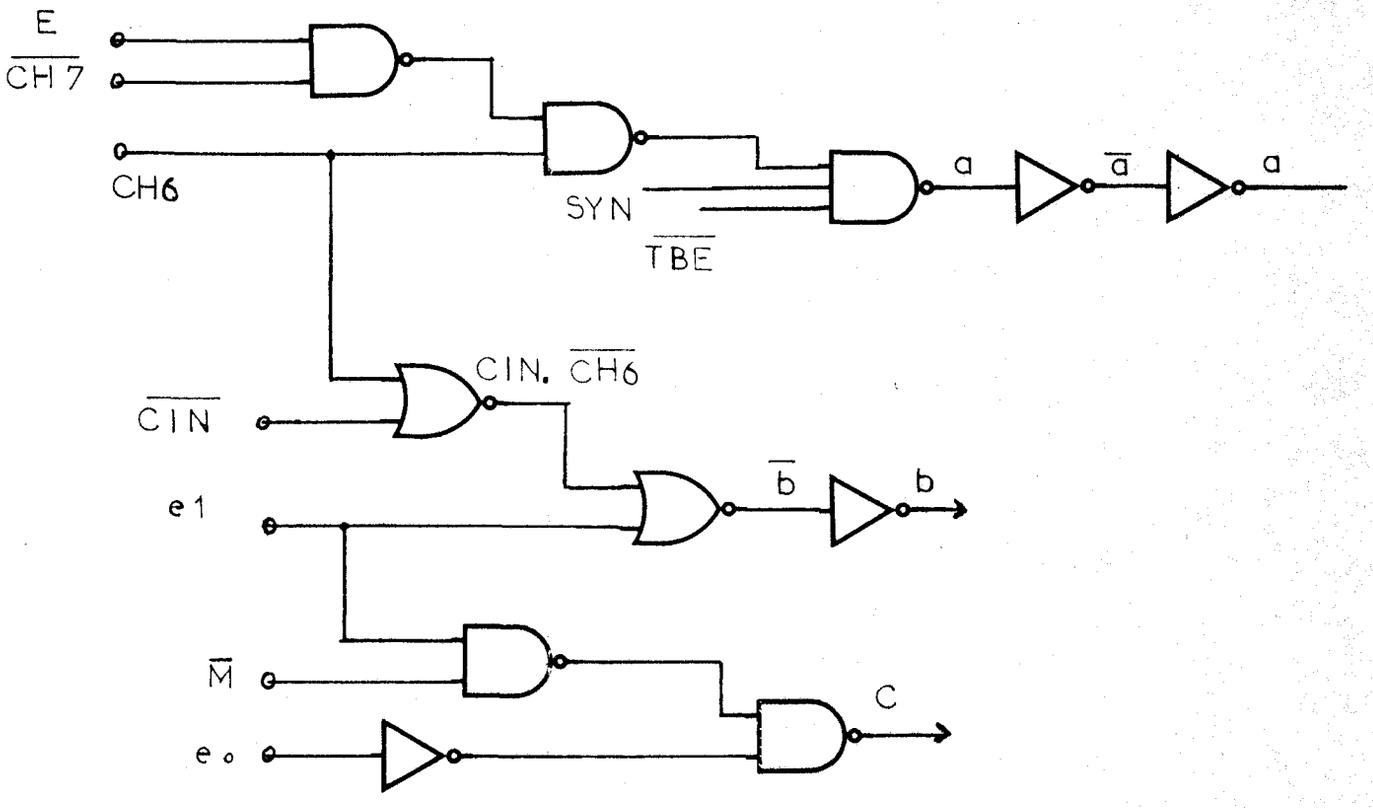


FIG. 4-19

SEÑALES DE CONTROL DEL BUS DE ENTRADA

$$r = \overline{T2.\emptyset 22} = \bar{d}$$

$$s = T3.\overline{SYN}$$

La señal a debe de ser cero cuando se cumplan las siguientes condiciones:

(1) en el intervalo de tiempo BE= $\overline{TBE}$.SYN, y
(2) en los ciclos de entrada de información PCC.$\overline{CH5}$.$\overline{CH4}$=1, o
(3) en los ciclos de lectura de memoria (PCI y PCR);

es decir,

$$\bar{a} = \overline{TBE}.SYN.(PCC.\overline{CH5}.\overline{CH4}+PCI+PCR) =$$

$$= \overline{TBE}.SYN\ (\overline{CH7}.CH6.E+\overline{CH6}.\overline{CH7}.\overline{CH6}.CH7);$$

con lo que:

$$a = \overline{\overline{TBE}.SYN(\overline{CH7}.E+\overline{CH6})}$$

Las señales b y c de direccionamiento del multiplexo hay que definirlas a partir de la tabla de verdad 4.1.

LLamando M a la señal de panel (secc. 2.5) que indica si se consideran como dispositivos de entrada los interruptores del panel (M=1) ó las básculas E1, (M=0), se tiene que:

$$c = e1.\bar{M} + e0$$

$$b = CIN=(PCI + PCR) + e1 = CIN(\overline{CH6}.\overline{CH7} + \overline{CH6}.CH7) +$$

$$+ e1 = CIN.\overline{CH6} + e1$$

En la figura 4.19 puede verse el esquema lógico de la generación de a, b y c.

4.4.3 INTERFASE DEL MICROORDENADOR CON LOS CONVERSORES DE ENTRADA Y SALIDA:

El conversor A/D de entrada (secc. 4.3.4) puede actuar en "modo iterativo", o en "simple conversión". En el primer modo realiza conversiones periódicas sin necesidad de señales de control. En "simple conversión" cada vez que se quiere obtener una conversión es necesario dar un impulso negativo en el terminal CONVER del dispositivo (figura 4.9).

La lógica de interfase se ha realizado utilizando el dispositivo en "simple conversión", ya que fácilmente:

(1) se puede implementar la sincronización entre conversor y microordenador, y

(2) se puede variar la velocidad de conversión, bien sea de forma discreta (por programa), o con una señal de sincronismo externa.

La señal de sincronismo externa esta constituida por un tren de impulsos que se introducen a través del terminal SX de la consola del dispositivo (secc. 2.5). Este tren de impulsos determina el periodo de muestreo T, haciendo entrar en ESPERA el microordenador durante un tiempo TW.

Usualmente se realiza la conversión de un conjunto de puntos que iterativamente se memorizan en posiciones consecutivas de la memoria RAM del microordenador. En esta memorización o tratamiento de las muestras hay que considerar tres tiempos (figura 4.20).

(1) tiempo de conversión (Tc), es el tiempo transcurrido entre el instante de dar la señal de conversión (CONVER) y el instante en que finaliza la misma (STATUS pasa a 0 a 1). Este tiempo es como máximo de 45 microsegundos (figura 4.20B).

(2) tiempo de espera debido al de sincronismo externo (TW).

(3) tiempo de programa (tp), es el tiempo que tarda el miro procesador en ejecutar las instrucciones comprendidas entre dos tomas consecutivas de datos.

El diseño podría hacerse de tal forma que en el momento de necesitar el microprocesador un dato, se diese la señal de conversión y el ordenador entrase en ESPERA hasta finalizar la conversión, de esta manera el tiempo efectivo de conversión (Tec) sería aproximadamente:

$$Tec = Tc + Tp + Tw.$$

Otra posibilidad de funcionamiento, con la que se reduce el tiempo mínimo de conversión y por lo tanto se aumenta el ancho de banda de las funciones susceptibles de ser analizadas por el conversor, consiste en que se realice simultaneamente la conversión y la ejcución del programa, con lo que:

$$Tec = máx.\ (Tc,\ Tp + TW)$$

ó:

$$Tec(Mín) = Máx.\ (Tc,\ Tp)$$

El dispositivo se ha diseñado con la segunda posibilidad.

En principio pueden darse dos casos:

Tc > Tp, el tiempo de conversión es mayor que el de programa; es decir, en el momento de tener que leer un dato se está realizando todavía su conversión (STATUS=0). En este caso habría que hacer entrar el microprocesador en "espera" hasta haber finalizado la conversión:

$$Tec\ (Mín.) = Tp.$$

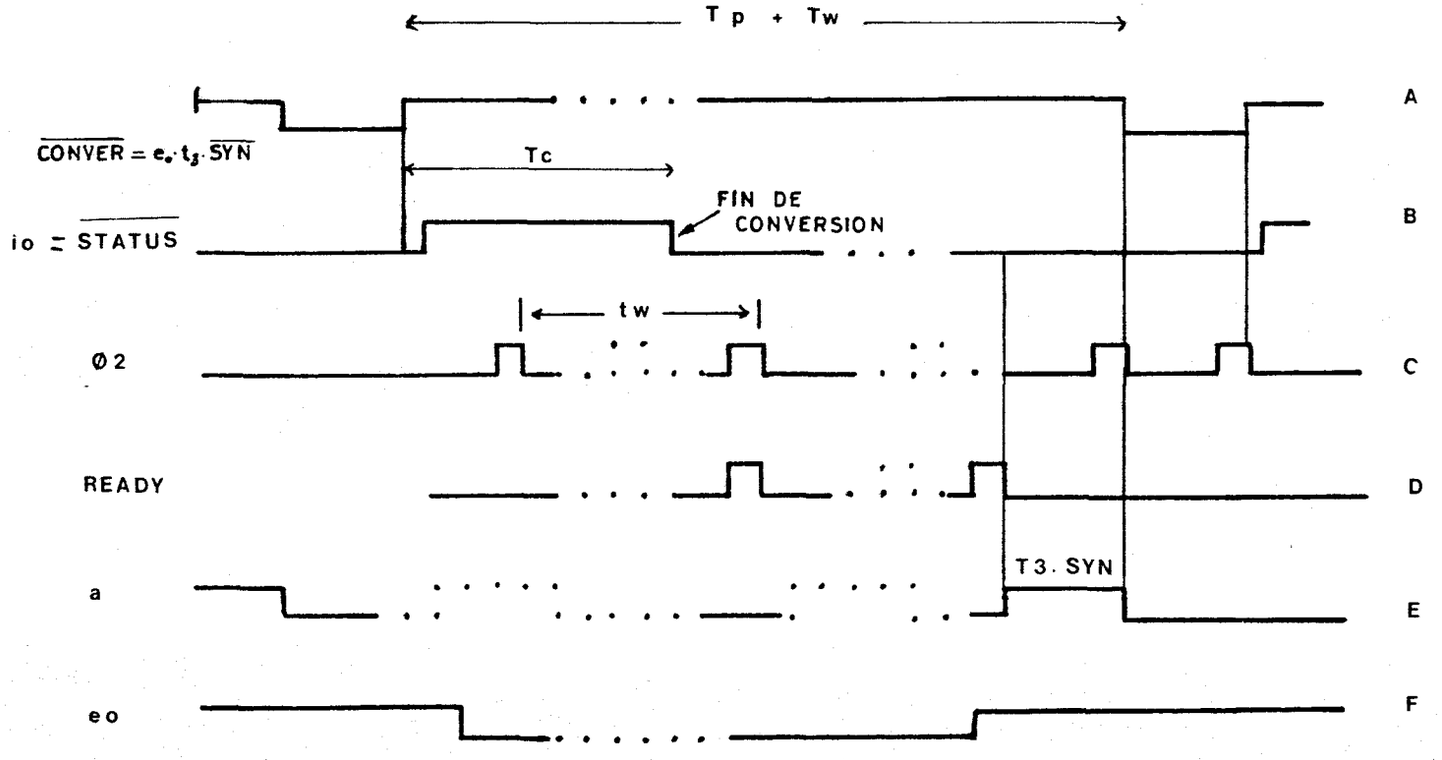


FIG. 4-20

DIAGRAMAS DE TIEMPOS PARA LA INTERFASE DEL CONVERSOR A/D

Tp > Tc, el tiempo de programa mayor que el tiempo de conversión. En este caso el tiempo mínimo posible de conversión viene fijado por el tiempo de programa;

$$Tec\ (mín.) = Tp.$$

Para el microprocesador utilizado se tiene la segunda situación ya que el tiempo de conversión es de unos 45 microsegundos, que equivale a la ejecución de tan sólo unas 3 ó 4 instrucciones de programa, insuficientes para escribir en memoria o efectuar cualquier tratamiento con el dato proporcionado por el conversor. En resumen, la interfase se ha realizado de forma que cuando no se da señal de sincronismo externa T= Tp y cuando se da T = Ts.

En la figura 4.21 puede verse un diagrama de las conexiones lógicas del conversor.

La señal STATUS, que es una salida del conversor, es alta cuando se está realizando una conversión, y baja cuando ha finalizado. (figura 4.20B).

Para implementar la interfase con las especificaciones mencionadas se tienen en cuenta las siguientes consideraciones:

(1) La lectura del conversor es iterativa.

(2) Una demanda de lectura por el ordenador viene dada por la señal de control eo=1 (lectura del dispositivo EO).

(3) El diseño se ha efectuado de forma que cuando la función eO se hace igual a "1" se realizan cronológicamente las siguientes operaciones:

   a) en el intervalo de tiempo dado por a = 1 (figura 4.20E) el microprocesador lee la información del registro EO; es decir, lee el dato de la conversión realizada inmediatamente antes (fig.4.20B,E),

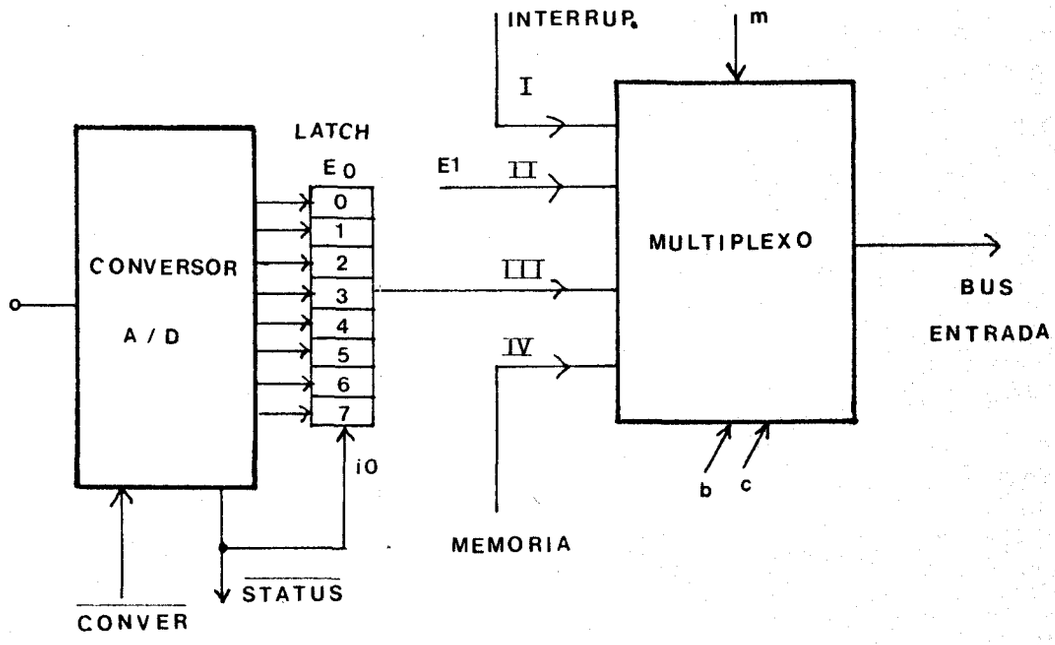


FIG. 4-21

CONEXIONES LOGICAS DEL CONVERSOR A/D

b) posteriormente se da la señal de control para realizar una nueva conversión (CONVER) (figura 4.20A). El resultado de esta conversión es leído en la siguiente instrucción de lectura.

(4) Independientemente del valor de e0 y del estado en que se encuentre el microprocesador cuando finaliza la conversión, (STATUS pasa de 0 a 1, figura 4.20B), se carga el registro EO. Es decir, la señal i0 de carga del registro es precisamente $\overline{\text{STATUS}}$ (figura 4.21). Al ser el registro un "latch" y no un registro "master-slave", durante la conversión cambian los valores de EO y quedan memorizados cuando $\overline{\text{STATUS}}$ pasa de 1 a 0. Como es lógico, mientras dura la conversión el microprocesador no debe captar la información de EO.

(5) en el caso de que, por utilizar un conversor más lento o un microprocesador más rápido, se verifique Tc > Tp, el microprocesador debería entrar en estado de ESPERA hasta finalizar la conversión.

(6) las lecturas del registro EO se hacen dentro de un lazo iterativo, Caso de no utilizar una señal de sincronismo externa y de que Tp > Tc, el periodo de muestreo, T, viene dado por Tp, que es el tiempo que tarda en realizarse una iteración del programa donde se encuentra la instrucción de lectura del conversor. Como T ha de ser constante las iteraciones deben contener siempre las mismas instrucciones.

Es evidente, según lo dicho en (3), que, por leer en primer lugar el dato de la conversión anterior y después dar la orden de una nueva conversión, el primer dato leído no es válido y a pesar de ello, para mantener T constante, debe estar dentro del lazo de iteración del programa y operar con él como si fuese otro dato válido más; sin

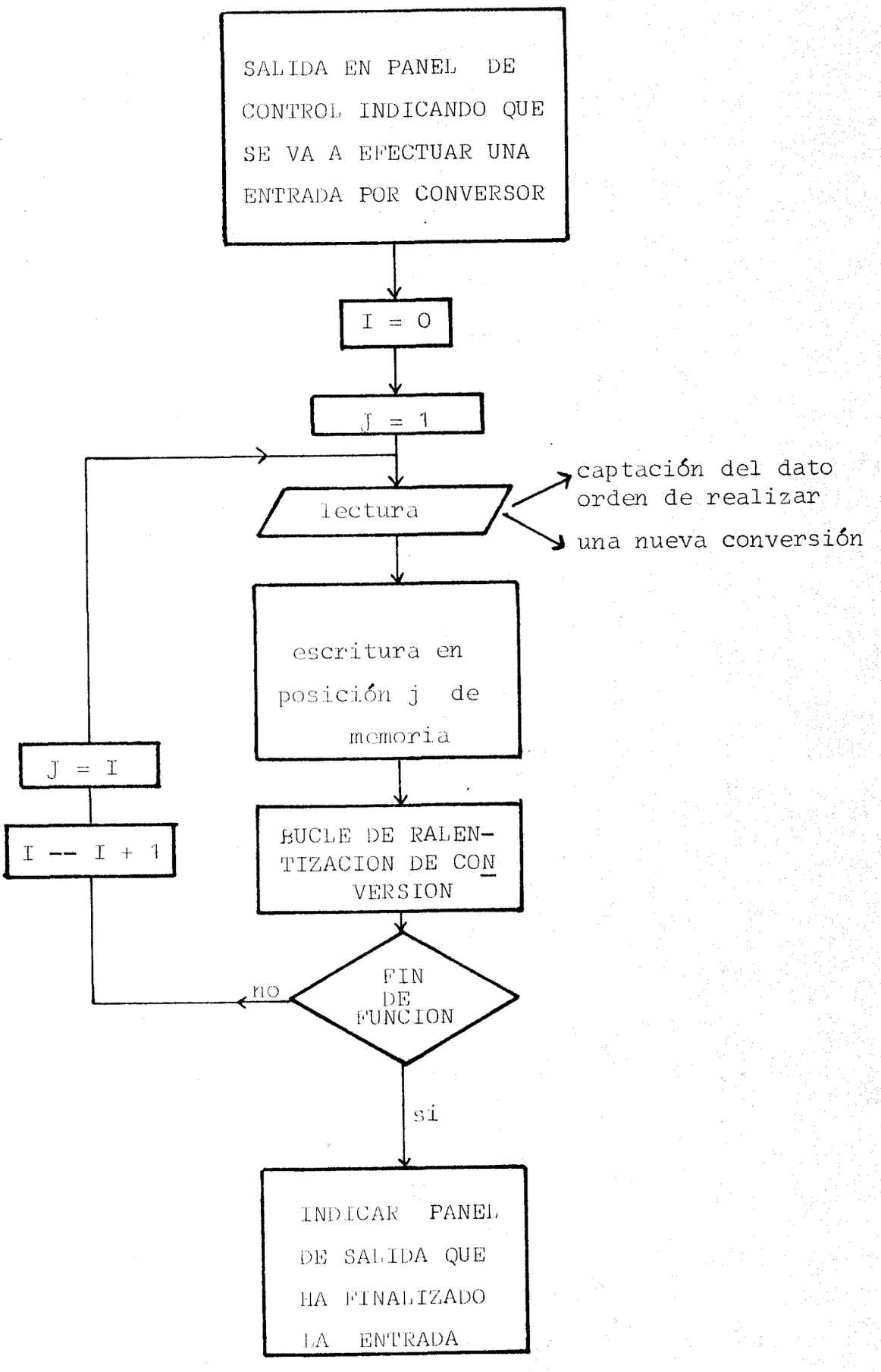


Figura 4.22
forma general de un programa de lectura del conversor A/D.

embargo, este dato, por no ser válido, debe de ser destruido. Los programas que impliquen la lectura de una función tienen que tener en cuenta esta consideración, y por tanto seguir una estrategia similar a la indicada en el organigrama de la figura 4.22. Como se ve el lazo de iteración es común para todas las lecturas, incluso la primera, y esta, por no ser válida, es destruida por la segunda. Los programas del sistema operativo de lectura de funciones (ejecución de INPF Fn) se han realizado de acuerdo con el organigrama de la figura 4.22 (secc. 3.5.3.3).

La función $\overline{\text{CONVER}}$ (figura 4.20A y 4.) se genera a partir de e0, T3 y SYN:

$$\overline{\text{CONVER}} = \overline{\text{e0.T3.SYN}} = \overline{\text{e0.S}}$$

En la sección 4.4.4 se efectuará el estudio y diseño de los circuitos externos de sincronismo.

En la figura 4.23 puede verse como se realiza la interfase con el conversor D/A de salida. En los lazos de los programas de escritura de una función hay que prever que la iteración, o escritura de dos puntos consecutivos, debe durar por lo menos 15 microsegundos, tiempo mínimo de conversión (113).

Las principales ventajas de la lógica de interfase propuesta además de la sencillez de implementación, son:

(1) simultaneidad en la ejecución del programa y conversión. Mientras se realiza una conversión el microprocesador continua trabajando con el programa de lectura o escritura de función. Es decir, se consigue una optimización del periodo mínimo de muestreo posible.

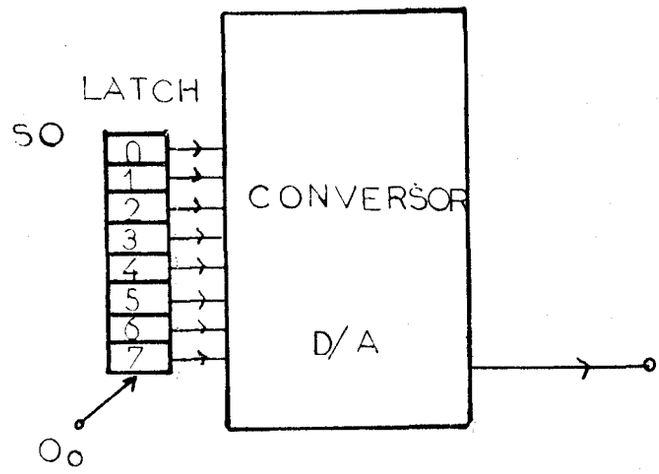


FIG 4-23
INTERFASE DEL CONVERSOR D/A

(2) la conversión se puede realizar bajo el control de un programa, dando numerosas posibilidades de trabajo. Fácilmente, con programas adecuados y con la consiguiente limitación del tiempo efectivo de conversión se puede:

a) trabajar en tiempo real.

b) realizar por programa cualquier tratamiento con los datos obtenidos de la conversión previamente a ser memorizados.

c) realizar, aumentando el número de conversores, una multiplexación entre varios de ellos.

d) variar la velocidad de conversión y el número de puntos a fijar por programa.

e) la posibilidad de realizar mediante programa, la conversión según una función arbitraria logarítmica, por ejemplo.

(3) El periodo de conversión se puede regular y aumentar a partir de un valor mínimo T(mín) por medio de una señal de sincronismo externa (SX).

4.4.4 LOGICA DE INTERRUPCION Y ESPERA:

El diseño de los circuitos de interrupción y espera (figura 4.24) se ha efectuado con las siguientes posibilidades de control desde la consola del microordenador (secc.2.5):

PULSADOR DE ESPERA: la señal que genera este pulsador se denomina PE. Al pulsar ESPERA (PE=1) el microordenador entra en el estado de ESPERA, hasta que es pulsado LISTO.

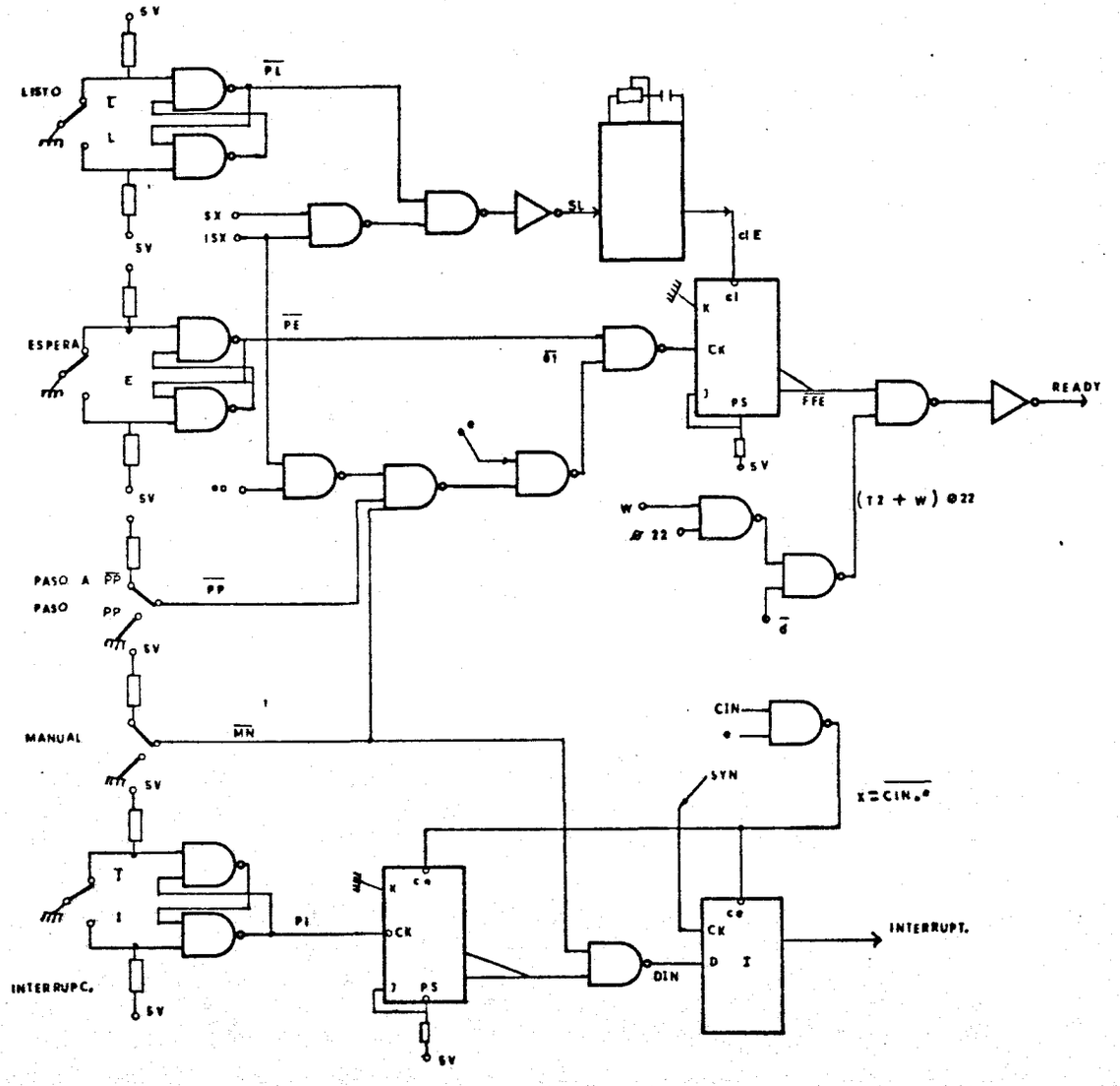


FIG 4-24

CIRCUITOS DE LOGICA DE INTERRUPCION Y ESPERA

INTERRUPTOR DE PASO A PASO: la señal que se obtiene de este interruptor se denomina PP. Cuando PP=1 el ordenador entra en ESPERA. Al pulsar LISTO se ejecuta un ciclo instrucción y vuelve a entrar en ESPERA.

PULSADOR DE LISTO: la señal generada por este pulsador se denomina PL, y con ella (PL=1) puede hacerse salir al microprocesador de ESPERA.

PULSADOR DE INTERRUPCION: con la señal proporcionada por este pulsador, PI, se posibilita que el microordenador pase por un ciclo de interrupción; con ello, tal como se ha realizado el diseño, puede ejecutarse una instrucción máquina o de control depositada en los interruptores de panel, o salir del estado de STOP.

INTERRUPTOR DE FUNCIONAMIENTO MANUAL: con la señal, MN, generada por este interruptor se controlan los circuitos lógicos que posibilitan la combinación de las facilidades que dan la interrupción y espera (secc. 2.5 y 2.6).

El ordenador debe entrar en ESPERA en el ciclo siguiente a aquél en el que haya tenido lugar una salida de información por S1. Esta entrada en ESPERA se realiza con el objeto de que el operador pueda leer la información de salida. Recuerdese que el dispositivo S1 está constituido por indicadores luminosos del panel de control.
Cuando se desea utilizar una señal de sincronismo externa, SX, el interruptor ISX debe colocarse en una posición tal que ISX=1. En este caso, si se está leyendo del conversor A/D, el microprocesador debe de entrar en ESPERA.;cuando llega un pulso de sincronismo (SX = 1) debe salir de ESPERA.

En la figura 4.24 puede verse el esquema lógico que se ha implementado y que a continuación se describe.

A la salida de los pulsadores del panel de control se conectan unas básculas RS para conformar los pulsos generados manualmente. Los pulsos dados en INTERRUPCION y ESPERA se memorizan en básculas JK.

La báscula JK de espera tiene su salida FFE en el nivel lógico "0" salvo cuando se demanda el estado de ESPERA. Las conexiones de este biestable (figura 4.24) se hacen de forma que FFE=1 siempre que, según lo dicho anteriormente, se verifique una de las siguientes posibilidades:

(1) PP=1 , MN=1 ó PE=1 (las señales de control de panel demandan ESPERA)

(2) o1 = 1 (en el ciclo anterior ha tenido lugar salida por S1).

y

(3) ISX.e0 (el control de periodo de muestreo se efectua por sincronismo externo).

Las señales anteriores, exceptuando PE y o1 se dan con sincronismo T1.Ø22=e. La señal de reloj de la báscula de espera JK, por lo tanto, se genera de acuerdo con la siguiente función:

$$ckE = (PP+MN+ISX.e0).e + PE + o1$$

El microprocesador debe de salir de ESPERA cuando se pulse LISTO PL=1) o, en el caso de funcionamiento con sincronismo externo (ISX=1) llegue un pulso de sincronismo (SX=1). La función que actúa sobre el terminal de "puesta a cero" (clear) del biestable JK de demanda de ESPERA debe de ser por lo tanto:

$$SL = \overline{ISX.SX + PL}$$

Esta función se da a la báscula JK por medio de un monostable.

La señal READY, se da al microprocesador siempre que FFE sea cero (no haya demanda de ESPERA), y en sincronismo con (T2 + W).Ø22. Esta sincronización de la señal de READY cumple las especificaciones mencionadas en la secc. 4.4.1. La señal READY, por lo tanto, es:

$$READY = \overline{FFE}.(T2+W).Ø22$$

Los pulsos de demanda de interrupción (PI) se memorizan en otra báscula JK (figura 4.24). La señal de interrupción se da a partir de la báscula JK por medio de una báscula, de tipo D, de forma que se satisfaga la especificación dada en la secc. 4.4.1. La señal de reloj de la báscula D se hace igual a SYN, y la entrada, DIN, se hace "1" siempre que este activado el interruptor de MANUAL (MN=1) o se haya pulsado previamente interrupción (FFI=1):

$$DIN = MN + FFI$$

Las básculas JK y D de la lógica de interrupción se ponen a cero una vez que el microprocesador ha entrado en el ciclo de interrupción (CIN=1) y en sincronización con e; es decir, la señal de puesta a cero es:

$$clI = \overline{CIN.\ e}$$

Al conectar el ordenador a la red (secc. 2.6), este entra automáticamente en estado de STOP, siendo necesario para su puesta en marcha pulsar INTERRUPCION.

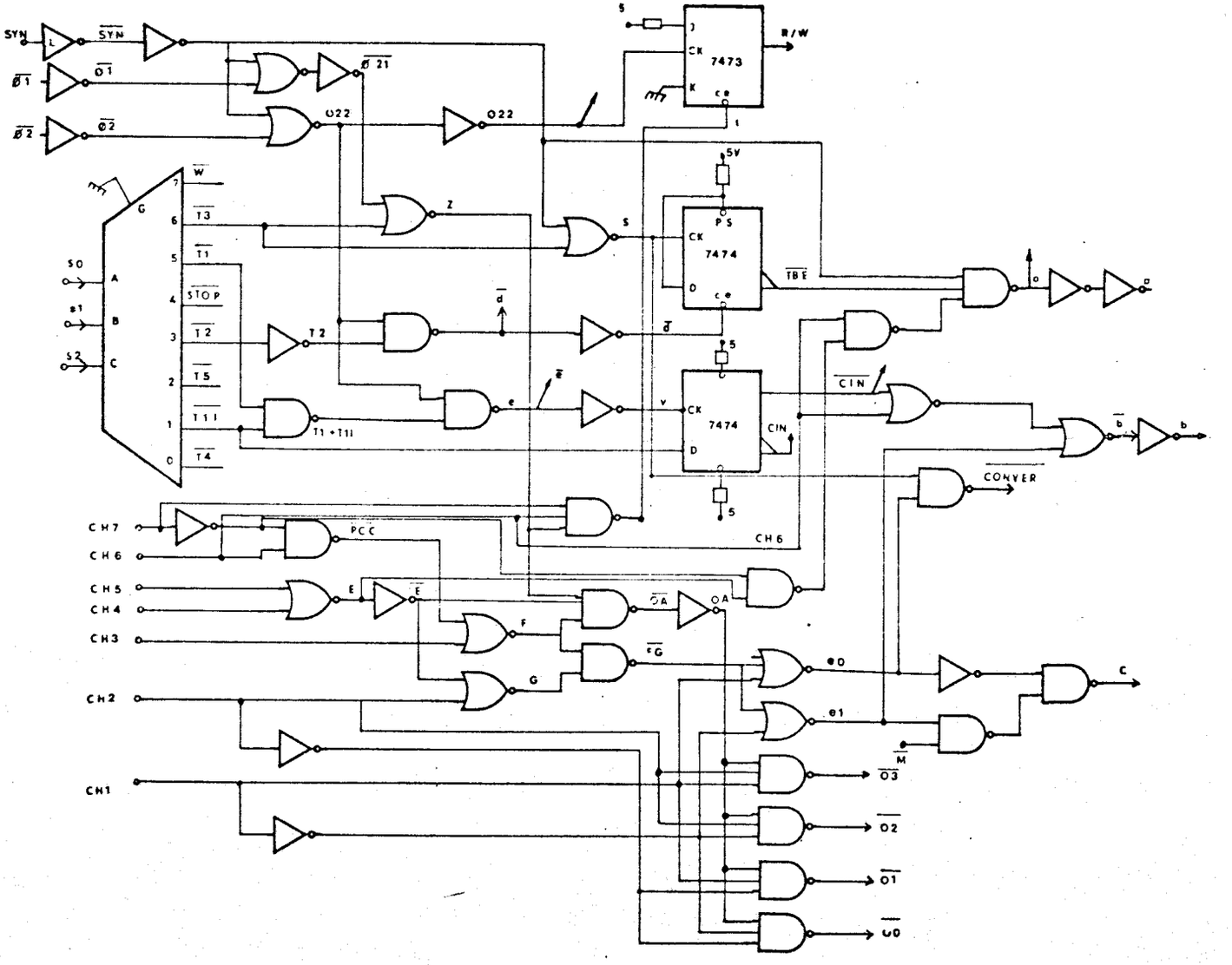


FIG. 4-25

CIRCUITOS QUE GENERAN LAS SEÑALES DE CONTROL

4.4.5 ESQUEMA GENERAL:

Las funciones que se han tenido que sintetizar, según los epígrafes anteriores, han sido las que se indican en la tabla 4.4.

Función	Sección
$\emptyset 21 = SYN, \emptyset 2$	4.4.1
$\emptyset 22 = \overline{SYN} . \emptyset 2$	4.4.1
$v = (T1+T1I).\emptyset 22$	4.4.2.1
$e = (T1+T1I).\emptyset 22 = v$	4.4.2.2
$d = T2.\emptyset 22$	4.4.2.2
$t = \overline{PCW} . T3.\emptyset 21$	4.4.2.3
$PCW = CH7.CH6$	4.4.2.3
$PCC = \overline{CH7} . CH6$	4.4.2.4
$E = \overline{CH5}.\overline{CH4}$	4.4.2.4
$F = PCC . \overline{CH3}$	4.4.2.4
$G = E.\overline{CH2}$	4.4.2.4
$z = T3.\emptyset 21$	4.4.2.4
$OA = F.\overline{E}.z$	4.4.2.4
$o0 = OA.\overline{CH2}.\overline{CH1}$	4.4.2.4
$o1 = OA.\overline{CH2}.CH1$	4.4.2.4
$o2 = OA.CH2 . \overline{CH3}$	4.4.2.4
$o3 = OA.CH2.CH3$	4.4.2.4
$e0 = F.G.\overline{CH1}$	4.4.2.4
$e1 = F.G.CH1$	4.4.2.4
$r = \overline{d}$	4.4.2.5
$s = T3.\overline{SYN}$	4.4.2.5
$a = \overline{TBE}.SYN.(\overline{CH7}.E+\overline{CH6})$	4.4.2.5
$c = e1.\overline{M}+e0$	4.4.2.5
$b = CIN.\overline{CH6} + e1$	4.4.2.5
$\overline{CONVER} = e0.s$	4.4.3
$ckE = \underline{(PP+MN+ISX} . e0)e+PE+o1$	4.4.4
$SL = \overline{ISX}.SX + PL$	4.4.4
$READY = FFE.(T2.W).\emptyset 22$	4.4.4
$DIN = MN+FFI$	4.4.4
$clI = \overline{CIN}.e$	4.4.4

Tabla 4.4

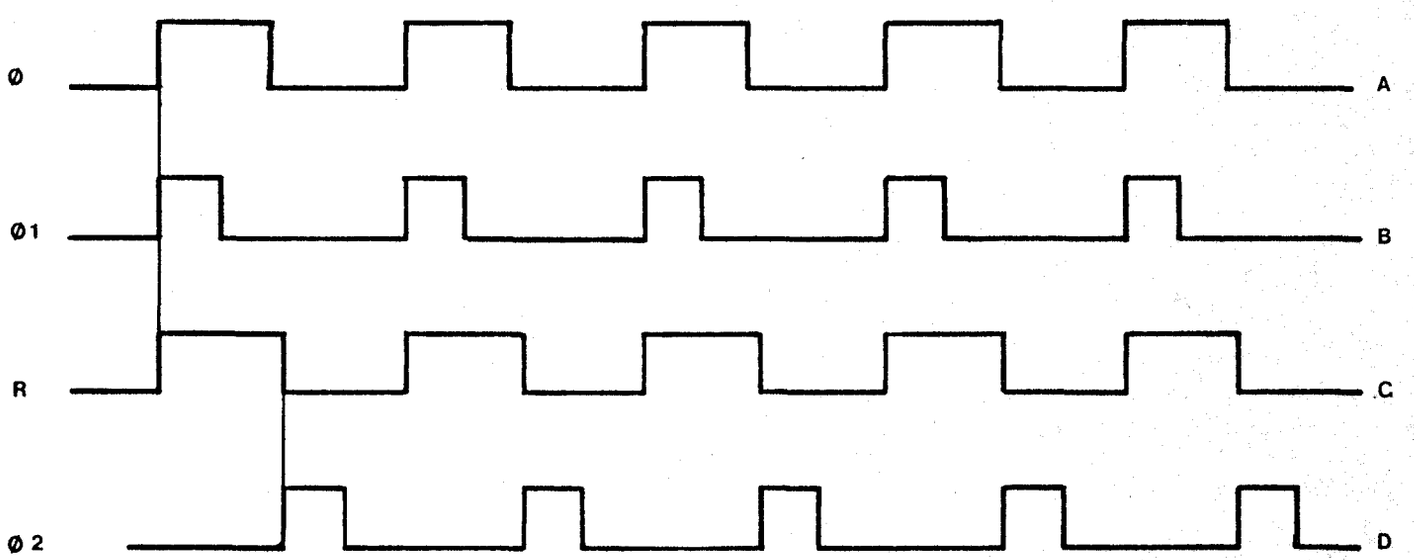


FIG. 4-26

DIAGRAMA DE TIEMPOS DE RELOJES

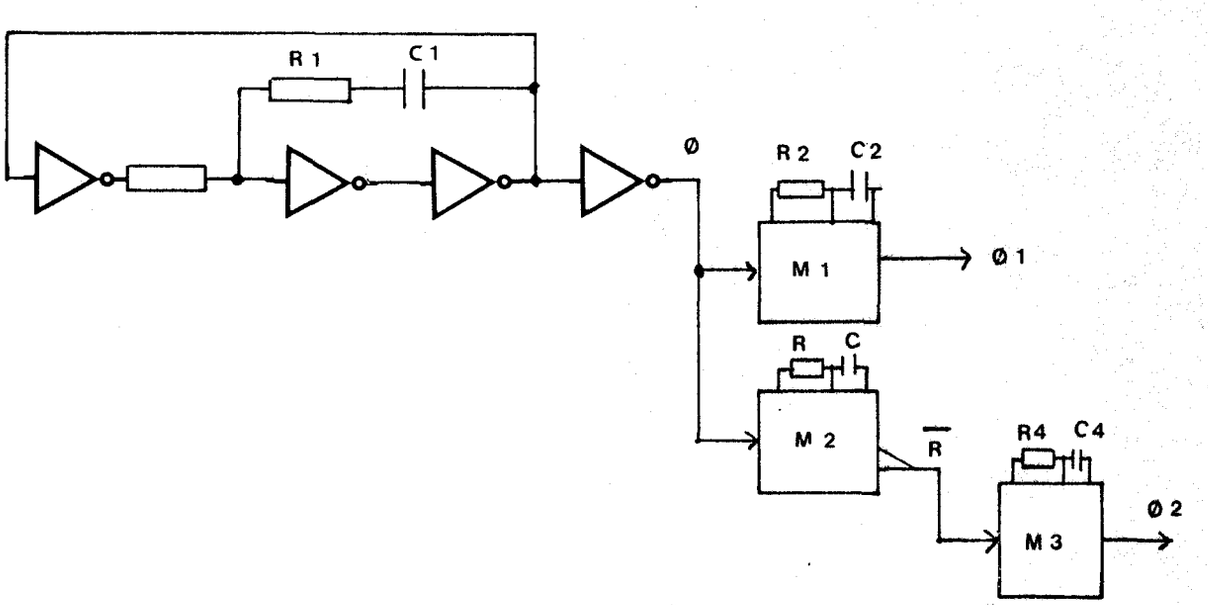


FIG. 4-27

ESQUEMA LOGICO DE RELOJES

En las figuras 4.24 y 4.25 pueden verse los esquemas de los circuitos que generan estas funciones booleanas.

4.4.6 RELOJES:

Las señales de reloj Ø1 y Ø2 deben verificar las especificaciones dadas en la secc. 4.4.1. El diseño de ha efectuado de forma que la anchura de los pulsos Ø1 y Ø2, y el desfase entre ellos pueda ser ajustado.

En primer lugar, figura 4.27, mediante cuatro inversores se realiza un oscilador (31) que produce una onda Ø casi cuadrada (figura 4.26A). Su frecuencia se puede ajustar mediante el condensador C1.

A partir de la señal Ø, figura 4.27, y mediante tres monostables (M1, M2, y M3) se generan las señales R. Ø1, y Ø2. Los monostables son activados con los frentes ascendentes de los pulsos de entrada y producen unos pulsos de duración (89):

$$t = R.C.\log_e 2$$

Tal y como se ha realizado el montaje (figura 4.26 y 4.27), con la resitencia variable,

R2 se ajusta al ancho de Ø1.
con la R3 se ajusta R, es decir, el desfase entre Ø2 y Ø1, y
con la R4 se ajusta el ancho de Ø2.

Los parámetros de los componentes utilizados son:

R1 = 330 ohmios, C1 = 4.7 nF, R2 y R4 = 10 kohmios,
R3 = 50 kohmios y C1, C3 y C4 = 100 pF.

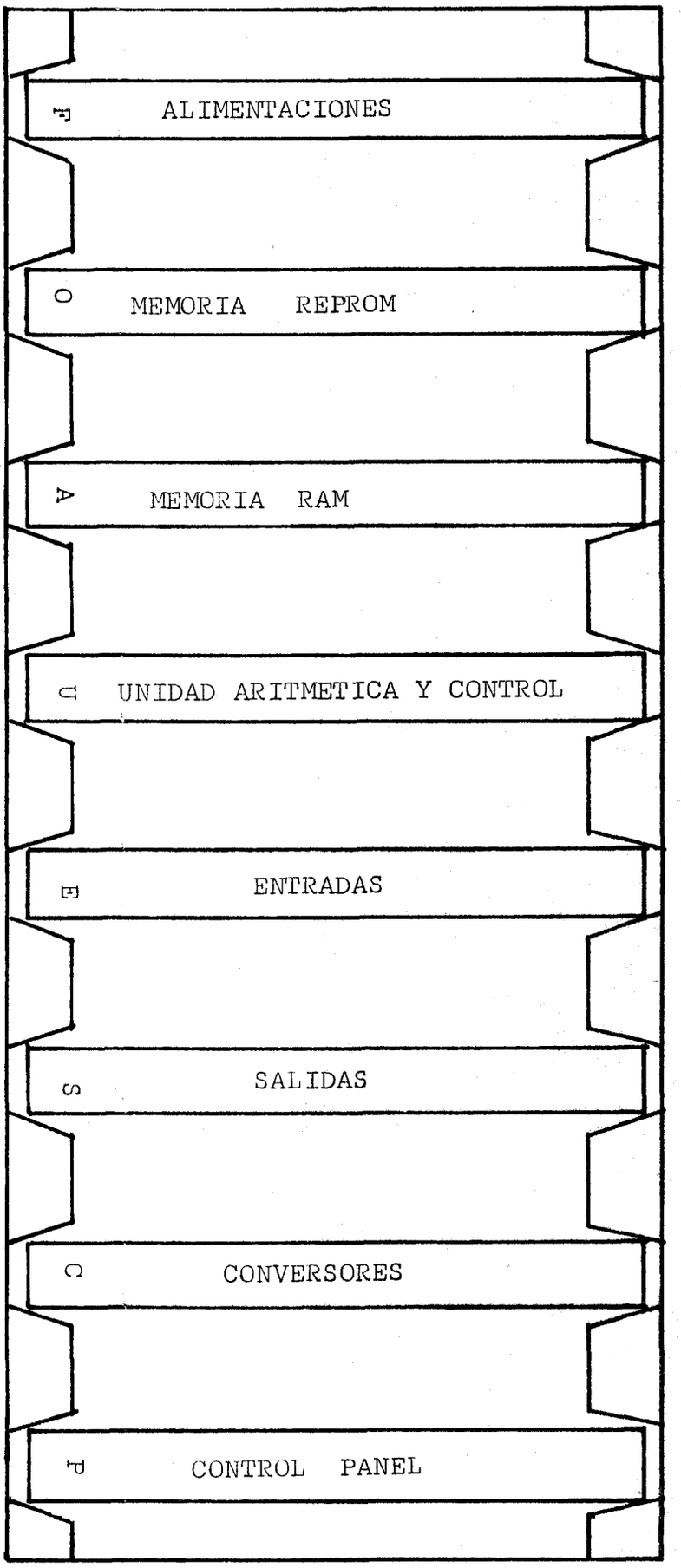


Figura 4.28
SITUACION DE TARJETAS

4.5 REALIZACION, MONTAJE Y MANTENIMIENTO DEL DISPOSITIVO:

Los circuitos que constituyen el ordenador se han montado en su totalidad sobre zócalos soldados en tarjetas estándar de 79 terminales tipo VERO NO.12761, cada una de las cuales se dedica a:

Tarjeta F: fuentes de alimentación
tarjeta O: memoria REPROM
tarjeta A. memoria RAM
tarjeta U: unidades aritmético-lógica y de control.
tarjeta E: entradas
tarjeta S: salidas
tarjeta C: conversores
tarjeta F: control de panel.

El montaje se ha realizado con gran modularidad; las tarjetas se introducen por medio de unas guías en una caja metálica, uniéndose entre ellas y al resto del sistema por medio de unos conectores VERO 10859-4 7351. La disposición de las tarjetas se ha realizado en la forma que se indica en la figura 4.28.

En el panel van situados los indicadores luminosos, los cuales son de baja potencia y son activados directamente por inversores en colector abierto. Se utilizan para el panel interruptores y conectores miniaturizados.

En el apéndice V se incluye una lista de los componentes utilizados en cada tarjeta y las señales de entrada/salida ("pins") de cada conector.

Nº de pastilla	Referencia	Función a realizar	Tecnología	Marca	Nº circuitos Ftes./pastilla	Nº total Ctos.Ftes.
1	C8008	microprocesador	MOS dinamica	Intel	-	-
1	4114	conversor A/D	microcirc.	Teledyne-Philbrick	-	-
1	4021	conversor D/A	microcirc.	Teledyne-Philbrick	-	-
8	C1702A	memoria REPROM	MOS estática	Intel	256x8 bits	16.384
24	C2102A	memoria RAM	MOS estática	Intel	1024 bits	24.576
8	7400	NAND	TTL	Texas-National	4	32
3	74L00	NAND low power	TTL	"	4	12
3	7402	NOR	TTL	"	4	12
8	7404	Inversores	TTL	"	6	48
5	7405	inver.colec.abierto	TTL	"	6	30
1	7408	AND	TTL	"	4	4
4	7410	NAND,3 entradas	TTL	"	3	12
1	7420	NAND,4 entradas	TTL	"	2	2
2	7472	JK master-slave	TTL	"	1	2
1	7473	JK master-slave	TTL	"	2	2
1	7474	báscula D	TTL	"	2	2
12	7475	Latch	TTL	"	4	48
2	7476	Exclusive-OR	TTL	"	4	8
4	74121	Monostable	TTL	"	1	4
2	74125	Buffer tres estados	TTL	"	4	8
4	74153	multiplexo 4:1	TTL	"	2	8
1	74154	decodificador 4:16	TTL	"	1	1
2	74155	decodificador 3:8	TTL	"	1	2
1	74175	báscula D	TTL	"	4	4
2	74198	registro 8 bits	TTL	"	4	2
112	T O T A L					41203

Tabla 4.5

CIRCUITOS INTEGRADOS UTILIZADOS

En el apéndice VII se presenta una serie de fotografías del dispositivo.

En la tabla 4.5 se indican los circuitos utilizados.

Se considera suficiente la documentación incluida en el apéndice V y en este capítulo para la realización del mantenimiento del ordenador.

4.6 SINOPSIS:

Este capítulo ha recogido los aspectos relacionados con el diseño y realización hardware del microordenador.

Se ha considerado la estructura de la memoria, circuitos básicos y circuitos de control. También se ha presentado la síntesis de todas las funciones a que da lugar el diseño del dispositivo.

Finalmente se exponen algunas consideraciones sobre el montaje del sistema.

CAPITULO V

CONSIDERACIONES FINALES Y CONCLUSIONES:

5.1 INTRODUCCION:

Este último capítulo pretende resumir las características generales y técnicas más impotantes del microordenador realizado, efectuar unas críticas generales sugiriendo ciertas optimizaciones y resaltar las principales conclusiones y aportaciones de la investigación.

5.2 CARACTERISTICAS GENERALES DEL MICROORDENADOR:

El ordenador puede operar de dos formas:

a) como un microordenador de proposito general o
b) como un microordenador para operar con señales analógicas.

El dispositivo se ha realizado utilizando un circuito integrado que contiene un microprocesador (INTEL 8008) y, por lo tanto, la mayor parte de sus características vienen fijadas por el microprocesador.

Las características fundamentales son:

MEMORIA:

Capacidad: configuración inicial: 5K x 8 bits
(1K=1.024)
ampliable a: 16K x 8 bits.

Tecnología:-2K de memoria REPROM (reprogramable Read Only Memory) de silicio tecnología MOS: 16 pastillas de 256 x 8 bits, Intel 1702A.
-3K de memoria RAM (Random Access Memory) de silicio, tecnología MOS: 24 pastillas de 1024 bits, Intel 2101A.

Tiempo de acceso máximo:

- RAM: 350 nanosegundos.
- REPROM: 1 microsegundo.

LOGITUD DE PALABRA:

8 bits.

TRANSMISION DE INFORMACION:

Transmisión en paralelo. La conexión con el microprocesador se hace a través de un bus bidireccional de 8 bits. que se bifurca en dos, uno de entrada y otro de salida.

PERIODO DE RELOJES:

Ajustable de 2 a 3 microsegundos.
Los relojes generan dos señales desfasadas, de acuerdo con las especificaciones que se dan en la secc. 4.4.6.

TIEMPO DE EJECUCION DE INSTRUCCIONES:(para un periodo de reloj a 2 microsegundos)

Variable de 10 a 22 microsegundos, incluyendo los ciclos de lectura de instrucción y dependiendo de la operación a ejecutar.

REGISTROS INDICES A LOS QUE SE TIENE ACCESO:

Desde el lenguaje LM: 7 registros (tipo "scratch pad")
Desde el lenguaje LPF: 3 registros.

NIVELES DE ANIDAMIENTO DE SUBRUTINAS:

a) Desde el lenguaje LM: 8 nivels: anidamientos realizados por hardware, en una memoria pila (stacK).

b) Desde el lenguaje LPF: 8 niveles. Los anidamientos se hacen por software.

TIPOS DE INSTRUCCIONES:

a) Lenguaje LM: pueden darse hasta 227 instrucciones distintas, agrupadas en:

. instrucciones de transferencia de información:

en estas instrucciones el direccionamiento puede ser (ver apéndice II):

- implícito o
- implícito indexado.

. instrucciones de operaciones aritméticas y lógicas:

Las operaciones se realizan entre el contenido del acumulador y otro dato proporcionado mediante un direccionamiento de los tipos:

- implícito.
- implícito - indexado o
- inmediato.

. instrucciones de bifurcación: estas instrucciones pueden ser condicionales o no y el direccionamiento se hace únicamente de forma absoluta.

. Otras instrucciones: instrucciones sin referencia a dirección de memoria ("zero addres instruction"), como son E/S, alto,...

b) Lenguaje LPF: existen 90 instrucciones agrupadas en:

. de transferencia de información:

a) con funciones: direccionamiento simbólico-inmediato.

b) con números:

. direccionamiento implícito-indexado-relativo.
. direccionamiento implícito
. inmediato.

. operaciones aritmético-lógicas:

a) con funciones: direcc. simbólico-inmediato.

b) con números:

. direccionamiento implícito.

. direccionamiento inmediato-relativo.

. inmediato.

. de bifurcación:

. direcc. inmediato-relativo.

. direccionamiento implícito-indexado-relativo.

. Otras: (entrada/salida,...)

FORMATO DE INSTRUCCIONES:

a) Lenguaje LM: (ver sección 2.2.1)

. formato de instrucciones con direccionamiento implícito (1 octeto).

. formato de instrucciones inmediatas (2 octetos).

. formato de instrucciones de direccionamiento absoluto (3 octetos).

b) Lenguaje LPF: (ver secc. 2.3.1)

. formato de instrucciones implícitas (1 octeto)

. formato de instrucciones de direccionamiento inmediato-simbólico (2 octetos)

. formato de instrucciones inmediatas (2 octetos)

. formato de instrucciones de direccionamiento relativo (2 octetos).

CONSOLA O PANEL DE CONTROL:

- lógica de "INTERRUPCION" y "ESPERA", con la posibilidad de trabajar en "automático", "paso a paso" y "manual".

- Indicadores luminosos de

  . ciclo en ejecución
  . dirección de memoria
  . mensajes de operador y dispositivo de salida (S1)

- Interruptores para:

  . dar entradas en ciclos de interrupción o como dispositivo de entrada (E1).
  . alimentación general, alimentación de conversor A/D y alimentación del conversor D/A.
  . determinar si el dispositivo de entrada está formado por los interruptores del panel o por el teletipo.

- Conectores: para salida y entrada de sincronismos.

ENTRADAS Y SALIDAS:

Entradas: tres dispositivos:

  . conversor A/D (E0)
  . interruptores de consola (ciclos de interrupción)
  . interruptores de consola/teletipo (E1).

pueden ampliarse hasta 8.

Salida: cuatro dispositivos:

  . conversor A/D (S0)
  . indicadores luminosos de consola (S1)
  . otros dos dispositivos (S2, S3), para conectar a teletipo,...

pueden ampliarse hasta 24.

SISTEMA OPERATIVO:

El programador puede utilizar los siguientes lenguajes:

lenguaje máquina del microordenador (LM)

lenguaje orientado a operar con funciones (LPF)

lenguaje de control (LC)

lenguaje ensamblador cruzado (LE)

Los programas se dan al microordenador codificados en binario.

El Sistema Operativo está constituido por programas monitores y traductores. Todo él está estructurado como un "intérprete", y se compone de:

- 7 programas 44 subrutinas que constituyen el MONITOR.
- 90 rutinas o macros correspondiente al lenguaje LPF.
- 16 subrutinas elementales y
- 19 subrutinas auxiliares.

El S.O. puede dar por consola 14 mensajes de errores, diagnósticos o demandas de información.

5.3 CONSIDERACIONES Y CRITICAS GENERALES. OPTIMIZACION QUE SE SUGIEREN:

A partir del dispositivo realizado, y siguiendo la filosofía de diseño expuesta, se podría concebir un modelo de microordenador comercial.

Las limitaciones que se deducen de la lectura de este epígrafe vienen dadas , en general, por el hecho de que el diseño se ha efectuado utilizando componentes de fácil adquisición en el mercado, y dentro de unas disponibilidades económicas.

Una de las limitaciones más importantes del prototipo realizado es la velocidad operativa del microprocesador. El ciclo de reloj es de 2.5 microsegundos, siendo necesario para ejecutar una instrucción un término medio de 14 ciclos. Por otro lado la velocidad de los conversores, del orden de 45 microsegundos, es relativamente pequeña permitiendo el procesamiento de sólo señales de frecuencias bajas.

En determinados casos con circuitos integrados más perfeccionados se reduciría la complejidad del diseño; asi, por ejemplo, utilizando un multiplexo de tres estados ("three state") en el bus de entrada al microprocesador se necesitarían 8 circuitos integrados menos, y disminuiría el "fan-out" necesario para la señal de control n.

Desde el punto de vista del "software", a pesar de la potencialidad del lenguaje LPF puesta de manifiesto en la secc. 2.3.3., hay que decir que tiene una estructura muy elemental que se podría denominar de "lenguaje macromáquina", ya que las instruc

ciones deben darse codificadas en binario y son, exceptuando las relativas a operar con funciones, muy elementales. Tal vez sería conveniente en un ulterior trabajo dotar al microordenador de un nivel superior de software que utilizase como lenguaje el actual LPF.

El número de rutinas del S.O. podría reducirse, aunque la complejidad, tiempo y tal vez capacidad de memoria aumentaría, con la realización de un "analizador", de forma que antes de efectuar la traducción por el intérprete de la instrucción LPF se le hiciese un tratamiento previo. El analizador evitaría, por ejemplo, el que para instrucciones del lenguaje LPF similares a las del lenguaje máquina se haya tenido que seguir la misma sistemática de otras instrucciones más complejas, direccionando en todo caso la rutina que biunívocamente le corresponde a la macroinstrucción.

Siguiendo la línea de investigación de este trabajo y como optimización para la realización de un microordenador de la misma naturaleza se sugiere:

1º) La utilización de un microprocesador más rápido y con un lenguaje básico más potente. Por ejemplo, con circuitos de la serie INTEL 3000 se conseguiría:

a) que el tamaño de palabra no venga impuesto por el microprocesador. Con la serie citada se puede organizar el microordenador en 2,4,6,8,12,14,... bits.

b) que el periodo de reloj fuese mucho más rápido: hasta 70 nanosegundos, consiguiendose una velocidad operativa cerca de mil veces superior al microprocesador utilizado.

c) que los circuitos auxiliares fuesen mucho menos complejos; por ejemplo, sólo haría falta una alimentación y un sólo reloj. Al tener tres buses de entrada y dos de salida se evitarían gran cantidad de dificultades hardware.

d) desde el punto de vista operativo el microprocesador contiene 11 registros y un acumulador.

e) utiliza internamente la técnica de microoprogramación pudiendose definir arbitrariamente microfunciones en bloques de 512.

f) debido a la mayor potencia del lenguaje básico, el S.O. resultaría mucho más simplificado.

2º) La utilización de conversores más rápidos y con un número mayor de bits; esto último permitiría una mayor precisión en las E/S y un error de cuantización menor, El uso de un microprocesador y conversores más rápidos facilitarían el funcionamiento del ordenador en tiempo real. (Con el conversor Teledipe 4107, de 10 bits, se conseguiría una velocidad de conversión de 7 micro segundos).

3º) La dotación de varios conversores A/D de entrada permitiría el operar simultáneamente con varias señales y el hacer con facilidad multiplexaciones en el tiempo.

4º) Realizar una lógica adecuada para poder leer o escribir en memoria bajo el control total de panel, y sin intervención del microprocesador ("off line"). De esta forma la carga de programas y su ulterior comprobación se efectuaría de forma más versátil. También la lectura y escritura de funciones en memoria podría realizarse con más amplias posibilidades.

No obstante esta mejoras, las ideas de concepción y diseño podrían permanecer inalterables.

5.4 CONCLUSIONES Y PRINCIPALES APROTACIONES:

Como punto final de esta memoria resulta conveniente hacer un breve comentario sobre la investigación desarrollada, destacando las principales aportaciones realizadas al campo de la arquitectura de ordenadores.

Todos los objetivos propuestos, enumerados al principio de la memoria, se han cubierto plenamente con resultados satisfactorios.

El trabajo ha llevado consigo la realización de un microordenador de propósito general al que se le ha dotado complementariamente de elementos, tanto hardware como software, que le permiten operar directamente con señales analógicas. Tal vez uno de los aspectos fundamentales radica en la versatilidad que resulta de operar con funciones a través de un lenguaje de programación.

Algunas de las facilidades que proporciona el dispositivo son:

a) se le ha dotado de un leguaje en el que existen instrucciones que operan globalmente con funciones, sin tener que hacer un tratamiento punto a punto. Estas instrucciones, desde el punto de vista operativo, son de alto nivel (integral, derivada, traslación,...).

b) puede trabajar o no en tiempo real.

c) cuando se memoriza una función el periodo de muestreo, así como el número de muestras, puede ser ajustado con facilidad por programa.

d) puede trabajar como un ordenador de propósito general.

Toda la filosofía de concepción, estructura y diseño se ha realizado expresamente para la consecución de los objetivos señalados a lo largo de la memoria, buscando en todo momento soluciones originales que superasen a las convencionales. En numerosos casos a estas soluciones se les ha pretendido dar una formalización y sistemática que las haga completamente generales.

Algunos aspectos y facetas más significativos y destacados son:

I. HARDWARE:

Se ha estructurado y diseñado un ordenador digital y con posterioridad se ha montado, probado y ajustado obteniendose resultados plenamente satisfactorios y en total acuerdo con el planteamiento realizado. Algunas características son:

a) para el diseño se han utilizado componentes plenamente vigentes:

microprocesador, memorias REPROM, memorias RAM tipo MOS estático,....

b) todo el dispositivo se ha realizado con circuitos integrados tanto de pequeña, de mediana como de gran escala. Dentro de la gama de estos circuitos se han utilizado, sin problemas de interconexión, diversas familias y tecnologías:

TTL normal
TTL "low power"
TTL colector abierto
TTL "Three-state"
TTL dinámico
TTL estático.

c) se han realizado interfases con conversores A/D y D/A, de entrada y salida, respectivamente.

II. SOFTWARE:

Se ha dotado al dispositivo de un sistema operativo. La labor realizada en este campo se ha efectuado partiendo de técnicas actuales, tales como paginación, microprogramación, "firmware", ... Por otra parte la aritmética incorpora las ventajas de la coma flotante, efectuandose ecualizaciones, normalizaciones,... También se han tenido en cuenta los problemas que lleva consigo la propagación de errores.

El sistema operativo está dotado de:

a) Programas monitores para el funcionamiento del sistema como un microordenador de propósito general.
b) Rutinas o macros para operar eficientemente con funciones.
c) Un programa supervisor que se encarga de interpretar las instrucciones de un lenguaje definido para operar con funciones. Este programa hace corresponder biunívocamente a cada código de operación una macro.
d) Rutinas apropiadas para dar mensajes tanto de diagnósticos como de errores.

Otras realizaciones dentro de este campo han sido:

a) Se ha definido, por medio de una terminología completamente general y fácilmente extensible a todo tipo de mini y microordenadores, un lenguaje orientado a operar con funciones (LPF), un lenguaje ensamblador (LE) y un lenguaje de control (LC).

b) El lenguaje para operar con funciones contiene una instrucción que permite insertar dentro de programas escritos en el mencionado lenguaje rutinas escritas en lenguaje máquina.

c) Se ha realizado un ensamblador cruzado ("cross assembler"), que ha sido implementado en lenguaje BASIC en un ordenador UNIVAC 1108, y que genera programas con direccionamiento absoluto en códigos binario y hexadecimal. Aprovechando esta facilidad, el S.O. ha sido escrito en lenguaje ensamblador.

d) A partir de los factores "tiempo de ejecución" y "capacidad de memoria" se dan criterios formalizados y completamente generalizados sobre la conveniencia o no de la definición de subrutinas.

III. FIRMWARE:

La filosofía de implementación del sistema operativo es análoga a la utilizada en la técnica de la microprogramación. El sistema operativo ha sido grabado en memorias REPROM. Algunos aspectos fundamentales de esta faceta de la investigación son:

a) Se aportan ideas originales en cuanto a conseguir una sistemática en la imbricación de microprogramas.

b) Se definen parámetros para la determinación de capacidad y eficiencia, en general de grabación de memorias ROM.

c) Se esboza un método para el estudio lógico de llamadas y anidamientos entre subrutinas. La metodología propuesta presenta una completa generalidad.

Por último, en base a la experiencia adquirida, se sugieren optimizaciones al prototipo realizado para la consecución de una mayor potencialidad.

Consideramos que los aspectos mencionados en este epígrafe constituyen la más valiosa aportación de esta investigación al joven, pero dinámico, campo de la Informática.

APENDICE I

RESUMEN DE LOS SIMBOLOS UTILIZADOS DE LA NOTACION IVERSON (APL)

I.· OPERACIONES ARITMETICAS Y LOGICAS:

Aritméticas:		y <--- s op t (op: +, - ,*,...)
lógicas:	AND	y <--- s ∧ t
	OR	y <--- s ∨ t
	EXCLUSIVE OR	y <--- s ⊕ t

II. DESPLAZAMIENTOS:

Rotación izda. k posiciones	y <--- k↓s (si k=1,k puede suprimirse)
rotación dcha. k posiciones	y <--- k↑s
desplaza. izda. k posiciones	y <--- k⇃s (completando con ceros)
desplaza. dcha. k posiciones	y <--- k↾s (completando con ceros)

III. TRANFERENCIAS:

Pasar el contenido de s a d	d<--- s
indexación (contenido de la posición de memoria s a d)	d<--- (s)

IV. COMPARACIONES Y SALTOS:

comparar:	d : s
operaciones condicionadas:	a:b;(=) ---> c <--- s op t
salto incondicional:	---> (n)
salto condicional:	a:b;(=,<) ---> (n)

V. CATENACION:

(d,s)

oo OOO oo

A lo largo de la memoria:

A,B,C,D,E,H, y L son registros o vectores de 8 bits;
A es el acumulador y H y L registros de indexación o de direccionamiento indirecto. (H,L) o M, indica el contenido de memoria de la posición; página H, dirección baja L.
c,z,p y s son básculas "banderas" de 1 bit.

APENDICE II

ABREVIATURAS MAS UTILIZADAS

ALU	unidad aritmético-lógica
A/D	conversor de analógico a digital
CP	contador de programa
CPU	procesador central (unidades aritmético-lógica y de control).
D/A	conversor de digital a analógico.
E/S	entradas y(o) salidas.
LC	lenguaje de control
LE	lenguaje ensamblador
LM	lenguaje máquina
LPF	lenguaje para operar con funciones.
MSB	bit más significativo
RI	registro instrucción.

****

⊗	operación exclusive OR
c	biestable de rebose
z	biestable de cero (si el resultado es cero z=1)
s	biestable del bit más significativo
p	biestable de paridad

****

secc.	sección o epígrafe
fgra.	figura
c	capítulo
p	página
( )	referencia bibliográfica

****

APENDICE III

PROGRAMAS EN ENSAMBLADOR Y LENGUAJE MAQUINA DEL SISTEMA OPERATIVO.

En este apéndice se incluyen los programas del S. O. y los resultados obtenidos con el ensamblador cruzado. En los listados se puede observar la dirección de memoria en decimal y el octeto correspondiente a la instrucción en hexadecimal. Ademas se tiene la instrucción en ensamblador y en binario. La direcciónes de memoria se incluyen también en hexadecimal.
Al principio de cada página figura la tabla de direcciones (dirección simbólica- dirección absoluta), de los puntos de entrada tanto internos como externos.

```
PMO              0          0
SE4              6         35
SA12A            6        190
SA12B            6        201
XQTM             9         77
XQT              8          0
PM1              0          8
PM2              0         16
PM3              0         24
PM4              0         32
PM5              0         40
PM7              0         56
PM7B             0         64
PM7A             0         77
S1PMO            0         97
PMOA             0        100
S1PM1            0        108
PM1A             0        120
PM1B             0        130
S2PM2            0        134
PM2A             0        136
S1PM3            0        144
PM3A             0        148
S1PM5            0        158
SM1              0        169
SM2              0        185
SM3              0        191
SM4              0        199
SA12             0        205
SO94             0        224
SE14             0        226
SO95             0        232
SE13             0        234
SO96             0        240
SO97             0        248
SA6              0        250
```

```
PMO   DIR   0,0
SE4   EQU   35,6
SA12AEQU    190,6
SA12BEQU    201,6
XQTM  EQU   77,9
XQT   EQU   0,8
```

```
0     06          LOAD  A,*0          0000 0110     00 0
1     00                                0000 0000     01 0
2     73          OUT   S1            0111 0011     02 0
3     44          JUMP  S1PMO         0100 0100     03 0
4     61                                0110 0001     04 0
5     00                                0000 0000     05 0
                  **********
6     **                                **** ****     06 0
7     **                                **** ****     07 0
                  PM1   DIR   8,0
8     06          LOAD  A,*0          0000 0110     08 0
9     00                                0000 0000     09 0
10    73          OUT   S1            0111 0011     0A 0
11    44          JUMP  S1PM1         0100 0100     0B 0
12    6C                                0110 1100     0C 0
13    00                                0000 0000     0D 0
                  **********
14    **                                **** ****     0E 0
15    **                                **** ****     0F 0
                  PM2   DIR   16,0
16    06          LOAD  A,*1          0000 0110     10 0
```

```
17   01                                  0000 0001   11 0
18   73          OUT   S1               0111 0011   12 0
19   2E          LOAD  H,*8             0010 1110   13 0
20   08                                  0000 1000   14 0
21   44          JUMP  S2PM2            0100 0100   15 0
22   86                                  1000 0110   16 0
23   00                                  0000 0000   17 0
                 ********************
                 PM3   DIR   24,0
24   06          LOAD  A,*1             0000 0110   18 0
25   01                                  0000 0001   19 0
26   73          OUT   S1               0111 0011   1A 0
27   44          JUMP  S1PM3            0100 0100   1B 0
28   90                                  1001 0000   1C 0
29   00                                  0000 0000   1D 0
                 ********************
30   **                                  **** ****   1E 0
31   **                                  **** ****   1F 0
                 PM4   DIR   32,0
32   44          JUMP  XQT              0100 0100   20 0
33   00                                  0000 0000   21 0
34   08                                  0000 1000   22 0
                 ********************
35   **                                  **** ****   23 0
36   **                                  **** ****   24 0
37   **                                  **** ****   25 0
38   **                                  **** ****   26 0
39   **                                  **** ****   27 0
                 PM5   DIR   40,0
40   F0          LOAD  L,A              1111 0000   28 0
41   73          OUT   S1               0111 0011   29 0
42   C1          LOAD  A,B              1100 0001   2A 0
43   73          OUT   S1               0111 0011   2B 0
44   C2          LOAD  A,C              1100 0010   2C 0
45   44          JUMP  S1PM5            0100 0100   2D 0
46   9E                                  1001 1110   2E 0
47   00                                  0000 0000   2F 0
                 ********************
48   **                                  **** ****   30 0
49   **                                  **** ****   31 0
50   **                                  **** ****   32 0
51   **                                  **** ****   33 0
52   **                                  **** ****   34 0
53   **                                  **** ****   35 0
54   **                                  **** ****   36 0
55   **                                  **** ****   37 0
                 PM7   DIR   56,0
56   06          LOAD  A,*63            0000 0110   38 0
57   3F                                  0011 1111   39 0
58   2E          LOAD  H,*9             0010 1110   3A 0
59   09                                  0000 1001   3B 0
60   36          LOAD  L,*10            0011 0110   3C 0
61   0A                                  0000 1010   3D 0
62   26          LOAD  E,*0             0010 0110   3E 0
63   00                                  0000 0000   3F 0
64   FC          LOAD  (HL),E           1111 1100   40 0
65   30          INR   L                0011 0000   41 0
66   BE          COMP  A,L              1011 1110   42 0
67   40          JUMP  CC,PM7B          0100 0000   43 0
68   40                                  0100 0000   44 0
69   00                                  0000 0000   45 0
70   36          LOAD  L,*77            0011 0110   46 0
71   4D                                  0100 1101   47 0
```

```
  72    3E          LOAD (HL),*68      0011 1110    48 0
  73    44                             0100 0100    49 0
  74    36          LOAD L,*7          0011 0110    4A 0
  75    07                             0000 0111    4B 0
  76    FC          LOAD (HL),E        1111 1100    4C 0
  77    F4   PM7A   LOAD L,E           1111 0100    4D 0
  78    2E          LOAD H,*8          0010 1110    4E 0
  79    08                             0000 1000    4F 0
  80    E7          LOAD E,(HL)        1110 0111    50 0
  81    46          CALL SE4           0100 0110    51 0
  82    23                             0010 0011    52 0
  83    06                             0000 0110    53 0
  84    46          CALL XQTM          0100 0110    54 0
  85    4D                             0100 1101    55 0
  86    09                             0000 1001    56 0
  87    2E          LOAD H,*9          0010 1110    57 0
  88    09                             0000 1001    58 0
  89    36          LOAD L,*7          0011 0110    59 0
  90    07                             0000 0111    5A 0
  91    E7          LOAD E,(HL)        1110 0111    5B 0
  92    20          INR  E             0010 0000    5C 0
  93    FC          LOAD (HL),E        1111 1100    5D 0
  94    44          JUMP PM7A          0100 0100    5E 0
  95    4D                             0100 1101    5F 0
  96    00                             0000 0000    60 0
                    **********
  97    46   S1PMOCALL SM1             0100 0110    61 0
  98    A9                             1010 1001    62 0
  99    00                             0000 0000    63 0
 100    C7   PMOA LOAD A,(HL)          1100 0111    64 0
 101    73        OUT  S1              0111 0011    65 0
 102    46        CALL SM2             0100 0110    66 0
 103    B9                             1011 1001    67 0
 104    00                             0000 0000    68 0
 105    44        JUMP PMOA            0100 0100    69 0
 106    64                             0110 0100    6A 0
 107    00                             0000 0000    6B 0
 108    46   S1PM1CALL SM1             0100 0110    6C 0
 109    A9                             1010 1001    6D 0
 110    00                             0000 0000    6E 0
 111    C5        LOAD A,H             1100 0101    6F 0
 112    14        SUB  A,*8            0001 0100    70 0
 113    08                             0000 1000    71 0
 114    60        JUMP CS,PM1B         0110 0000    72 0
 115    82                             1000 0010    73 0
 116    00                             0000 0000    74 0
 117    06        LOAD A,*8            0000 0110    75 0
 118    08                             0000 1000    76 0
 119    73        OUT  S1              0111 0011    77 0
 120    C7   PM1A LOAD A,(HL)          1100 0111    78 0
 121    73        OUT  S1              0111 0011    79 0
 122    43        INP  E1              0100 0011    7A 0
 123    F8        LOAD (HL),A          1111 1000    7B 0
 124    46        CALL SM2             0100 0110    7C 0
 125    B9                             1011 1001    7D 0
 126    00                             0000 0000    7E 0
 127    44        JUMP PM1A            0100 0100    7F 0
 128    78                             0111 1000    80 0
 129    00                             0000 0000    81 0
 130    06   PM1B LOAD A,*31           0000 0110    82 0
 131    1F                             0001 1111    83 0
 132    73        OUT  S1              0111 0011    84 0
 133    00        HLT                  0000 0000    85 0
```

134	43	S2PM2I	NP	E1	0100 0011	86 0
135	F0		LOAD	L•A	1111 0000	87 0
136	C7	PM2A	LOAD	A•(HL)	1100 0111	88 0
137	73		OUT	S1	0111 0011	89 0
138	46		CALL	SM4	0100 0110	8A 0
139	C7				1100 0111	8B 0
140	00				0000 0000	8C 0
141	44		JUMP	PM2A	0100 0100	8D 0
142	88				1000 1000	8E 0
143	00				0000 0000	8F 0
144	2E	S1PM3L	OAD	H•*8	0010 1110	90 0
145	08				0000 1000	91 0
146	43		INP	E1	0100 0011	92 0
147	F0		LOAD	L•A	1111 0000	93 0
148	C7	PM3A	LOAD	A•(HL)	1100 0111	94 0
149	73		OUT	S1	0111 0011	95 0
150	43		INP	E1	0100 0011	96 0
151	F8		LOAD	(HL)•A	1111 1000	97 0
152	46		CALL	SM4	0100 0110	98 0
153	C7				1100 0111	99 0
154	00				0000 0000	9A 0
155	44		JUMP	PM3A	0100 0100	9B 0
156	94				1001 0100	9C 0
157	00				0000 0000	9D 0
158	73	S1PM5	OUT	S1	0111 0011	9E 0
159	C3		LOAD	A•D	1100 0011	9F 0
160	73		OUT	S1	0111 0011	A0 0
161	C4		LOAD	A•E	1100 0100	A1 0
162	73		OUT	S1	0111 0011	A2 0
163	C5		LOAD	A•H	1100 0101	A3 0
164	73		OUT	S1	0111 0011	A4 0
165	C6		LOAD	A•L	1100 0110	A5 0
166	73		OUT	S1	0111 0011	A6 0
167	00		HLT		0000 0000	A7 0
168	07		RET		0000 0111	A8 0
169	06	SM1	LOAD	A•*1	0000 0110	A9 0
170	01				0000 0001	AA 0
171	73		OUT	S1	0111 0011	AB 0
172	43		INP	E1	0100 0011	AC 0
173	F0		LOAD	L•A	1111 0000	AD 0
174	06		LOAD	A•*128	0000 0110	AE 0
175	80				1000 0000	AF 0
176	73		OUT	S1	0111 0011	B0 0
177	43		INP	E1	0100 0011	B1 0
178	E8		LOAD	H•A	1110 1000	B2 0
179	06		LOAD	A•*0	0000 0110	B3 0
180	00				0000 0000	B4 0
181	73		OUT	S1	0111 0011	B5 0
182	44		JUMP	SM3	0100 0100	B6 0
183	BF				1011 1111	B7 0
184	00				0000 0000	B8 0
185	30	SM2	INR	L	0011 0000	B9 0
186	0B		RET	ZC	0000 1011	BA 0
187	28		INR	H	0010 1000	BB 0
188	44		JUMP	SM3	0100 0100	BC 0
189	BF				1011 1111	BD 0
190	00				0000 0000	BE 0
191	06	SM3	LOAD	A•*19	0000 0110	BF 0
192	13				0001 0011	C0 0
193	95		SUB	A•H	1001 0101	C1 0
194	03		RET	CC	0000 0011	C2 0
195	06		LOAD	A•*15	0000 0110	C3 0
196	0F				0000 1111	C4 0

```
197  73        OUT   S1              0111 0011   C5 0
198  00        HLT                   0000 0000   C6 0
199  30   SM4  INR   L               0011 0000   C7 0
200  08        RET   ZC              0000 1011   C8 0
201  06        LOAD  A,*47           0000 0110   C9 0
202  2F                              0010 1111   CA 0
203  73        OUT   S1              0111 0011   CB 0
204  00        HLT                   0000 0000   CC 0
               **********
205  46   SA12 CALL  SA12A           0100 0110   CD 0
206  8E                              1011 1110   CE 0
207  06                              0000 0110   CF 0
208  1E        LOAD  D,*0            0001 1110   D0 0
209  00                              0000 0000   D1 0
210  44        JUMP  SA12B           0100 0100   D2 0
211  C9                              1100 1001   D3 0
212  06                              0000 0110   D4 0
213  **                              **** ****   D5 0
214  **                              **** ****   D6 0
215  **                              **** ****   D7 0
216  **                              **** ****   D8 0
217  **                              **** ****   D9 0
218  **                              **** ****   DA 0
219  **                              **** ****   DB 0
220  **                              **** ****   DC 0
221  **                              **** ****   DD 0
222  **                              **** ****   DE 0
223  **                              **** ****   DF 0
               S094 DIR   224,0
224  41        INP   E0              0100 0001   E0 0
225  07        RET                   0000 0111   E1 0
226  06   SE14 LOAD  A,*241          0000 0110   E2 0
227  F1                              1111 0001   E3 0
228  73        OUT   S1              0111 0011   E4 0
229  00        HLT                   0000 0000   E5 0
230  07        RET                   0000 0111   E6 0
231  **                              **** ****   E7 0
               S095 DIR   232,0
232  43        INP   E1              0100 0011   E8 0
233  07        RET                   0000 0111   E9 0
234  06   SE13 LOAD  A,*242          0000 0110   EA 0
235  F2                              1111 0010   EB 0
236  73        OUT   S1              0111 0011   EC 0
237  00        HLT                   0000 0000   ED 0
238  07        RET                   0000 0111   EE 0
239  **                              **** ****   EF 0
               S096 DIR   240,0
240  71        OUT   S0              0111 0001   F0 0
241  07        RET                   0000 0111   F1 0
242  **                              **** ****   F2 0
243  **                              **** ****   F3 0
244  **                              **** ****   F4 0
245  **                              **** ****   F5 0
246  **                              **** ****   F6 0
247  **                              **** ****   F7 0
               S097 DIR   248,0
248  73        OUT   S1              0111 0011   F8 0
249  07        RET                   0000 0111   F9 0
250  C4   SA6  LOAD  A,E             1100 0100   FA 0
251  83        ADD   D               1000 0011   FB 0
252  E8        LOAD  H,A             1110 1000   FC 0
253  C7        LOAD  A,(HL)          1100 0111   FD 0
254  07        RET                   0000 0111   FE 0
```

S101	1	0
SE1	5	211
SE2	6	0
SE3	7	114
SE5	7	172
SE7	4	162
SE10	6	80
SE17	7	234
SA2	6	102
SA3	3	220
SA3E	3	231
SA4	2	246
SA5	2	65
SA10	6	130
SA12	0	205
SA20	7	80
SA13	6	211
SS104	3	175
S106F	2	39
S101A	1	9
S101Z	1	28
S01ZA	1	38
SE15	1	57
S102	1	64
S102A	1	73
S102Z	1	89
S02ZA	1	99
S103	1	120
S103A	1	134
S104	1	152
S104A	1	169
S105	1	184
S105A	1	187
S105B	1	192
S105D	1	213
S106	1	224
S106A	1	227
S106B	1	252

```
     S101 DIR  0,1
     SE1  EQU  211,5
     SE2  EQU  0,6
     SE3  EQU  114,7
     SE5  EQU  172,7
     SE7  EQU  162,4
     SE10 EQU  80,6
     SE17 EQU  234,7
     SA2  EQU  102,6
     SA3  EQU  220,3
     SA3E EQU  231,7
     SA4  EQU  246,2
     SA5  EQU  65,2
     SA10 EQU  130,6
     SA12 EQU  205,0
     SA20 EQU  80,7
     SA13 EQU  211,6
     SS104EQU  175,7
     S106FEQU  39,2
0    46        CALL SA3        0100 0110    00 1
1    DC                        1101 1100    01 1
2    03                        0000 0011    02 1
3    46        CALL SA2        0100 0110    03 1
4    66                        0110 0110    04 1
5    06                        0000 0110    05 1
```

```
 6   1E        LOAD D,*0        0001 1110   06 1
 7   00                         0000 0000   07 1
 8   F3        LOAD L,D         1111 0011   08 1
 9   46   S101ACALL SA4         0100 0110   09 1
10   F6                         1111 0110   0A 1
11   02                         0000 0010   0B 1
12   46        CALL SE3         0100 0110   0C 1
13   72                         0111 0010   0D 1
14   07                         0000 0111   0E 1
15   60        JUMP CS,S101Z    0110 0000   0F 1
16   1C                         0001 1100   10 1
17   01                         0000 0001   11 1
18   F8        LOAD (HL),A      1111 1000   12 1
19   46        CALL SE7         0100 0110   13 1
20   A2                         1010 0010   14 1
21   04                         0000 0100   15 1
22   40        JUMP CC,S101A    0100 0000   16 1
23   09                         0000 1001   17 1
24   01                         0000 0001   18 1
25   44        JUMP SE10        0100 0100   19 1
26   50                         0101 0000   1A 1
27   06                         0000 0110   1B 1
28   36   S101ZLOAD L,*10       0011 0110   1C 1
29   0A                         0000 1010   1D 1
30   2E        LOAD H,*9        0010 1110   1E 1
31   09                         0000 1001   1F 1
32   CF        LOAD B,(HL)      1100 1111   20 1
33   08        INR  B           0000 1000   21 1
34   F9        LOAD (HL),B      1111 1001   22 1
35   1E        LOAD D,*0        0001 1110   23 1
36   00                         0000 0000   24 1
37   F3        LOAD L,D         1111 0011   25 1
38   46   S01ZACALL SA4         0100 0110   26 1
39   F6                         1111 0110   27 1
40   02                         0000 0010   28 1
41   46        CALL SE3         0100 0110   29 1
42   72                         0111 0010   2A 1
43   07                         0000 0111   2B 1
44   46        CALL SE17        0100 0110   2C 1
45   EA                         1110 1010   2D 1
46   07                         0000 0111   2E 1
47   F8        LOAD (HL),A      1111 1000   2F 1
48   46        CALL SE7         0100 0110   30 1
49   A2                         1010 0010   31 1
50   04                         0000 0100   32 1
51   40        JUMP CC,S01ZA    0100 0000   33 1
52   26                         0010 0110   34 1
53   01                         0000 0001   35 1
54   44        JUMP SE10        0100 0100   36 1
55   50                         0101 0000   37 1
56   06                         0000 0110   38 1
57   2E   SE15 LOAD H,*9        0010 1110   39 1
58   09                         0000 1001   3A 1
59   44        JUMP SA3E        0100 0100   3B 1
60   E7                         1110 0111   3C 1
61   03                         0000 0011   3D 1
62   **                         **** ****   3E 1
63   **                         **** ****   3F 1
          S102 ORG  64,1
64   46        CALL SA3         0100 0110   40 1
65   DC                         1101 1100   41 1
66   03                         0000 0011   42 1
67   46        CALL SA2         0100 0110   43 1
```

```
 68   66                                  0110 0110   44 1
 69   06                                  0000 0110   45 1
 70   1E          LOAD D,*0               0001 1110   46 1
 71   00                                  0000 0000   47 1
 72   F3          LOAD L,D                1111 0011   48 1
 73   46    S102ACALL SA4                 0100 0110   49 1
 74   F6                                  1111 0110   4A 1
 75   02                                  0000 0010   4B 1
 76   46          CALL SE5                0100 0110   4C 1
 77   AC                                  1010 1100   4D 1
 78   07                                  0000 0111   4E 1
 79   60          JUMP CS,S102Z           0110 0000   4F 1
 80   59                                  0101 1001   50 1
 81   01                                  0000 0001   51 1
 82   F8          LOAD (HL),A             1111 1000   52 1
 83   40          JUMP CC,S102A           0100 0000   53 1
 84   49                                  0100 1001   54 1
 85   01                                  0000 0001   55 1
 86   44          JUMP SE10               0100 0100   56 1
 87   50                                  0101 0000   57 1
 88   06                                  0000 0110   58 1
 89   36    S102ZLOAD L,*10               0011 0110   59 1
 90   0A                                  0000 1010   5A 1
 91   2E          LOAD H,*9               0010 1110   5B 1
 92   09                                  0000 1001   5C 1
 93   CF          LOAD B,(HL)             1100 1111   5D 1
 94   08          INR  B                  0000 1000   5E 1
 95   F9          LOAD (HL),B             1111 1001   5F 1
 96   1E          LOAD D,*0               0001 1110   60 1
 97   00                                  0000 0000   61 1
 98   F3          LOAD L,D                1111 0011   62 1
 99   46    S02ZACALL SA4                 0100 0110   63 1
100   F6                                  1111 0110   64 1
101   02                                  0000 0010   65 1
102   46          CALL SE5                0100 0110   66 1
103   AC                                  1010 1100   67 1
104   07                                  0000 0111   68 1
105   46          CALL SE17               0100 0110   69 1
106   EA                                  1110 1010   6A 1
107   07                                  0000 0111   6B 1
108   F8          LOAD (HL),A             1111 1000   6C 1
109   46          CALL SE7                0100 0110   6D 1
110   A2                                  1010 0010   6E 1
111   04                                  0000 0100   6F 1
112   40          JUMP CC,S02ZA           0100 0000   70 1
113   63                                  0110 0011   71 1
114   01                                  0000 0001   72 1
115   44          JUMP SE10               0100 0100   73 1
116   50                                  0101 0000   74 1
117   06                                  0000 0110   75 1
118   **                                  **** ****   76 1
119   **                                  **** ****   77 1
            S103 DIR  120,1
120   46          CALL SA3                0100 0110   78 1
121   DC                                  1101 1100   79 1
122   03                                  0000 0011   7A 1
123   46          CALL SA5                0100 0110   7B 1
124   41                                  0100 0001   7C 1
125   02                                  0000 0010   7D 1
126   D1          LOAD C,B                1101 0001   7E 1
127   46          CALL SE3                0100 0110   7F 1
128   72                                  0111 0010   80 1
129   07                                  0000 0111   81 1
```

```
130   F8          LOAD (HL)•A          1111 1000   82 1
131   1E          LOAD D•*0            0001 1110   83 1
132   00                               0000 0000   84 1
133   E3          LOAD E•D             1110 0011   85 1
134   46    S103ACALL SA4              0100 0110   86 1
135   F6                               1111 0110   87 1
136   02                               0000 0010   88 1
137   C8          LOAD B•A             1100 1000   89 1
138   46          CALL SE1             0100 0110   8A 1
139   03                               1101 0011   8B 1
140   05                               0000 0101   8C 1
141   F8          LOAD (HL)•A          1111 1000   8D 1
142   46          CALL SE7             0100 0110   8E 1
143   A2                               1010 0010   8F 1
144   04                               0000 0100   90 1
145   40          JUMP CC•S103A        0100 0000   91 1
146   86                               1000 0110   92 1
147   01                               0000 0001   93 1
148   44          JUMP SE10            0100 0100   94 1
149   50                               0101 0000   95 1
150   06                               0000 0110   96 1
151   **                               **** ****   97 1
            S104 DIR  152•1
152   46          CALL SA3             0100 0110   98 1
153   DC                               1101 1100   99 1
154   03                               0000 0011   9A 1
155   46          CALL SA13            0100 0110   9B 1
156   D3                               1101 0011   9C 1
157   06                               0000 0110   9D 1
158   46          CALL SA5             0100 0110   9E 1
159   41                               0100 0001   9F 1
160   02                               0000 0010   A0 1
161   D1          LOAD C•B             1101 0001   A1 1
162   46          CALL SE5             0100 0110   A2 1
163   AC                               1010 1100   A3 1
164   07                               0000 0111   A4 1
165   F8          LOAD (HL)•A          1111 1000   A5 1
166   1E          LOAD D•*0            0001 1110   A6 1
167   00                               0000 0000   A7 1
168   E3          LOAD E•D             1110 0011   A8 1
169   46    S104ACALL SA4              0100 0110   A9 1
170   F6                               1111 0110   AA 1
171   02                               0000 0010   AB 1
172   C8          LOAD B•A             1100 1000   AC 1
173   46          CALL SE2             0100 0110   AD 1
174   00                               0000 0000   AE 1
175   06                               0000 0110   AF 1
176   F8          LOAD (HL)•A          1111 1000   B0 1
177   44          JUMP SS104           0100 0100   B1 1
178   AF                               1010 1111   B2 1
179   03                               0000 0011   B3 1
180   **                               **** ****   B4 1
181   **                               **** ****   B5 1
182   **                               **** ****   B6 1
183   **                               **** ****   B7 1
            S105 DIR  184•1
184   46          CALL SA3             0100 0110   B8 1
185   DC                               1101 1100   B9 1
186   03                               0000 0011   BA 1
187   1E    S105ALOAD D•*0             0001 1110   BB 1
188   00                               0000 0000   BC 1
189   F3          LOAD L•D             1111 0011   BD 1
190   CB          LOAD B•D             1100 1011   BE 1
```

```
191  D3          LOAD C,D                1101 0011   BF 1
192  C1   S105BLOAD A,B                  1100 0001   C0 1
193  46          CALL SE3                0100 0110   C1 1
194  72                                  0111 0010   C2 1
195  07                                  0000 0111   C3 1
196  60          JUMP CS,S105D           0110 0000   C4 1
197  D5                                  1101 0101   C5 1
198  01                                  0000 0001   C6 1
199  C8          LOAD B,A                1100 1000   C7 1
200  46          CALL SA4                0100 0110   C8 1
201  F6                                  1111 0110   C9 1
202  02                                  0000 0010   CA 1
203  F9          LOAD (HL),B             1111 1001   CB 1
204  46          CALL SE7                0100 0110   CC 1
205  A2                                  1010 0010   CD 1
206  04                                  0000 0100   CE 1
207  40          JUMP CC,S1058           0100 0000   CF 1
208  C0                                  1100 0000   D0 1
209  01                                  0000 0001   D1 1
210  44          JUMP SE10               0100 0100   D2 1
211  50                                  0101 0000   D3 1
212  06                                  0000 0110   D4 1
213  16   S105DLOAD C,*1                 0001 0110   D5 1
214  01                                  0000 0001   D6 1
215  46          CALL SA10               0100 0110   D7 1
216  82                                  1000 0010   D8 1
217  06                                  0000 0110   D9 1
218  44          JUMP S105A              0100 0100   DA 1
219  BB                                  1011 1011   DB 1
220  01                                  0000 0001   DC 1
221  **                                  **** ****   DD 1
222  **                                  **** ****   DE 1
223  **                                  **** ****   DF 1
               S105 DIR  224,1
224  46          CALL SA3                0100 0110   E0 1
225  DC                                  1101 1100   E1 1
226  03                                  0000 0011   E2 1
227  46   S106ACALL SA20                 0100 0110   E3 1
228  50                                  0101 0000   E4 1
229  07                                  0000 0111   E5 1
230  CC          LOAD B,E                1100 1100   E6 1
231  26          LOAD E,*10              0010 0110   E7 1
232  0A                                  0000 1010   E8 1
233  46          CALL SA12               0100 0110   E9 1
234  CD                                  1100 1101   EA 1
235  00                                  0000 0000   EB 1
236  E1          LOAD E,B                1110 0001   EC 1
237  1E          LOAD D,*0               0001 1110   ED 1
238  00                                  0000 0000   EE 1
239  F3          LOAD L,D                1111 0011   EF 1
240  46          CALL SA4                0100 0110   F0 1
241  F6                                  1111 0110   F1 1
242  02                                  0000 0010   F2 1
243  46          CALL SE5                0100 0110   F3 1
244  AC                                  1010 1100   F4 1
245  07                                  0000 0111   F5 1
246  60          JUMP CS,S106F           0110 0000   F6 1
247  27                                  0010 0111   F7 1
248  02                                  0000 0010   F8 1
249  CA          LOAD B,C                1100 1010   F9 1
250  F8          LOAD (HL),A             1111 1000   FA 1
251  30          INR  L                  0011 0000   FB 1
252  46   S106BCALL SA4                  0100 0110   FC 1
```

253	F6		1111 0110	FD	1
254	02		0000 0010	FE	1
255	C0	NOP	1100 0000	FF	1

```
SS106          2          0
SE6            6         57
SE7            4        162
SE10           6         80
SA3            3        220
SA6            0        250
SE5            7        172
SE17           7        234
SA10           6        130
SA12           0        205
SA12A          6        190
SA12B          6        201
SA20           7         80
SA21           7         98
S106A          1        227
S106B          1        252
S106F          2         39
S107           2         48
S107A          2         51
SA5            2         65
S108           2         72
S108A          2         78
SSA5           2         92
S109           2         96
S109A          2         99
S110           2        112
S110A          2        115
S111           2        128
S111A          2        136
S111B          2        144
S112C          2        150
S112           2        160
S112A          2        168
S112B          2        176
SA7            2        185
S121           2        192
SS123          2        201
S122           2        208
S122A          2        219
S123           2        240
SA4            2        246
```

```
SS106DIR  0,2
SE6   EQU  57,6
SE7   EQU  162,4
SE10  EQU  80,6
SA3   EQU  220,3
SA6   EQU  250,0
SE5   EQU  172,7
SE17  EQU  234,7
SA10  EQU  130,6
SA12  EQU  205,0
SA12AEQU  190,6
SA12BEQU  201,6
SA20  EQU  80,7
SA21  EQU  98,7
S106AEQU  227,1
S106BEQU  252,1
```

```
0     46          CALL SE17        0100 0110     00 2
1     EA                           1110 1010     01 2
2     07                           0000 0111     02 2
3     E8          LOAD H,A         1110 1000     03 2
4     C1          LOAD A,B         1100 0001     04 2
5     CA          LOAD B,C         1100 1010     05 2
```

```
   6    46          CALL SE17          0100 0110    06 2
   7    EA                             1110 1010    07 2
   8    07                             0000 0111    08 2
   9    D0          LOAD C,A           1101 0000    09 2
  10    C5          LOAD A,H           1100 0101    0A 2
  11    46          CALL SE5           0100 0110    0B 2
  12    AC                             1010 1100    0C 2
  13    07                             0000 0111    0D 2
  14    2E          LOAD H,*10         0010 1110    0E 2
  15    0A                             0000 1010    0F 2
  16    F8          LOAD (HL),A        1111 1000    10 2
  17    46          CALL SE7           0100 0110    11 2
  18    A2                             1010 0010    12 2
  19    04                             0000 0100    13 2
  20    40          JUMP CC,S106B      0100 0000    14 2
  21    FC                             1111 1100    15 2
  22    01                             0000 0001    16 2
  23    19          DCR  D             0001 1001    17 2
  24    C3          LOAD A,D           1100 0011    18 2
  25    04          ADD  A,*10         0000 0100    19 2
  26    0A                             0000 1010    1A 2
  27    E8          LOAD H,A           1110 1000    1B 2
  28    36          LOAD L,*254        0011 0110    1C 2
  29    FE                             1111 1110    1D 2
  30    D7          LOAD C,(HL)        1101 0111    1E 2
  31    C1          LOAD A,B           1100 0001    1F 2
  32    46          CALL SE5           0100 0110    20 2
  33    AC                             1010 1100    21 2
  34    07                             0000 0111    22 2
  35    F8          LOAD (HL),A        1111 1000    23 2
  36    40          JUMP CC,SE10       0100 0000    24 2
  37    50                             0101 0000    25 2
  38    06                             0000 0110    26 2
  39    16    S106FLOAD C,*1           0001 0110    27 2
  40    01                             0000 0001    28 2
  41    46          CALL SA10          0100 0110    29 2
  42    82                             1000 0010    2A 2
  43    06                             0000 0110    2B 2
  44    44          JUMP S106A         0100 0100    2C 2
  45    E3                             1110 0011    2D 2
  46    01                             0000 0001    2E 2
  47    **                             **** ****    2F 2
              S107 DIR  48,2
  48    46          CALL SA3           0100 0110    30 2
  49    DC                             1101 1100    31 2
  50    03                             0000 0011    32 2
  51    46    S107ACALL SA21           0100 0110    33 2
  52    62                             0110 0010    34 2
  53    07                             0000 0111    35 2
  54    EC          LOAD H,E           1110 1100    36 2
  55    36          LOAD L,*0          0011 0110    37 2
  56    00                             0000 0000    38 2
  57    F9          LOAD (HL),B        1111 1001    39 2
  58    11          DCR  C             0001 0001    3A 2
  59    48          JUMP ZC,S107A      0100 1000    3B 2
  60    33                             0011 0011    3C 2
  61    02                             0000 0010    3D 2
  62    44          JUMP SE10          0100 0100    3E 2
  63    50                             0101 0000    3F 2
  64    06                             0000 0110    40 2
  65    F4    SA5   LOAD L,E           1111 0100    41 2
  66    2E          LOAD H,*9          0010 1110    42 2
  67    09                             0000 1001    43 2
```

```
  68   CF          LOAD B,(HL)          1100 1111    44 2
  69   44          JUMP SSA5            0100 0100    45 2
  70   5C                               0101 1100    46 2
  71   02                               0000 0010    47 2
                S108 DIR  72,2
  72   46          CALL SA3             0100 0110    48 2
  73   DC                               1101 1100    49 2
  74   03                               0000 0011    4A 2
  75   46          CALL SA12A           0100 0110    4B 2
  76   BE                               1011 1110    4C 2
  77   06                               0000 0110    4D 2
  78   46     S108ACALL SA12B           0100 0110    4E 2
  79   C9                               1100 1001    4F 2
  80   06                               0000 0110    50 2
  81   46          CALL SA12A           0100 0110    51 2
  82   BE                               1011 1110    52 2
  83   06                               0000 0110    53 2
  84   F8          LOAD (HL),D          1111 1011    54 2
  85   11          DCR  C               0001 0001    55 2
  86   48          JUMP ZC,S108A        0100 1000    56 2
  87   4E                               0100 1110    57 2
  88   02                               0000 0010    58 2
  89   44          JUMP SE10            0100 0100    59 2
  90   50                               0101 0000    5A 2
  91   06                               0000 0110    5B 2
  92   36     SSA5 LOAD L,*10           0011 0110    5C 2
  93   0A                               0000 1010    5D 2
  94   C7          LOAD A,(HL)          1100 0111    5E 2
  95   07          RET                  0000 0111    5F 2
                S109 DIR  96,2
  96   46          CALL SA3             0100 0110    60 2
  97   DC                               1101 1100    61 2
  98   03                               0000 0011    62 2
  99   46     S109ACALL SA21            0100 0110    63 2
 100   62                               0110 0010    64 2
 101   07                               0000 0111    65 2
 102   11          DCR  C               0001 0001    66 2
 103   48          JUMP ZC,S109A        0100 1000    67 2
 104   63                               0110 0011    68 2
 105   02                               0000 0010    69 2
 106   44          JUMP SE10            0100 0100    6A 2
 107   50                               0101 0000    6B 2
 108   06                               0000 0110    6C 2
 109   **                               **** ****    6D 2
 110   **                               **** ****    6E 2
 111   **                               **** ****    6F 2
                S110 DIR  112,2
 112   46          CALL SA3             0100 0110    70 2
 113   DC                               1101 1100    71 2
 114   03                               0000 0011    72 2
 115   46     S110ACALL SA12            0100 0110    73 2
 116   CD                               1100 1101    74 2
 117   00                               0000 0000    75 2
 118   11          DCR  C               0001 0001    76 2
 119   48          JUMP ZC,S110A        0100 1000    77 2
 120   73                               0111 0011    78 2
 121   02                               0000 0010    79 2
 122   44          JUMP SE10            0100 0100    7A 2
 123   50                               0101 0000    7B 2
 124   06                               0000 0110    7C 2
 125   **                               **** ****    7D 2
 126   **                               **** ****    7E 2
 127   **                               **** ****    7F 2
```

```
                        S111 DIR  128,2
128     46              S111ACALL SA3       0100 0110   80 2
129     DC                                  1101 1100   81 2
130     03                                  0000 0011   82 2
131     16              S111 LOAD C,*127    0001 0110   83 2
132     7F                                  0111 1111   84 2
133     1E              S111 LOAD D,*0      0001 1110   85 2
134     00                                  0000 0000   86 2
135     F3              S111 LOAD L,D       1111 0011   87 2
136     46              S111ACALL SA6       0100 0110   88 2
137     FA                                  1111 1010   89 2
138     00                                  0000 0000   8A 2
139     BA              S111 COMP A,C       1011 1010   8B 2
140     40              S111 JUMP CC,S111B  0100 0000   8C 2
141     90                                  1001 0000   8D 2
142     02                                  0000 0010   8E 2
143     D0              S111 LOAD C,A       1101 0000   8F 2
144     46              S111BCALL SE7       0100 0110   90 2
145     A2                                  1010 0010   91 2
146     04                                  0000 0100   92 2
147     40              S111 JUMP CC,S111A  0100 0000   93 2
148     88                                  1000 1000   94 2
149     02                                  0000 0010   95 2
150     E2              S112CLOAD E,C       1110 0010   96 2
151     46              S112 CALL SE10      0100 0110   97 2
152     50                                  0101 0000   98 2
153     06                                  0000 0110   99 2
154     C4              S112 LOAD A,E       1100 0100   9A 2
155     07              S112 RET            0000 0111   9B 2
156     **                                  **** ****   9C 2
157     **                                  **** ****   9D 2
158     **                                  **** ****   9E 2
159     **                                  **** ****   9F 2
                        S112 DIR  160,2
160     46              S112ACALL SA3       0100 0110   A0 2
161     DC                                  1101 1100   A1 2
162     03                                  0000 0011   A2 2
163     16              S112 LOAD C,*128    0001 0110   A3 2
164     80                                  1000 0000   A4 2
165     1E              S112 LOAD D,*0      0001 1110   A5 2
166     00                                  0000 0000   A6 2
167     F3              S112 LOAD L,D       1111 0011   A7 2
168     46              S112ACALL SA6       0100 0110   A8 2
169     FA                                  1111 1010   A9 2
170     00                                  0000 0000   AA 2
171     BA              S112 COMP A,C       1011 1010   AB 2
172     60              S112 JUMP CS,S112B  0110 0000   AC 2
173     B0                                  1011 0000   AD 2
174     02                                  0000 0010   AE 2
175     D0              S112 LOAD C,A       1101 0000   AF 2
176     46              S112BCALL SE7       0100 0110   B0 2
177     A2                                  1010 0010   B1 2
178     04                                  0000 0100   B2 2
179     40              S112 JUMP CC,S112A  0100 0000   B3 2
180     A8                                  1010 1000   B4 2
181     02                                  0000 0010   B5 2
182     44              S112 JUMP S112C     0100 0100   B6 2
183     96                                  1001 0110   B7 2
184     02                                  0000 0010   B8 2
185     46              SA7  CALL SE6       0100 0110   B9 2
186     39                                  0011 1001   BA 2
187     06                                  0000 0110   BB 2
188     2E              S112 LOAD H,*8      0010 1110   BC 2
```

```
189   08                                  0000 1000   BD 2
190   F7          LOAD L,(HL)             1111 0111   BE 2
191   07          RET                     0000 0111   BF 2
              S121 DIR  192,2
192   46          CALL SA3                0100 0110   C0 2
193   DC                                  1101 1100   C1 2
194   03                                  0000 0011   C2 2
195   46          CALL SA20               0100 0110   C3 2
196   50                                  0101 0000   C4 2
197   07                                  0000 0111   C5 2
198   44          JUMP SE10               0100 0100   C6 2
199   50                                  0101 0000   C7 2
200   06                                  0000 0110   C8 2
201   46    SS123 CALL SA10               0100 0110   C9 2
202   82                                  1000 0010   CA 2
203   06                                  0000 0110   CB 2
204   44          JUMP SE10               0100 0100   CC 2
205   50                                  0101 0000   CD 2
206   06                                  0000 0110   CE 2
207   **                                  **** ****   CF 2
              S122 DIR  208,2
208   46          CALL SA3                0100 0110   D0 2
209   DC                                  1101 1100   D1 2
210   03                                  0000 0011   D2 2
211   46          CALL SA5                0100 0110   D3 2
212   41                                  0100 0001   D4 2
213   02                                  0000 0010   D5 2
214   F4          LOAD L,E                1111 0100   D6 2
215   F8          LOAD (HL),A             1111 1000   D7 2
216   1E          LOAD D,*0               0001 1110   D8 2
217   00                                  0000 0000   D9 2
218   F3          LOAD L,D                1111 0011   DA 2
219   46    S122A CALL SA4                0100 0110   DB 2
220   F6                                  1111 0110   DC 2
221   02                                  0000 0010   DD 2
222   C8          LOAD B,A                1100 1000   DE 2
223   46          CALL SA6                0100 0110   DF 2
224   FA                                  1111 1010   E0 2
225   00                                  0000 0000   E1 2
226   F9          LOAD (HL),B             1111 1001   E2 2
227   46          CALL SE7                0100 0110   E3 2
228   A2                                  1010 0010   E4 2
229   04                                  0000 0100   E5 2
230   40          JUMP CC,S122A           0100 0000   E6 2
231   DB                                  1101 1011   E7 2
232   02                                  0000 0010   E8 2
233   44          JUMP SE10               0100 0100   E9 2
234   50                                  0101 0000   EA 2
235   06                                  0000 0110   EB 2
236   **                                  **** ****   EC 2
237   **                                  **** ****   ED 2
238   **                                  **** ****   EE 2
239   **                                  **** ****   EF 2
              S123 DIR  240,2
240   46          CALL SA3                0100 0110   F0 2
241   DC                                  1101 1100   F1 2
242   03                                  0000 0011   F2 2
243   44          JUMP SS123              0100 0100   F3 2
244   C9                                  1100 1001   F4 2
245   02                                  0000 0010   F5 2
246   C4    SA4   LOAD A,E                1100 0100   F6 2
247   83          ADD  A,D                1000 0011   F7 2
248   E8          LOAD H,A                1110 1000   F8 2
```

249	D7	LOAD C,(HL)	1101 0111	F9 2
250	C6	LOAD A,*10	0000 0110	FA 2
251	0A		0000 1010	FB 2
252	83	ADD A,D	1000 0011	FC 2
253	E8	LOAD H,A	1110 1000	FD 2
254	C7	LOAD A,(HL)	1100 0111	FE 2
255	07	RET	0000 0111	FF 2

S124	3	0
SE6	6	57
SE7	4	162
SE9	6	68
SE10	6	80
SE15	1	57
SA6	0	250
SA11	6	159
SA14	7	7
SA15	7	24
SA16	7	36
SA17	7	51
SA18	7	68
S104A	1	169
S125	3	16
S125A	3	23
S126	3	40
S127	3	48
S130	3	56
S130E	3	58
S132	3	72
S131	3	80
S133	3	88
S134	3	96
S135	3	104
S136	3	112
S137	3	120
S150	3	128
S150E	3	131
SS120	3	146
S151	3	152
S152	3	160
S152E	3	163
SS104	3	175
S153	3	184
SA1	3	191
SA1A	3	194
SA1B	3	197
SA3	3	220
SA3E	3	231
S120	3	248

```
S124 DIR  0,3
SE6  EQU  57,6
SE7  EQU  162,4
SE9  EQU  68,6
SE10 EQU  80,6
SE15 EQU  57,1
SA6  EQU  250,0
SA11 EQU  159,6
SA14 EQU  7,7
SA15 EQU  24,7
SA16 EQU  36,7
SA17 EQU  51,7
SA18 EQU  68,7
S104AEQU  169,1
```

```
0   46        CALL SA3        0100 0110   00 3
1   DC                        1101 1100   01 3
2   03                        0000 0011   02 3
3   46        CALL SA11       0100 0110   03 3
4   9F                        1001 1111   04 3
5   06                        0000 0110   05 3
6   44        JUMP SE10       0100 0100   06 3
7   50                        0101 0000   07 3
```

```
 8    06                                  0000 0110    08 3
 9    **                                  **** ****    09 3
10    **                                  **** ****    0A 3
11    **                                  **** ****    0B 3
12    **                                  **** ****    0C 3
13    **                                  **** ****    0D 3
14    **                                  **** ****    0E 3
15    **                                  **** ****    0F 3
                S125 DIR  16,3
16    46          CALL SA3                0100 0110    10 3
17    DC                                  1101 1100    11 3
18    03                                  0000 0011    12 3
19    D0          LOAD C,A                1101 0000    13 3
20    1E          LOAD D,*0               0001 1110    14 3
21    00                                  0000 0000    15 3
22    F3          LOAD L,D                1111 0011    16 3
23    C4     S125ALOAD A,E                1100 0100    17 3
24    83          ADD  D                  1000 0011    18 3
25    E8          LOAD H,A                1110 1000    19 3
26    FA          LOAD (HL),C             1111 1010    1A 3
27    46          CALL SE7                0100 0110    1B 3
28    A2                                  1010 0010    1C 3
29    04                                  0000 0100    1D 3
30    40          JUMP CC,S125A           0100 0000    1E 3
31    17                                  0001 0111    1F 3
32    03                                  0000 0011    20 3
33    44          JUMP SE10               0100 0100    21 3
34    50                                  0101 0000    22 3
35    06                                  0000 0110    23 3
36    **                                  **** ****    24 3
37    **                                  **** ****    25 3
38    **                                  **** ****    26 3
39    **                                  **** ****    27 3
                S126 DIR  40,3
40    46          CALL SE15               0100 0110    28 3
41    39                                  0011 1001    29 3
42    01                                  0000 0001    2A 3
43    F2          LOAD L,C                1111 0010    2B 3
44    EC          LOAD H,E                1110 1100    2C 3
45    C7          LOAD A,(HL)             1100 0111    2D 3
46    07          RET                     0000 0111    2E 3
47    **                                  **** ****    2F 3
                S127 DIR  48,3
48    46          CALL SE15               0100 0110    30 3
49    39                                  0011 1001    31 3
50    01                                  0000 0001    32 3
51    F2          LOAD L,C                1111 0010    33 3
52    EC          LOAD H,E                1110 1100    34 3
53    F8          LOAD (HL),A             1111 1000    35 3
54    07          RET                     0000 0111    36 3
55    **                                  **** ****    37 3
                S130 DIR  56,3
56    1E          LOAD D,*8               0001 1110    38 3
57    08                                  0000 1000    39 3
58    46     S130CALL SE6                 0100 0110    3A 3
59    39                                  0011 1001    3B 3
60    06                                  0000 0110    3C 3
61    E6          LOAD E,L                1110 0110    3D 3
62    2E          LOAD H,*9               0010 1110    3E 3
63    09                                  0000 1001    3F 3
64    F3          LOAD L,D                1111 0011    40 3
65    FC          LOAD (HL),E             1111 1100    41 3
66    07          RET                     0000 0111    42 3
```

```
                    ********************
 67      **                                 **** ****      43 3
 68      **                                 **** ****      44 3
 69      **                                 **** ****      45 3
 70      **                                 **** ****      46 3
 71      **                                 **** ****      47 3
                    S132 DIR  72,3
 72      1E              LOAD D,*9          0001 1110      48 3
 73      09                                 0000 1001      49 3
 74      44              JUMP S130E         0100 0100      4A 3
 75      3A                                 0011 1010      4B 3
 76      03                                 0000 0011      4C 3
 77      **                                 **** ****      4D 3
 78      **                                 **** ****      4E 3
 79      **                                 **** ****      4F 3
                    S131 DIR  80,3
 80      36              LOAD L,*8          0011 0110      50 3
 81      08                                 0000 1000      51 3
 82      2E              LOAD H,*9          0010 1110      52 3
 83      09                                 0000 1001      53 3
 84      FA              LOAD (HL),C        1111 1010      54 3
 85      07              RET                0000 0111      55 3
 86      **                                 **** ****      56 3
 87      **                                 **** ****      57 3
                    S133 DIR  88,3
 88      36              LOAD L,*9          0011 0110      58 3
 89      09                                 0000 1001      59 3
 90      EE              LOAD H,L           1110 1110      5A 3
 91      FA              LOAD (HL),C        1111 1010      5B 3
 92      07              RET                0000 0111      5C 3
 93      **                                 **** ****      5D 3
 94      **                                 **** ****      5E 3
 95      **                                 **** ****      5F 3
                    S134 DIR  96,3
 96      2E              LOAD H,*9          0010 1110      60 3
 97      09                                 0000 1001      61 3
 98      36              LOAD L,*8          0011 0110      62 3
 99      08                                 0000 1000      63 3
100      C7              LOAD A,(HL)        1100 0111      64 3
101      07              RET                0000 0111      65 3
102      **                                 **** ****      66 3
103      **                                 **** ****      67 3
                    S135 DIR  104,3
104      2E              LOAD H,*9          0010 1110      68 3
105      09                                 0000 1001      69 3
106      F5              LOAD L,H           1111 0101      6A 3
107      C7              LOAD A,(HL)        1100 0111      6B 3
108      07              RET                0000 0111      6C 3
109      **                                 **** ****      6D 3
110      **                                 **** ****      6E 3
111      **                                 **** ****      6F 3
                    S136 DIR  112,3
112      46              CALL SE15          0100 0110      70 3
113      39                                 0011 1001      71 3
114      01                                 0000 0001      72 3
115      2E              LOAD H,*9          0010 1110      73 3
116      09                                 0000 1001      74 3
117      FA              LOAD (HL),C        1111 1010      75 3
118      07              RET                0000 0111      76 3
119      **                                 **** ****      77 3
                    S137 DIR  120,3
120      46              CALL SE15          0100 0110      78 3
121      39                                 0011 1001      79 3
```

```
122  01                                0000 0001  7A 3
123  2E        LOAD H,*9               0010 1110  7B 3
124  09                                0000 1001  7C 3
125  C7        LOAD A,(HL)             1100 0111  7D 3
126  07        RET                     0000 0111  7E 3
127  **                                **** ****  7F 3
                S150 DIR  128,3
128  46        CALL SA3                0100 0110  80 3
129  DC                                1101 1100  81 3
130  03                                0000 0011  82 3
131  06   S150ELOAD A,*244             0000 0110  83 3
132  F4                                1111 0100  84 3
133  73        OUT  S1                 0111 0011  85 3
134  46        CALL SA16               0100 0110  86 3
135  24                                0010 0100  87 3
136  07                                0000 0111  88 3
137  6A        CALL ZS,SA15            0110 1010  89 3
138  18                                0001 1000  8A 3
139  07                                0000 0111  8B 3
140  4A        CALL ZC,SA14            0100 1010  8C 3
141  07                                0000 0111  8D 3
142  07                                0000 0111  8E 3
143  44        JUMP SE10               0100 0100  8F 3
144  50                                0101 0000  90 3
145  06                                0000 0110  91 3
146  46   SS120CALL SA1                0100 0110  92 3
147  BF                                1011 1111  93 3
148  03                                0000 0011  94 3
149  44        JUMP SE10               0100 0100  95 3
150  50                                0101 0000  96 3
151  06                                0000 0110  97 3
                S151 DIR  152,3
152  46        CALL SE9                0100 0110  98 3
153  44                                0100 0100  99 3
154  06                                0000 0110  9A 3
155  F2        LOAD L,C                1111 0010  9B 3
156  44        JUMP S150E              0100 0100  9C 3
157  83                                1000 0011  9D 3
158  03                                0000 0011  9E 3
159  **                                **** ****  9F 3
                S152 DIR  160,3
160  46        CALL SA3                0100 0110  A0 3
161  DC                                1101 1100  A1 3
162  03                                0000 0011  A2 3
163  46   S152ECALL SA16               0100 0110  A3 3
164  24                                0010 0100  A4 3
165  C7                                0000 0111  A5 3
166  6A        CALL ZS,SA18            0110 1010  A6 3
167  44                                0100 0100  A7 3
168  07                                0000 0111  A8 3
169  4A        CALL ZC,SA17            0100 1010  A9 3
170  33                                0011 0011  AA 3
171  07                                0000 0111  AB 3
172  44        JUMP SE10               0100 0100  AC 3
173  50                                0101 0000  AD 3
174  06                                0000 0110  AE 3
175  46   SS104CALL SE7                0100 0110  AF 3
176  A2                                1010 0010  B0 3
177  04                                0000 0100  B1 3
178  40        JUMP CC,S104A           0100 0000  B2 3
179  A9                                1010 1001  B3 3
180  01                                0000 0001  B4 3
181  44        JUMP SE10               0100 0100  B5 3
```

```
182   50                                  0101 0000   B6 3
183   06                                  0000 0110   B7 3
                S153 DIR  184.*7
184   46        CALL SE9                  0100 0110   B8 3
185   44                                  0100 0100   B9 3
186   06                                  0000 0110   BA 3
187   F2        LOAD L.C                  1111 0010   BB 3
188   44        JUMP S152E                0100 0100   BC 3
189   A3                                  1010 0011   BD 3
190   03                                  0000 0011   BE 3
191   46  SA1   CALL SA3                  0100 0110   BF 3
192   DC                                  1101 1100   C0 3
193   03                                  0000 0011   C1 3
194   1E  SA1A  LOAD D.*0                 0001 1110   C2 3
195   00                                  0000 0000   C3 3
196   F3        LOAD L.D                  1111 0011   C4 3
197   46  SA1B  CALL SA6                  0100 0110   C5 3
198   FA                                  1111 1010   C6 3
199   00                                  0000 0000   C7 3
200   C8        LOAD B.A                  1100 1000   C8 3
201   02        RL   A                    0000 0010   C9 3
202   A9        XOR  B                    1010 1001   CA 3
203   70        JUMP SS.SE10              0111 0000   CB 3
204   50                                  0101 0000   CC 3
205   06                                  0000 0110   CD 3
206   46        CALL SE7                  0100 0110   CE 3
207   A2                                  1010 0010   CF 3
208   C4                                  0000 0100   D0 3
209   40        JUMP CC.SA1B              0100 0000   D1 3
210   C5                                  1100 0101   D2 3
211   03                                  0000 0011   D3 3
212   16        LOAD C.*1                 0001 0110   D4 3
213   01                                  0000 0001   D5 3
214   46        CALL SA11                 0100 0110   D6 3
215   9F                                  1001 1111   D7 3
216   06                                  0000 0110   D8 3
217   44        JUMP SA1A                 0100 0100   D9 3
218   C2                                  1100 0010   DA 3
219   03                                  0000 0011   DB 3
220   2E  SA3   LOAD H.*9                 0010 1110   DC 3
221   09                                  0000 1001   DD 3
222   36        LOAD L.*72                0011 0110   DE 3
223   48                                  0100 1000   DF 3
224   F8        LOAD (HL).A               1111 1000   E0 3
225   30        INR  L                    0011 0000   E1 3
226   F9        LOAD (HL).B               1111 1001   E2 3
227   30        INR  L                    0011 0000   E3 3
228   FA        LOAD (HL).C               1111 1010   E4 3
229   30        INR  L                    0011 0000   E5 3
230   FB        LOAD (HL).D               1111 1011   E6 3
231   36  SA3E  LOAD L.*7                 0011 0110   E7 3
232   07                                  0000 0111   E8 3
233   E7        LOAD E.(HL)               1110 0111   E9 3
234   20        INR  E                    0010 0000   EA 3
235   FC        LOAD (HL).E               1111 1100   EB 3
236   29        DCR  H                    0010 1001   EC 3
237   F4        LOAD L.E                  1111 0100   ED 3
238   F7        LOAD L.(HL)               1111 0111   EE 3
239   E8        LOAD H.A                  1110 1000   EF 3
240   06        LOAD A.*10                0000 0110   F0 3
241   0A                                  0000 1010   F1 3
242   86        ADD  L                    1000 0110   F2 3
243   F0        LOAD L.A                  1111 0000   F3 3
```

```
244   E0        LOAD E,A          1110 0000   F4 3
245   C5        LOAD A,H          1100 0101   F5 3
246   07        RET               0000 0111   F6 3
247   **                          **** ****   F7 3
                S120 DTR  248,7
248   46          CALL SA3        0100 0110   F8 3
249   DC                          1101 1100   F9 3
250   03                          0000 0011   FA 3
251   44          JUMP SS120      0100 0100   FB 3
252   92                          1001 0010   FC 3
253   03                          0000 0011   FD 3
```

```
                    SD51            4          0
                    SE6             6         57
                    SE2B            6         22
                    SE10            6         80
                    SE10E           6         86
                    SD83            4          8
                    SD83E           4          9
                    SD85            4         16
                    SD85E           4         17
                    SD87            4         24
                    SD87E           4         25
                    SD55            4         32
                    SD84            4         40
                    SD86            4         48
                    SD88            4         56
                    SD59            4         64
                    SD91            4         72
                    SD93            4         80
                    SD81            4         88
                    SD81E           4         89
                    SD63            4         96
                    SD63E           4        100
                    SD92            4        104
                    SE12            4        110
                    SD82            4        120
                    SSD93           4        127
                    SD52            4        136
                    SSD63           4        138
                    SD53            4        144
                    SSD92           4        146
                    SD54            4        152
                    SD56            4        160
                    SE7             4        162
                    SD57            4        176
                    TSD92           4        178
                    SD58            4        184
                    SD60            4        192
                    SSE2            4        194
                    SD61            4        200
                    SA9             4        202
                    SA9A            4        207
                    SD62            4        216
                    SD64            4        224
                    SD65            4        232
                    SD66            4        240
                    SE8             4        244
                    SSE8            4        254
                SD51 DIR  0,4
                SE6  EQU  57,6
                SE2B EQU  22,6
                SE10 EQU  80,6
                SE10EEQU  86,6
0       46          CALL SE6            0100 0110    00 4
1       39                              0011 1001    01 4
2       06                              0000 0110    02 4
3       C6          LOAD A,L            1100 0110    03 4
4       07          RET                 0000 0111    04 4
5       **                              **** ****    05 4
6       **                              **** ****    06 4
7       **                              **** ****    07 4
                SD83 DIR  8,4
8       E2          LOAD E,C            1110 0010    08 4
9       3C      SD83EC OMP A,*0         0011 1100    09 4
```

```
10   00                                  0000 0000   0A 4
11   13          RET   SC               0001 0011   0B 4
12   44          JUMP  S081E             0100 0100   0C 4
13   59                                  0101 1001   0D 4
14   04                                  0000 0100   0E 4
15   **                                  **** ****   0F 4
                 S085  DIR   16,4
16   E2          LOAD  E,C               1110 0010   10 4
17   3C     S085EC OMP  A,*1             0011 1100   11 4
18   01                                  0000 0001   12 4
19   33          RET   SS                0011 0011   13 4
20   44          JUMP  S081E             0100 0100   14 4
21   59                                  0101 1001   15 4
22   04                                  0000 0100   16 4
23   **                                  **** ****   17 4
                 S087  DIR   24,4
24   E2          LOAD  E,C               1110 0010   18 4
25   04     S087EADD  A,*0               0000 0100   19 4
26   00                                  0000 0000   1A 4
27   0B          RET   ZC                0000 1011   1B 4
28   44          JUMP  S081E             0100 0100   1C 4
29   59                                  0101 1001   1D 4
30   04                                  0000 0100   1E 4
31   **                                  **** ****   1F 4
                 S055  DIR   32,4
32   46          CALL  SE6               0100 0110   20 4
33   39                                  0011 1001   21 4
34   06                                  0000 0110   22 4
35   CE          LOAD  B,L               1100 1110   23 4
36   07          RET                     0000 0111   24 4
37   **                                  **** ****   25 4
38   **                                  **** ****   26 4
39   **                                  **** ****   27 4
                 S084  DIR   40,4
40   46          CALL  SE6               0100 0110   28 4
41   39                                  0011 1001   29 4
42   06                                  0000 0110   2A 4
43   E6          LOAD  E,L               1110 0110   2B 4
44   44          JUMP  S083E             0100 0100   2C 4
45   09                                  0000 1001   2D 4
46   04                                  0000 0100   2E 4
47   **                                  **** ****   2F 4
                 S08E  DIR   48,4
48   46          CALL  SE6               0100 0110   30 4
49   39                                  0011 1001   31 4
50   06                                  0000 0110   32 4
51   E6          LOAD  E,L               1110 0110   33 4
52   44          JUMP  S085E             0100 0100   34 4
53   11                                  0001 0001   35 4
54   04                                  0000 0100   36 4
55   **                                  **** ****   37 4
                 S088  DIR   56,4
56   46          CALL  SE6               0100 0110   38 4
57   39                                  0011 1001   39 4
58   06                                  0000 0110   3A 4
59   E6          LOAD  E,L               1110 0110   3B 4
60   44          JUMP  S087E             0100 0100   3C 4
61   19                                  0001 1001   3D 4
62   04                                  0000 0100   3E 4
63   **                                  **** ****   3F 4
                 S059  DIR   64,4
64   46          CALL  SE6               0100 0110   40 4
65   39                                  0011 1001   41 4
```

```
  66   06                          0000 0110   42 4
  67   D6        LOAD C,L          1101 0110   43 4
  68   07        RET               0000 0111   44 4
  69   **                          **** ****   45 4
  70   **                          **** ****   46 4
  71   **                          **** ****   47 4
                 SD91 DIR  72,4
  72   46        CALL SA9          0100 0110   48 4
  73   CA                          1100 1010   49 4
  74   04                          0000 0100   4A 4
  75   36        LOAD L,#7         0011 0110   4B 4
  76   07                          0000 0111   4C 4
  77   11        DCR  C            0001 0001   4D 4
  78   FA        LOAD (HL),C       1111 1010   4E 4
  79   07        RET               0000 0111   4F 4
                 SD93 DIR  80,4
  80   2E        LOAD H,#9         0010 1110   50 4
  81   09                          0000 1001   51 4
  82   36        LOAD L,#7         0011 0110   52 4
  83   07                          0000 0111   53 4
  84   44        JUMP SSD93        0100 0100   54 4
  85   7F                          0111 1111   55 4
  86   04                          0000 0100   56 4
  87   **                          **** ****   57 4
                 SD81 DIR  88,4
  88   E2        LOAD E,C          1110 0010   58 4
  89   36     SD81ELOAD L,#7       0011 0110   59 4
  90   07                          0000 0111   5A 4
  91   2E        LOAD H,#9         0010 1110   5B 4
  92   09                          0000 1001   5C 4
  93   21        DCR  E            0010 0001   5D 4
  94   FC        LOAD (HL),E       1111 1100   5E 4
  95   07        RET               0000 0111   5F 4
                 SD63 DIR  96,4
  96   46        CALL SE6          0100 0110   60 4
  97   39                          0011 1001   61 4
  98   06                          0000 0110   62 4
  99   E6        LOAD E,L          1110 0110   63 4
 100   F2     SD63ELOAD L,C        1111 0010   64 4
 101   44        JUMP SSD63        0100 0100   65 4
 102   8A                          1000 1010   66 4
 103   04                          0000 0100   67 4
                 SD92 DIR  104,4
 104   46        CALL SA9          0100 0110   68 4
 105   CA                          1100 1010   69 4
 106   04                          0000 0100   6A 4
 107   44        JUMP SSD92        0100 0100   6B 4
 108   92                          1001 0010   6C 4
 109   04                          0000 0100   6D 4
 110   2E     SE12 LOAD H,#3       0010 1110   6E 4
 111   09                          0000 1001   6F 4
 112   36        LOAD L,#64        0011 0110   70 4
 113   40                          0100 0000   71 4
 114   46        CALL SE10F        0100 0110   72 4
 115   56                          0101 0110   73 4
 116   06                          0000 0110   74 4
 117   30        INR  L            0011 0000   75 4
 118   E7        LOAD E,(HL)       1110 0111   76 4
 119   07        RET               0000 0111   77 4
                 SD82 DIR  120,4
 120   46        CALL SE6          0100 0110   78 4
 121   39                          0011 1001   79 4
 122   06                          0000 0110   7A 4
```

```
123   E6          LOAD E,L          1110 0110   7B 4
124   44          JUMP S081E        0100 0100   7C 4
125   59                            0101 1001   7D 4
126   04                            0000 0100   7E 4
127   31    SS093 DCR  L            0011 0001   7F 4
128   E7          LOAD E,(HL)       1110 0111   80 4
129   30          INR  L            0011 0000   81 4
130   FC          LOAD (HL),E       1111 1100   82 4
131   31          DCR  L            0011 0001   83 4
132   48          JUMP ZC,SS093     0100 1000   84 4
133   7F                            0111 1111   85 4
134   04                            0000 0100   86 4
135   07          RET               0000 0111   87 4
            S052  DIR  136,4
136   C1          LOAD A,B          1100 0001   88 4
137   07          RET               0000 0111   89 4
138   2E    SS063 LOAD H,*8         0010 1110   8A 4
139   08                            0000 1000   8B 4
140   FC          LOAD (HL),E       1111 1100   8C 4
141   07          RET               0000 0111   8D 4
142   **                            **** ****   8E 4
143   **                            **** ****   8F 4
            S053  DIR  144,4
144   C2          LOAD A,C          1100 0010   90 4
145   07          RET               0000 0111   91 4
146   46    SS092 CALL SE6          0100 0110   92 4
147   39                            0011 1001   93 4
148   06                            0000 0110   94 4
149   44          JUMP TS092        0100 0100   95 4
150   B2                            1011 0010   96 4
151   04                            0000 0100   97 4
            S054  DIR  152,4
152   F2          LOAD L,C          1111 0010   98 4
153   2E          LOAD H,*8         0010 1110   99 4
154   08                            0000 1000   9A 4
155   C7          LOAD A,(HL)       1100 0111   9B 4
156   07          RET               0000 0111   9C 4
157   **                            **** ****   9D 4
158   **                            **** ****   9E 4
159   **                            **** ****   9F 4
            S056  DIR  160,4
160   C8          LOAD B,A          1100 1000   A0 4
161   07          RET               0000 0111   A1 4
162   30    SE7   INR  L            0011 0000   A2 4
163   0B          RET  ZC           0000 1011   A3 4
164   18          INR  D            0001 1000   A4 4
165   2E          LOAD H,*9         0010 1110   A5 4
166   09                            0000 1001   A6 4
167   36          LOAD L,*8         0011 0110   A7 4
168   08                            0000 1000   A8 4
169   EF          LOAD H,(HL)       1110 1111   A9 4
170   36          LOAD L,*0         0011 0110   AA 4
171   00                            0000 0000   AB 4
172   29          DCR  H            0010 1001   AC 4
173   C5          LOAD A,H          1100 0101   AD 4
174   BB          COMP A,D          1011 1011   AE 4
175   07          RET               0000 0111   AF 4
            S057  DIR  176,4
176   CA          LOAD B,C          1100 1010   B0 4
177   07          RET               0000 0111   B1 4
178   E6    TS092 LOAD E,L          1110 0110   B2 4
179   44          JUMP S081E        0100 0100   B3 4
180   59                            0101 1001   B4 4
```

```
181   04                                  0000 0100   B5 4
182   **                                  **** ****   B6 4
183   **                                  **** ****   B7 4
                SD58 DIR  184,4
184   F2          LOAD L,C                1111 0010   B8 4
185   2E          LOAD H,*8               0010 1110   B9 4
186   08                                  0000 1000   BA 4
187   CF          LOAD B,(HL)             1100 1111   BB 4
188   07          RET                     0000 0111   BC 4
189   **                                  **** ****   BD 4
190   **                                  **** ****   BE 4
191   **                                  **** ****   BF 4
                SD60 DIR  192,4
192   D0          LOAD C,A                1101 0000   C0 4
193   07          RET                     0000 0111   C1 4
194   C1      SSE2 LOAD A,B               1100 0001   C2 4
195   80          ADD  A                  1000 0000   C3 4
196   82          ADD  C                  1000 0010   C4 4
197   44          JUMP SE2B               0100 0100   C5 4
198   16                                  0001 0110   C6 4
199   06                                  0000 0110   C7 4
                SD61 DIR  200,4
200   D1          LOAD C,B                1101 0001   C8 4
201   07          RET                     0000 0111   C9 4
202   2E      SA9  LOAD H,*9              0010 1110   CA 4
203   09                                  0000 1001   CB 4
204   36          LOAD L,*7               0011 0110   CC 4
205   07                                  0000 0111   CD 4
206   E7          LOAD E,(HL)             1110 0111   CE 4
207   31      SA9A DCR  L                 0011 0001   CF 4
208   DF          LOAD D,(HL)             1101 1111   D0 4
209   FC          LOAD (HL),E             1111 1100   D1 4
210   E3          LOAD E,D                1110 0011   D2 4
211   48          JUMP ZC,SA9A            0100 1000   D3 4
212   CF                                  1100 1111   D4 4
213   04                                  0000 0100   D5 4
214   07          RET                     0000 0111   D6 4
215   **                                  **** ****   D7 4
                SD62 DIR  216,4
216   F2          LOAD L,C                1111 0010   D8 4
217   2E          LOAD H,*8               0010 1110   D9 4
218   08                                  0000 1000   DA 4
219   D7          LOAD C,(HL)             1101 0111   DB 4
220   07          RET                     0000 0111   DC 4
221   **                                  **** ****   DD 4
222   **                                  **** ****   DE 4
223   **                                  **** ****   DF 4
                SD64 DIR  224,4
224   E0          LOAD E,A                1110 0000   E0 4
225   44          JUMP SD63E              0100 0100   E1 4
226   64                                  0110 0100   E2 4
227   04                                  0000 0100   E3 4
228   **                                  **** ****   E4 4
229   **                                  **** ****   E5 4
230   **                                  **** ****   E6 4
231   **                                  **** ****   E7 4
                SD65 DIR  232,4
232   E1          LOAD E,B                1110 0001   E8 4
233   44          JUMP SD63E              0100 0100   E9 4
234   64                                  0110 0100   EA 4
235   04                                  0000 0100   EB 4
236   **                                  **** ****   EC 4
237   **                                  **** ****   ED 4
```

```
238     **                                  **** ****     EE 4
239     **                                  **** ****     EF 4
                SD6E DIR   240,4
240     E2          LOAD E,C                1110 0010     F0 4
241     44          JUMP SD63E              0100 0100     F1 4
242     64                                  0110 0100     F2 4
243     04                                  0000 0100     F3 4
244     C6      SE8 LOAD A,L                1100 0110     F4 4
245     04          ADD  A,*0               0000 0100     F5 4
246     00                                  0000 0000     F6 4
247     48          JUMP ZC,SSE8            0100 1000     F7 4
248     FE                                  1111 1110     F8 4
249     04                                  0000 0100     F9 4
250     29          DCR  H                  0010 1001     FA 4
251     C5          LOAD A,H                1100 0101     FB 4
252     BC          COMP A,E                1011 1100     FC 4
253     23          RET  CS                 0010 0011     FD 4
254     31      SSE8 DCR  L                 0011 0001     FE 4
255     07          RET                     0000 0111     FF 4
```

S001	5	0	
SE2	6	0	
SE3	7	114	
SE6	6	57	
SE5	7	172	
SE11	6	92	
SE12	4	110	
SE17	7	234	
SA7	2	185	
S001A	5	2	
S002	5	8	
S002A	5	11	
S003	5	16	
S022	5	24	
SS010	5	26	
S004	5	32	
S004A	5	34	
S005	5	40	
S005A	5	43	
S006	5	48	
S025	5	56	
S007	5	64	
ES007	5	67	
S008	5	72	
S009	5	80	
S023	5	88	
ES011	5	90	
S010	5	96	
ES010	5	99	
S011	5	104	
S012	5	112	
S026	5	120	
S008Z	5	122	
S013	5	128	
S1 SE1	5	130	
S014	5	136	
S015	5	144	
SS007	5	149	
S016	5	160	
S3 SE1	5	162	
S017	5	168	
S018	5	176	
S2 SE1	5	181	
ES008	5	185	
S019	5	192	
SS021	5	194	
S020	5	200	
S021	5	208	
SE1	5	211	
SE1A	5	228	
SE1B	5	245	

```
     S001  DIR   0,5
     SE2   EQU   0,6
     SE3   EQU   114,7
     SE6   EQU   57,6
     SE5   EQU   172,7
     SE11  EQU   92,6
     SE12  EQU   110,4
     SE17  EQU   234,7
     SA7   EQU   185,2
0    E2         LOAD  E,C          1110 0010    00 5
1    D1         LOAD  C,B          1101 0001    01 5
2    46    S001ACALL  SE3          0100 0110    02 5
```

```
 3    72                                  0111 0010    03 5
 4    07                                  0000 0111    04 5
 5    D4          LOAD C,E                1101 0100    05 5
 6    07          RET                     0000 0111    06 5
 7    **                                  **** ****    07 5
                  SD02 DIR  8,5
 8    46          CALL SA7                0100 0110    08 5
 9    B9                                  1011 1001    09 5
10    02                                  0000 0010    0A 5
11    E2    SD02AL OAD E,C                1110 0010    0B 5
12    D6          LOAD C,L                1101 0110    0C 5
13    44          JUMP SD01A              0100 0100    0D 5
14    02                                  0000 0010    0E 5
15    05                                  0000 0101    0F 5
                  SD03 DIR  16,5
16    46          CALL SE6                0100 0110    10 5
17    39                                  0011 1001    11 5
18    06                                  0000 0110    12 5
19    44          JUMP SD02A              0100 0100    13 5
20    0B                                  0000 1011    14 5
21    05                                  0000 0101    15 5
22    **                                  **** ****    16 5
23    **                                  **** ****    17 5
                  SD22 DIR  24,5
24    80          ADD  A,A                1000 0000    18 5
25    07          RET                     0000 0111    19 5
26    46    SS010CALL SE2                 0100 0110    1A 5
27    00                                  0000 0000    1B 5
28    06                                  0000 0110    1C 5
29    44          JUMP SE12               0100 0100    1D 5
30    6E                                  0110 1110    1E 5
31    04                                  0000 0100    1F 5
                  SD04 DIR  32,5
32    E2          LOAD E,C                1110 0010    20 5
33    01          LOAD C,B                1101 0001    21 5
34    46    SD04AC ALL SE5                0100 0110    22 5
35    AC                                  1010 1100    23 5
36    07                                  0000 0111    24 5
37    D4          LOAD C,E                1101 0100    25 5
38    07          RET                     0000 0111    26 5
39    **                                  **** ****    27 5
                  SD05 DIR  40,5
40    46          CALL SA7                0100 0110    28 5
41    B9                                  1011 1001    29 5
42    02                                  0000 0010    2A 5
43    E2    SD05AL OAD E,C                1110 0010    2B 5
44    D6          LOAD C,L                1101 0110    2C 5
45    44          JUMP SD04A              0100 0100    2D 5
46    22                                  0010 0010    2E 5
47    05                                  0000 0101    2F 5
                  SD0E DIR  48,5
48    46          CALL SE6                0100 0110    30 5
49    39                                  0011 1001    31 5
50    06                                  0000 0110    32 5
51    44          JUMP SD05A              0100 0100    33 5
52    2B                                  0010 1011    34 5
53    05                                  0000 0101    35 5
54    **                                  **** ****    36 5
55    **                                  **** ****    37 5
                  SD25 DIR  56,5
56    04          ADD  A,*0               0000 0100    38 5
57    00                                  0000 0000    39 5
58    1A          RRC  A                  0001 1010    3A 5
```

```
 59   07          RET                  0000 0111    3B 5
 60   **                               **** ****    3C 5
 61   **                               **** ****    3D 5
 62   **                               **** ****    3E 5
 63   **                               **** ****    3F 5
                SS07 DIR  64,5
 64   46          CALL SE11            0100 0110    40 5
 65   5C                               0101 1100    41 5
 66   06                               0000 0110    42 5
 67   D1     ES007LOAD C,B             1101 0001    43 5
 68   C8          LOAD B,A             1100 1000    44 5
 69   44          JUMP SS007           0100 0100    45 5
 70   95                               1001 0101    46 5
 71   05                               0000 0101    47 5
                SS08 DIR  72,5
 72   46          CALL SA7             0100 0110    48 5
 73   B9                               1011 1001    49 5
 74   02                               0000 0010    4A 5
 75   44          JUMP ES008           0100 0100    4B 5
 76   B9                               1011 1001    4C 5
 77   05                               0000 0101    4D 5
 78   **                               **** ****    4E 5
 79   **                               **** ****    4F 5
                SS09 DIR  80,5
 80   46          CALL SE6             0100 0110    50 5
 81   39                               0011 1001    51 5
 82   06                               0000 0110    52 5
 83   44          JUMP ES008           0100 0100    53 5
 84   B9                               1011 1001    54 5
 85   05                               0000 0101    55 5
 86   **                               **** ****    56 5
 87   **                               **** ****    57 5
                SS23 DIR  88,5
 88   02          RL   A               0000 0010    58 5
 89   07          RET                  0000 0111    59 5
 90   46     ES011CALL SS087           0100 0110    5A 5
 91   7A                               0111 1010    5B 5
 92   05                               0000 0101    5C 5
 93   44          JUMP ES010           0100 0100    5D 5
 94   63                               0110 0011    5E 5
 95   05                               0000 0101    5F 5
                SS10 DIR  96,5
 96   46          CALL SE11            0100 0110    60 5
 97   5C                               0101 1100    61 5
 98   06                               0000 0110    62 5
 99   D1     ES010LOAD C,B             1101 0001    63 5
100   C8          LOAD B,A             1100 1000    64 5
101   44          JUMP SS010           0100 0100    65 5
102   1A                               0001 1010    66 5
103   05                               0000 0101    67 5
                SS11 DIR  104,5
104   46          CALL SA7             0100 0110    68 5
105   B9                               1011 1001    69 5
106   02                               0000 0010    6A 5
107   44          JUMP ES011           0100 0100    6B 5
108   5A                               0101 1010    6C 5
109   05                               0000 0101    6D 5
110   **                               **** ****    6E 5
111   **                               **** ****    6F 5
                SS12 DIR  112,5
112   46          CALL SE6             0100 0110    70 5
113   39                               0011 1001    71 5
114   06                               0000 0110    72 5
```

```
115  44          JUMP ES011      0100 0100   73 5
116  5A                          0101 1010   74 5
117  05                          0000 0101   75 5
118  **                          **** ****   76 5
119  **                          **** ****   77 5
                 S02E DIR  120,5
120  0A          RR   A          0000 1010   78 5
121  07          RET             0000 0111   79 5
122  DE     SD08ZLOAD D,L        1101 1110   7A 5
123  46          CALL SE11       0100 0110   7B 5
124  5C                          0101 1100   7C 5
125  06                          0000 0110   7D 5
126  CB          LOAD B,D        1100 1011   7E 5
127  07          RET             0000 0111   7F 5
                 S013 DIR  128,5
128  A1          AND  A,B        1010 0001   80 5
129  07          RET             0000 0111   81 5
130  C3     S1SE1LOAD A,D        1100 0011   82 5
131  44          JUMP SE18       0100 0100   83 5
132  F5                          1111 0101   84 5
133  05                          0000 0101   85 5
134  **                          **** ****   86 5
135  **                          **** ****   87 5
                 S014 DIR  136,5
136  46          CALL SA7        0100 0110   88 5
137  B9                          1011 1001   89 5
138  02                          0000 0010   8A 5
139  A6          AND  A,L        1010 0110   8B 5
140  07          RET             0000 0111   8C 5
141  **                          **** ****   8D 5
142  **                          **** ****   8E 5
143  **                          **** ****   8F 5
                 S015 DIR  144,5
144  46          CALL SE6        0100 0110   90 5
145  39                          0011 1001   91 5
146  06                          0000 0110   92 5
147  A6          AND  A,L        1010 0110   93 5
148  07          RET             0000 0111   94 5
149  46     SS007CALL SE1        0100 0110   95 5
150  D3                          1101 0011   96 5
151  05                          0000 0101   97 5
152  44          JUMP SE12       0100 0100   98 5
153  6E                          0110 1110   99 5
154  04                          0000 0100   9A 5
155  **                          **** ****   9B 5
156  **                          **** ****   9C 5
157  **                          **** ****   9D 5
158  **                          **** ****   9E 5
159  **                          **** ****   9F 5
                 S016 DIR  160,5
160  A9          XOR  A,B        1010 1001   A0 5
161  07          RET             0000 0111   A1 5
162  34     S3SE10R   A,*1       0011 0100   A2 5
163  01                          0000 0001   A3 5
164  44          JUMP SE12       0100 0100   A4 5
165  6E                          0110 1110   A5 5
166  04                          0000 0100   A6 5
167  **                          **** ****   A7 5
                 S017 DIR  168,5
168  46          CALL SA7        0100 0110   A8 5
169  B9                          1011 1001   A9 5
170  02                          0000 0010   AA 5
171  AE          XOR  A,L        1010 1110   AB 5
```

```
172   07          RET                 0000 0111   AC 5
173   **                              **** ****   AD 5
174   **                              **** ****   AE 5
175   **                              **** ****   AF 5
                  SD18 DIR  176.5
176   46          CALL SE6            0100 0110   B0 5
177   39                              0011 1001   B1 5
178   06                              0000 0110   B2 5
179   AE          XOR  A.L            1010 1110   B3 5
180   07          RET                 0000 0111   B4 5
181   91    S2SE1SUB  A.B             1001 0001   B5 5
182   44          JUMP SE18           0100 0100   B6 5
183   F5                              1111 0101   B7 5
184   05                              0000 0101   B8 5
185   46    ES008CALL SS008           0100 0110   B9 5
186   7A                              0111 1010   BA 5
187   05                              0000 0101   BB 5
188   44          JUMP ES007          0100 0100   BC 5
189   43                              0100 0011   BD 5
190   05                              0000 0101   BE 5
191   **                              **** ****   BF 5
                  SD19 DIR  192.5
192   B1          OR   A.B            1011 0001   C0 5
193   07          RET                 0000 0111   C1 5
194   46    SS021CALL SE6             0100 0110   C2 5
195   39                              0011 1001   C3 5
196   06                              0000 0110   C4 5
197   BE          OR   A.L            1011 0110   C5 5
198   07          RET                 0000 0111   C6 5
199   **                              **** ****   C7 5
                  SD20 DIR  200.5
200   46          CALL SA7            0100 0110   C8 5
201   B9                              1011 1001   C9 5
202   C2                              0000 0010   CA 5
203   B6          OR   A.L            1011 0110   CB 5
204   07          RET                 0000 0111   CC 5
205   **                              **** ****   CD 5
206   **                              **** ****   CE 5
207   **                              **** ****   CF 5
                  SD21 DIR  208.5
208   44          JUMP SSD21          0100 0100   D0 5
209   C2                              1100 0010   D1 5
210   05                              0000 0101   D2 5
211   90    SE1   SUB  A.A            1001 0000   D3 5
212   B9          COMP A.B            1011 1001   D4 5
213   2B          RET  ZS             0010 1011   D5 5
214   BA          COMP A.C            1011 1010   D6 5
215   2B          RET  ZS             0010 1011   D7 5
216   46          CALL SE11           0100 0110   D8 5
217   5C                              0101 1100   D9 5
218   06                              0000 0110   DA 5
219   E0          LOAD E.A            1110 0000   DB 5
220   36          LOAD L.*8           0011 0110   DC 5
221   08                              0000 1000   DD 5
222   DC          LOAD D.E            1101 1100   DE 5
223   C2          LOAD A.C            1100 0010   DF 5
224   84          ADD  A.E            1000 0100   E0 5
225   12          RLC  A              0001 0010   E1 5
226   AA          XOR  A.C            1010 1010   E2 5
227   E8          LOAD H.A            1110 1000   E3 5
228   C2    SE1A  LOAD A.C            1100 0010   E4 5
229   0A          RR   A              0000 1010   E5 5
230   D0          LOAD C.A            1101 0000   E6 5
```

231	C5		LOAD	A,H	1100 0101	E7 5
232	0A		RR	A	0000 1010	E8 5
233	84		ADD	A,E	1000 0100	E9 5
234	E8		LOAD	H,A	1110 1000	EA 5
235	50		JUMP	SC,S1SE1	0101 0000	EB 5
236	82				1000 0010	EC 5
237	05				0000 0101	ED 5
238	C2		LOAD	A,C	1100 0010	EE 5
239	84		ADD	A,E	1000 0100	EF 5
240	C3		LOAD	A,D	1100 0011	F0 5
241	70		JUMP	SS,S2SE1	0111 0000	F1 5
242	B5				1011 0101	F2 5
243	05				0000 0101	F3 5
244	81		ADD	A,B	1000 0001	F4 5
245	31	SE1B	DCR	L	0011 0001	F5 5
246	68		JUMP	ZS,S3SE1	0110 1000	F6 5
247	A2				1010 0010	F7 5
248	05				0000 0101	F8 5
249	46		CALL	SE17	0100 0110	F9 5
250	EA				1110 1010	FA 5
251	07				0000 0111	FB 5
252	D8		LOAD	D,A	1101 1000	FC 5
253	44		JUMP	SE1A	0100 0100	FD 5
254	E4				1110 0100	FE 5
255	05				0000 0101	FF 5

```
SE2             6         0
SSE2            4       194
SE3             7       114
SE5             7       172
SE7             4       162
SE8             4       244
SE12            4       110
SE14            0       226
SE17            7       234
SA4             2       246
SA5             2        65
SA6             0       250
SA13C           7         0
SE2A            6        13
SE2B            6        22
SE4             6        35
SE6             6        57
SE9             6        68
SE9E            6        74
SE10            6        80
SE10E           6        86
SE11            6        92
SA2             6       102
SA2A            6       124
SA10            6       130
SA10A           6       138
SA10B           6       141
SA10C           6       147
SA11            6       159
SA11A           6       168
SA11B           6       171
SA12A           6       190
SA12B           6       201
SA13            6       211
SA13A           6       214
SA13B           6       218
SA13F           6       226
SA13D           6       247
```

```
SE2   DIR  0,6
SSE2  EQU  194,4
SE3   EQU  114,7
SE5   EQU  172,7
SE7   EQU  162,4
SE8   EQU  244,4
SE12  EQU  110,4
SE14  EQU  226,0
SE17  EQU  234,7
SA4   EQU  246,2
SA5   EQU  65,2
SA6   EQU  250,0
SA13CEQU   0,7
```

```
0    90          SUB   A,A          1001 0000    00 6
1    B9          COMP  A,B          1011 1001    01 6
2    2B          RET   ZS           0010 1011    02 6
3    BA          COMP  A,C          1011 1010    03 6
4    6A          CALL  ZS,SE14      0110 1010    04 6
5    E2                             1110 0010    05 6
6    00                             0000 0000    06 6
7    46          CALL  SE11         0100 0110    07 6
8    5C                             0101 1100    08 6
9    06                             0000 0110    09 6
10   D8          LOAD  D,A          1101 1000    0A 6
11   36          LOAD  L,*7         0011 0110    0B 6
```

```
12  07                                  0000 0111   0C 6
13  C2      SE2A  LOAD  A,C            1100 0010   0D 6
14  A9            XOR   A,B            1010 1001   0E 6
15  70            JUMP  SS,SSE2        0111 0000   0F 6
16  C2                                  1100 0010   10 6
17  04                                  0000 0100   11 6
18  18            INR   D              0001 1000   12 6
19  C1            LOAD  A,B            1100 0001   13 6
20  80            ADD   A,A            1000 0000   14 6
21  92            SUB   A,C            1001 0010   15 6
22  C8      SE2B  LOAD  B,A            1100 1000   16 6
23  C3            LOAD  A,D            1100 0011   17 6
24  80            ADD   A,A            1000 0000   18 6
25  D8            LOAD  D,A            1101 1000   19 6
26  31            DCR   L              0011 0001   1A 6
27  48            JUMP  ZC,SE2A        0100 1000   1B 6
28  0D                                  0000 1101   1C 6
29  06                                  0000 0110   1D 6
30  04            ADD   A,*129         0000 0100   1E 6
31  81                                  1000 0001   1F 6
32  44            JUMP  SE12           0100 0100   20 6
33  6E                                  0110 1110   21 6
34  04                                  0000 0100   22 6
35  D8      SE4   LOAD  D,A            1101 1000   23 6
36  2E            LOAD  H,*9           0010 1110   24 6
37  09                                  0000 1001   25 6
38  36            LOAD  L,*78          0011 0110   26 6
39  4E                                  0100 1110   27 6
40  C4            LOAD  A,E            1100 0100   28 6
41  80            ADD   A,A            1000 0000   29 6
42  80            ADD   A,A            1000 0000   2A 6
43  80            ADD   A,A            1000 0000   2B 6
44  F8            LOAD  (HL),A         1111 1000   2C 6
45  30            INR   L              0011 0000   2D 6
46  C4            LOAD  A,E            1100 0100   2E 6
47  0A            RR    A              0000 1010   2F 6
48  0A            RR    A              0000 1010   30 6
49  0A            RR    A              0000 1010   31 6
50  0A            RR    A              0000 1010   32 6
51  0A            RR    A              0000 1010   33 6
52  24            AND   A,*7           0010 0100   34 6
53  07                                  0000 0111   35 6
54  F8            LOAD  (HL),A         1111 1000   36 6
55  C3            LOAD  A,D            1100 0011   37 6
56  07            RET                  0000 0111   38 6
57  2E      SE6   LOAD  H,*9           0010 1110   39 6
58  09                                  0000 1001   3A 6
59  36            LOAD  L,*7           0011 0110   3B 6
60  07                                  0000 0111   3C 6
61  E7            LOAD  E,(HL)         1110 0111   3D 6
62  20            INR   E              0010 0000   3E 6
63  FC            LOAD  (HL),E         1111 1100   3F 6
64  29            DCR   H              0010 1001   40 6
65  F4            LOAD  L,E            1111 0100   41 6
66  F7            LOAD  L,(HL)         1111 0111   42 6
67  07            RET                  0000 0111   43 6
68  2E      SE9   LOAD  H,*9           0010 1110   44 6
69  09                                  0000 1001   45 6
70  36            LOAD  L,*72          0011 0110   46 6
71  48                                  0100 1000   47 6
72  F8            LOAD  (HL),A         1111 1000   48 6
73  30            INR   L              0011 0000   49 6
74  F9      SE9E  LOAD  (HL),B         1111 1001   4A 6
```

```
 75   30          INR  L          0011 0000   4B 6
 76   FA          LOAD (HL)•C     1111 1010   4C 6
 77   30          INR  L          0011 0000   4D 6
 78   FB          LOAD (HL)•D     1111 1011   4E 6
 79   07          RET             0000 0111   4F 6
 80   2E    SE10  LOAD H•*9       0010 1110   50 6
 81   09                          0000 1001   51 6
 82   36          LOAD L•*72      0011 0110   52 6
 83   48                          0100 1000   53 6
 84   C7          LOAD A•(HL)     1100 0111   54 6
 85   30          INR  L          0011 0000   55 6
 86   CF    SE10E LOAD B•(HL)     1100 1111   56 6
 87   30          INR  L          0011 0000   57 6
 88   D7          LOAD C•(HL)     1101 0111   58 6
 89   30          INR  L          0011 0000   59 6
 90   DF          LOAD D•(HL)     1101 1111   5A 6
 91   07          RET             0000 0111   5B 6
 92   2E    SE11  LOAD H•*9       0010 1110   5C 6
 93   09                          0000 1001   5D 6
 94   36          LOAD L•*64      0011 0110   5E 6
 95   40                          0100 0000   5F 6
 96   46          CALL SE9E       0100 0110   60 6
 97   4A                          0100 1010   61 6
 98   06                          0000 0110   62 6
 99   30          INR  L          0011 0000   63 6
100   FC          LOAD (HL)•E     1111 1100   64 6
101   07          RET             0000 0111   65 6
102   46    SA2   CALL SA5        0100 0110   66 6
103   41                          0100 0001   67 6
104   02                          0000 0010   68 6
105   D1          LOAD C•B        1101 0001   69 6
106   BA          COMP A•C        1011 1010   6A 6
107   28          RET  ZS         0010 1011   6B 6
108   46          CALL SE5        0100 0110   6C 6
109   AC                          1010 1100   6D 6
110   07                          0000 0111   6E 6
111   04          ADD  A•*0       0000 0100   6F 6
112   00                          0000 0000   70 6
113   CC          LOAD B•E        1100 1100   71 6
114   50          JUMP SC•SA2A    0101 0000   72 6
115   7C                          0111 1100   73 6
116   06                          0000 0110   74 6
117   F0          LOAD L•A        1111 0000   75 6
118   80          ADD  A•A        1000 0000   76 6
119   E8          LOAD H•A        1110 1000   77 6
120   C6          LOAD A•L        1100 0110   78 6
121   96          SUB  A•L        1001 0110   79 6
122   26          LOAD E•*10      0010 0110   7A 6
123   0A                          0000 1010   7B 6
124   D0    SA2A  LOAD C•A        1101 0000   7C 6
125   46          CALL SA10       0100 0110   7D 6
126   82                          1000 0010   7E 6
127   06                          0000 0110   7F 6
128   E1          LOAD E•B        1110 0001   80 6
129   07          RET             0000 0111   81 6
130   F4    SA10  LOAD L•E        1111 0100   82 6
131   2E          LOAD H•*9       0010 1110   83 6
132   09                          0000 1001   84 6
133   C7          LOAD A•(HL)     1100 0111   85 6
134   46          CALL SE3        0100 0110   86 6
135   72                          0111 0010   87 6
136   07                          0000 0111   88 6
137   FB          LOAD (HL)•D     1111 1011   89 6
```

```
138  1E   SA10ALOAD D,*0          0001 1110   8A 6
139  00                          0000 0000   8B 6
140  F3        LOAD L,D          1111 0011   8C 6
141  46   SA10BCALL SA6           0100 0110   8D 6
142  FA                          1111 1010   8E 6
143  00                          0000 0000   8F 6
144  46        CALL SE17         0100 0110   90 6
145  EA                          1110 1010   91 6
146  07                          0000 0111   92 6
147  F8   SA10CLOAD (HL),A       1111 1000   93 6
148  46        CALL SE7          0100 0110   94 6
149  A2                          1010 0010   95 6
150  04                          0000 0100   96 6
151  40        JUMP CC,SA10B     0100 0000   97 6
152  8D                          1000 1101   98 6
153  06                          0000 0110   99 6
154  11        DCR  C            0001 0001   9A 6
155  48        JUMP ZC,SA10A     0100 1000   9B 6
156  8A                          1000 1010   9C 6
157  06                          0000 0110   9D 6
158  07        RET               0000 0111   9E 6
159  F4   SA11 LOAD L,E          1111 0100   9F 6
160  2E        LOAD H,*9         0010 1110   A0 6
161  09                          0000 1001   A1 6
162  C7        LOAD A,(HL)       1100 0111   A2 6
163  46        CALL SE5          0100 0110   A3 6
164  AC                          1010 1100   A4 6
165  07                          0000 0111   A5 6
166  F8        LOAD (HL),A       1111 1000   A6 6
167  CA        LOAD B,C          1100 1010   A7 6
168  1E   SA11ALOAD D,*0          0001 1110   A8 6
169  00                          0000 0000   A9 6
170  F3        LOAD L,D          1111 0011   AA 6
171  46   SA11BCALL SA6           0100 0110   AB 6
172  FA                          1111 1010   AC 6
173  00                          0000 0000   AD 6
174  D0        LOAD C,A          1101 0000   AE 6
175  46        CALL SE3          0100 0110   AF 6
176  72                          0111 0010   B0 6
177  07                          0000 0111   B1 6
178  F8        LOAD (HL),A       1111 1000   B2 6
179  46        CALL SE7          0100 0110   B3 6
180  A2                          1010 0010   B4 6
181  04                          0000 0100   B5 6
182  40        JUMP CC,SA11B     0100 0000   B6 6
183  AB                          1010 1011   B7 6
184  06                          0000 0110   B8 6
185  09        DCR  B            0000 1001   B9 6
186  48        JUMP ZC,SA11A     0100 1000   BA 6
187  A8                          1010 1000   BB 6
188  06                          0000 0110   BC 6
189  07        RET               0000 0111   BD 6
190  2E   SA12ALOAD H,*9          0010 1110   BE 6
191  09                          0000 1001   BF 6
192  36        LOAD L,*8         0011 0110   C0 6
193  08                          0000 1000   C1 6
194  C7        LOAD A,(HL)       1100 0111   C2 6
195  84        ADD  A,E          1000 0100   C3 6
196  E8        LOAD H,A          1110 1000   C4 6
197  29        DCR  H            0010 1001   C5 6
198  36        LOAD L,*255       0011 0110   C6 6
199  FF                          1111 1111   C7 6
200  07        RET               0000 0111   C8 6
```

```
201  C7  SA12BLOAD A,(HL)      1100 0111  C9 6
202  FB       LOAD (HL),D      1111 1011  CA 6
203  D8       LOAD D,A         1101 1000  CB 6
204  46       CALL SE8         0100 0110  CC 6
205  F4                        1111 0100  CD 6
206  04                        0000 0100  CE 6
207  40       JUMP CC,SA12B    0100 0000  CF 6
208  C9                        1100 1001  D0 6
209  06                        0000 0110  D1 6
210  07       RET              0000 0111  D2 6
211  1E  SA13 LOAD D,#0        0001 1110  D3 6
212  00                        0000 0000  D4 6
213  F3       LOAD L,D         1111 0011  D5 6
214  46  SA13ACALL SA4         0100 0110  D6 6
215  F6                        1111 0110  D7 6
216  02                        0000 0010  D8 6
217  C8       LOAD B,A         1100 1000  D9 6
218  AA  SA13BXOR  A,C         1010 1010  DA 6
219  70       JUMP SS,SA13D    0111 0000  DB 6
220  F7                        1111 0111  DC 6
221  06                        0000 0110  DD 6
222  C1       LOAD A,B         1100 0001  DE 6
223  46       CALL SE5         0100 0110  DF 6
224  AC                        1010 1100  E0 6
225  07                        0000 0111  E1 6
226  A9  SA13FXOR  A,B         1010 1001  E2 6
227  70       JUMP SS,SA13C    0111 0000  E3 6
228  00                        0000 0000  E4 6
229  07                        0000 0111  E5 6
230  46       CALL SE11        0100 0110  E6 6
231  5C                        0101 1100  E7 6
232  06                        0000 0110  E8 6
233  26       LOAD E,#10       0010 0110  E9 6
234  0A                        0000 1010  EA 6
235  16       LOAD C,#1        0001 0110  EB 6
236  01                        0000 0001  EC 6
237  46       CALL SA10        0100 0110  ED 6
238  82                        1000 0010  EE 6
239  06                        0000 0110  EF 6
240  46       CALL SE12        0100 0110  F0 6
241  6E                        0110 1110  F1 6
242  04                        0000 0100  F2 6
243  C1       LOAD A,B         1100 0001  F3 6
244  44       JUMP SA13B       0100 0100  F4 6
245  DA                        1101 1010  F5 6
246  06                        0000 0110  F6 6
247  C1  SA13DLOAD A,B         1100 0001  F7 6
248  46       CALL SE3         0100 0110  F8 6
249  72                        0111 0010  F9 6
250  07                        0000 0111  FA 6
251  44       JUMP SA13F       0100 0100  FB 6
252  E2                        1110 0010  FC 6
253  06                        0000 0110  FD 6
254  C0       NOP              1100 0000  FE 6
255  C0       NOP              1100 0000  FF 6
```

```
SA7             2       185
SA6             0       250
SA5             2        65
SA4             2       246
SA13A           6       214
SE7             4       162
SE11            6        92
SE12            4       110
SE14            0       226
SA13C           7         0
SA14            7         7
SA14A           7        10
XQTM            9        77
SA15            7        24
SA16            7        36
SA17            7        51
SA17A           7        54
SA18            7        68
SA20            7        80
SA20A           7        87
SA21            7        98
SA21A           7       102
SE3             7       114
SE3F            7       131
SE3E            7       145
SE3H            7       146
SE3A            7       151
SE3G            7       156
SE5             7       172
SE5C            7       192
SE5B            7       198
S000            7       216
SE17            7       234
S999            7       248
```

```
    SA7   EQU  185,2
    SA6   EQU  250,0
    SA5   EQU  65,2
    SA4   EQU  246,2
    SA13AEQU  214,6
    SE7   EQU  162,4
    SE11  EQU  92,6
    SE12  EQU  110,4
    SE14  EQU  226,0
    SA13CDIR   0,7
```

```
 0   46          CALL SE7          0100 0110   00 7
 1   A2                            1010 0010   01 7
 2   04                            0000 0100   02 7
 3   40          JUMP CC,SA13A     0100 0000   03 7
 4   D6                            1101 0110   04 7
 5   06                            0000 0110   05 7
 6   07          RET               0000 0111   06 7
 7   41    SA14  INP  E0           0100 0001   07 7
 8   F8          LOAD (HL),A       1111 1000   08 7
 9   E1          LOAD E,B          1110 0001   09 7
10   21    SA14ADCR   E            0010 0001   0A 7
          XQTM EQU  77,9
11   48          JUMP ZC,SA14A     0100 1000   0B 7
12   0A                            0000 1010   0C 7
13   07                            0000 0111   0D 7
14   30          INR  L            0011 0000   0E 7
15   48          JUMP ZC,SA14      0100 1000   0F 7
16   07                            0000 0111   10 7
17   07                            0000 0111   11 7
```

```
18  28          INR  H          0010 1000  12 7
19  19          DCR  D          0001 1001  13 7
20  48          JUMP ZC,SA14    0100 1000  14 7
21  07                          0000 0111  15 7
22  07                          0000 0111  16 7
23  07          RET             0000 0111  17 7
24  41  SA15    INP  E0         0100 0001  18 7
25  F8          LOAD (HL),A     1111 1000  19 7
26  30          INR  L          0011 0000  1A 7
27  48          JUMP ZC,SA15    0100 1000  1B 7
28  18                          0001 1000  1C 7
29  07                          0000 0111  1D 7
30  28          INR  H          0010 1000  1E 7
31  19          DCR  D          0001 1001  1F 7
32  48          JUMP ZC,SA15    0100 1000  20 7
33  18                          0001 1000  21 7
34  07                          0000 0111  22 7
35  07          RET             0000 0111  23 7
36  2E  SA16    LOAD H,*9       0010 1110  24 7
37  09                          0000 1001  25 7
38  36          LOAD L,*8       0011 0110  26 7
39  08                          0000 1000  27 7
40  DF          LOAD D,(HL)     1101 1111  28 7
41  19          DCR  D          0001 1001  29 7
42  30          INR  L          0011 0000  2A 7
43  CF          LOAD B,(HL)     1100 1111  2B 7
44  36          LOAD L,*0       0011 0110  2C 7
45  00                          0000 0000  2D 7
46  C6          LOAD A,L        1100 0110  2E 7
47  81          ADD  B          1000 0001  2F 7
48  41          INP  E0         0100 0001  30 7
49  EC          LOAD H,E        1110 1100  31 7
50  07          RET             0000 0111  32 7
51  C7  SA17    LOAD A,(HL)     1100 0111  33 7
52  71          OUT  S0         0111 0001  34 7
53  E1          LOAD E,B        1110 0001  35 7
54  21  SA17ADCR  E            0010 0001  36 7
55  48          JUMP ZC,SA17A   0100 1000  37 7
56  36                          0011 0110  38 7
57  07                          0000 0111  39 7
58  30          INR  L          0011 0000  3A 7
59  48          JUMP ZC,SA17    0100 1000  3B 7
60  33                          0011 0011  3C 7
61  07                          0000 0111  3D 7
62  28          INR  H          0010 1000  3E 7
63  19          DCR  D          0001 1001  3F 7
64  48          JUMP ZC,SA17    0100 1000  40 7
65  33                          0011 0011  41 7
66  07                          0000 0111  42 7
67  07          RET             0000 0111  43 7
68  C7  SA18    LOAD A,(HL)     1100 0111  44 7
69  71          OUT  S0         0111 0001  45 7
70  30          INR  L          0011 0000  46 7
71  48          JUMP ZC,SA18    0100 1000  47 7
72  44                          0100 0100  48 7
73  07                          0000 0111  49 7
74  28          INR  H          0010 1000  4A 7
75  19          DCR  D          0001 1001  4B 7
76  48          JUMP ZC,SA18    0100 1000  4C 7
77  44                          0100 0100  4D 7
78  07                          0000 0111  4E 7
79  07          RET             0000 0111  4F 7
80  46  SA20    CALL SA5        0100 0110  50 7
```

```
  81    41                          0100 0001    51 7
  82    02                          0000 0010    52 7
  83    F9        LOAD (HL),B       1111 1001    53 7
  84    1E        LOAD D,*0         0001 1110    54 7
  85    00                          0000 0000    55 7
  86    F3        LOAD L,D          1111 0011    56 7
  87    46  SA20  CALL SA4          0100 0110    57 7
  88    F6                          1111 0110    58 7
  89    02                          0000 0010    59 7
  90    FA        LOAD (HL),C       1111 1010    5A 7
  91    46        CALL SE7          0100 0110    5B 7
  92    A2                          1010 0010    5C 7
  93    04                          0000 0100    5D 7
  94    40        JUMP CC,SA20A     0100 0000    5E 7
  95    57                          0101 0111    5F 7
  96    07                          0000 0111    60 7
  97    07        RET               0000 0111    61 7
  98    1E  SA21  LOAD D,*0         0001 1110    62 7
  99    00                          0000 0000    63 7
 100    F3        LOAD L,D          1111 0011    64 7
 101    CB        LOAD B,D          1100 1011    65 7
 102    46  SA21A CALL SA6          0100 0110    66 7
 103    FA                          1111 1010    67 7
 104    00                          0000 0000    68 7
 105    F9        LOAD (HL),B       1111 1001    69 7
 106    C8        LOAD B,A          1100 1000    6A 7
 107    46        CALL SE7          0100 0110    6B 7
 108    A2                          1010 0010    6C 7
 109    04                          0000 0100    6D 7
 110    40        JUMP CC,SA21A     0100 0000    6E 7
 111    66                          0110 0110    6F 7
 112    07                          0000 0111    70 7
 113    07        RET               0000 0111    71 7
 114    46  SE3   CALL SE11         0100 0110    72 7
 115    5C                          0101 1100    73 7
 116    06                          0000 0110    74 7
 117    C8        LOAD B,A          1100 1000    75 7
 118    C2        LOAD A,C          1100 0010    76 7
 119    26        LOAD E,*0         0010 0110    77 7
 120    00                          0000 0000    78 7
 121    84        ADD  A,E          1000 0100    79 7
 122    C1        LOAD A,B          1100 0001    7A 7
 123    70        JUMP SS,SE3A      0111 0000    7B 7
 124    97                          1001 0111    7C 7
 125    07                          0000 0111    7D 7
 126    84        ADD  A,E          1000 0100    7E 7
 127    70        JUMP SS,SE3E      0111 0000    7F 7
 128    91                          1001 0001    80 7
 129    07                          0000 0111    81 7
 130    82        ADD  A,C          1000 0010    82 7
 131    50  SE3F  JUMP SC,SE3H      0101 0000    83 7
 132    92                          1001 0010    84 7
 133    07                          0000 0111    85 7
 134    46        CALL SE14         0100 0110    86 7
 135    E2                          1110 0010    87 7
 136    00                          0000 0000    88 7
 137    06        LOAD A,*255       0000 0110    89 7
 138    FF                          1111 1111    8A 7
 139    80        ADD  A            1000 0000    8B 7
 140    06        LOAD A,*127       0000 0110    8C 7
 141    7F                          0111 1111    8D 7
 142    44        JUMP SE12         0100 0100    8E 7
 143    6E                          0110 1110    8F 7
```

```
144   04                                  0000 0100   90 7
145   82    SE3E ADD   A,C                1000 0010   91 7
146   04    SE3H ADD   A,*0               0000 0100   92 7
147   00                                  0000 0000   93 7
148   44         JUMP  SE12               0100 0100   94 7
149   6E                                  0110 1110   95 7
150   04                                  0000 0100   96 7
151   84    SE3A ADD   A,E                1000 0100   97 7
152   50         JUMP  SC,SE3E            0101 0000   98 7
153   91                                  1001 0001   99 7
154   07                                  0000 0111   9A 7
155   82         ADD   A,C                1000 0010   9B 7
156   70    SE3G JUMP  SS,SE3H            0111 0000   9C 7
157   92                                  1001 0010   9D 7
158   07                                  0000 0111   9E 7
159   46         CALL  SE14               0100 0110   9F 7
160   E2                                  1110 0010   A0 7
161   00                                  0000 0000   A1 7
162   06         LOAD  A,*-128            0000 0110   A2 7
163   80                                  1000 0000   A3 7
164   44         JUMP  SE3H               0100 0100   A4 7
165   92                                  1001 0010   A5 7
166   07                                  0000 0111   A6 7
167   **                                  **** ****   A7 7
168   **                                  **** ****   A8 7
169   **                                  **** ****   A9 7
170   **                                  **** ****   AA 7
171   **                                  **** ****   AB 7
              SE5  DIR   172,7
172   46         CALL  SE11               0100 0110   AC 7
173   5C                                  0101 1100   AD 7
174   06                                  0000 0110   AE 7
175   C8         LOAD  B,A                1100 1000   AF 7
176   C2         LOAD  A,C                1100 0010   B0 7
177   26         LOAD  E,*0               0010 0110   B1 7
178   00                                  0000 0000   B2 7
179   84         ADD   A,E                1000 0100   B3 7
180   C1         LOAD  A,B                1100 0001   B4 7
181   50         JUMP  SC,SE5B            0101 0000   B5 7
182   C6                                  1100 0110   B6 7
183   07                                  0000 0111   B7 7
184   84         ADD   E                  1000 0100   B8 7
185   70         JUMP  SS,SE5C            0111 0000   B9 7
186   C0                                  1100 0000   BA 7
187   07                                  0000 0111   BB 7
188   92         SUB   A,C                1001 0010   BC 7
189   44         JUMP  SE3F               0100 0100   BD 7
190   83                                  1000 0011   BE 7
191   07                                  0000 0111   BF 7
192   92    SE5C SUB   A,C                1001 0010   C0 7
193   04         ADD   A,*0               0000 0100   C1 7
194   00                                  0000 0000   C2 7
195   44         JUMP  SE12               0100 0100   C3 7
196   6E                                  0110 1110   C4 7
197   04                                  0000 0100   C5 7
198   84    SE5B ADD   E                  1000 0100   C6 7
199   50         JUMP  SC,SE5C            0101 0000   C7 7
200   C0                                  1100 0000   C8 7
201   07                                  0000 0111   C9 7
202   92         SUB   C                  1001 0010   CA 7
203   44         JUMP  SE3G               0100 0100   CB 7
204   9C                                  1001 1100   CC 7
205   07                                  0000 0111   CD 7
```

```
206   **                                  **** ****   CE 7
207   **                                  **** ****   CF 7
208   **                                  **** ****   D0 7
209   **                                  **** ****   D1 7
210   **                                  **** ****   D2 7
211   **                                  **** ****   D3 7
212   **                                  **** ****   D4 7
213   **                                  **** ****   D5 7
214   **                                  **** ****   D6 7
215   **                                  **** ****   D7 7
                S000 DTR  216,7
216   46        CALL SA7                  0100 0110   D8 7
217   B9                                  1011 1001   D9 7
218   02                                  0000 0010   DA 7
219   E6        LOAD E,L                  1110 0110   DB 7
220   46        CALL SA7                  0100 0110   DC 7
221   B9                                  1011 1001   DD 7
222   02                                  0000 0010   DE 7
223   DE        LOAD D,L                  1101 1110   DF 7
224   2E        LOAD H,*9                 0010 1110   E0 7
225   09                                  0000 1001   E1 7
226   36        LOAD L,*78                0011 0110   E2 7
227   4E                                  0100 1110   E3 7
228   FC        LOAD (HL),E               1111 1100   E4 7
229   30        INR  L                    0011 0000   E5 7
230   FB        LOAD (HL),D               1111 1011   E6 7
231   44        JUMP XQTM                 0100 0100   E7 7
232   4D                                  0100 1101   E8 7
233   09                                  0000 1001   E9 7
234   04  SE17  ADD  A,*0                 0000 0100   EA 7
235   00                                  0000 0000   EB 7
236   1A        RRC  A                    0001 1010   EC 7
237   34        OR   A,*1                 0011 0100   ED 7
238   01                                  0000 0001   EE 7
239   13        RET  SC                   0001 0011   EF 7
240   34        OR   A,*128               0011 0100   F0 7
241   80                                  1000 0000   F1 7
242   07        RET                       0000 0111   F2 7
243   **                                  **** ****   F3 7
244   **                                  **** ****   F4 7
245   **                                  **** ****   F5 7
246   **                                  **** ****   F6 7
247   **                                  **** ****   F7 7
                S999 DTR  248,7
248   06        LOAD A,*255               0000 0110   F8 7
249   FF                                  1111 1111   F9 7
250   73        OUT  S1                   0111 0011   FA 7
251   00        HLT                       0000 0000   FB 7
252   44        JUMP S999                 0100 0100   FC 7
253   F8                                  1111 1000   FD 7
254   07                                  0000 0111   FE 7
```

APENDICE IV

DIRECTORIO DE PUNTOS DE ENTRADA DEL SISTEMA OPERATIVO

decim.	nº interior	nº exte.	
0		PM0	*
8		PM1	*
16		PM2	*
24		PM3	*
32		PM4	*
40		PM5	*
56		PM7	*
64	PM7B		
77	PM7A		
97	S1PM0		
100	PM0A		
108	S1PM1		
120	PM1A		
130	PM1B		
134	S2PM2		
136	PM2A		
144	S1PM3		
148	PM3A		
158	S1PM5		
169	SM1		*
185	SM2		*
191	SM3		*
199	SM4		*
205	SA12		*
224		SO94	*
226	SE14		*
232		SO95	*
234	SE13		*
240		SO96	*
248		SO97	*
250	SA6		*

CON * SE INDICAN LOS PUNTOS DE ENTRADA O DIRECCIONES DE COMIENZO DE RUTINAS.

decim.	nº inter.	nº ext	
0		S101	*
9	S101A		
28	S101Z		*
38	S101A		
57	SE15		*
64		S102	*
73	S102A		
89	S1012Z		*
99	S02ZA		
120		S103	*
134	S103A		
152		S104	*
169	S104A		
184		S105	*
187	S105A		
192	S105B		
213	S105D		
224		S106	*
227	S106A		
252	S106B		

decim.	nº inter.	n.exte.	
39	S106F		
48		S107	*
51	S107A		
65	SA5		*
72		S108	*
78	S108A		
96		S109	*
99	S109A		
112		S110	*
115	S110A		
128		S111	*
136	S111A		
144	S111B		
150	S112C		
160		S112	*
168	S112A		
176	S112B		
185	SA7		*
192		S121	*
208		S122	*
219	S122A		
240		S123	*
246	SA4		*

decim.	nº inter.	nºexter.	
0		S124	*
16		S125	*
23	S125A		
40		S126	*
48		S127	*
56		S130	*
58	S130E		
72		S132	*
80		S131	*
88		S133	*
96		S134	*
104		S135	*
112		S136	*
120		S137	*
128		S150	*
131	S150E		
152		S151	*
160		S152	*
163	S152E		
184		S153	*
191	SA1		*
194	SA1A		
197	SA1B		
220	SA3		*
231	SA3E		
248		S120	*

decim.	nº inte.	n.exte	
0		SO51	*
8		SO83	*
9	SO83E		
16		SO85	*
17	SO85E		
24		SO87	*
25	SO87E		
32		SO55	*
40		SO84	*
48		SO86	*
56		SO88	*
64		SO59	*
72		SO91	*
80		SO93	*
88		SO81	*
89	SO81E		
96		SO63	*
100	SO63E		
104		SO92	*
110	SE12		*
120		SO82	*
136		SO52	*
144		SO53	*
152		SO54	*
160		SO56	*
162	SE7		*
176		SO57	*
184		SO58	*
192		SO60	*
200		SO61	*
202	SA9		*
207	SA9A		
216		SO62	*
224		SO64	*
232		SO65	*
240		SO66	*
244	SE8		*

decim.	nºinter.	n.exte	
0		S001	*
2	S001A		
8		S002	*
11	S002A		
16		S003	*
24		S022	*
32		S004	*
34	S004A		
40		S005	*
43	S005A		
48		S006	*
56		S025	*
64		S007	*
67	ES007		*
72		S008	*
80		S009	*
88		S023	*
90	ES011		*
96		S010	*
99	ES010		*
104		S011	*
112		S012	*
120		S026	*
128	S013		*
136		S014	*
144		S015	*
160		S016	*
168		S017	*
176		S018	*
185	ES008		*
192		S019	*
200		S020	*
208		S021	*
211	SE1		*
228	SE1A		
245	SE1B		

decim	n.inter	n.exter	
0	SE2		*
13	SE2A		
22	SE2B		
35	SE4		*
57	SE6		*
68	SE9		*
74	SE9E		
80	SE10		*
86	SE10E		
92	SE11		*
102	SA2		*
124	SA2A		
130	SA10		*
138	SA10A		
141	SA10B		
147	SA10C		
159	SA11		*
168	SA11A		
171	SA11B		
190	SA12A		
201	SA12B		
211	SA13		*
214	SA13A		
218	SA13B		
226	SA13F		
247	SA13D		

decim.	nºinter.	n.exte	
0	SA13C		
7	SA14		*
10	SA14A		
24	SA15		*
36	SA16		*
51	SA17		*
54	SA17A		
68	SA18		*
80	SA20		*
87	SA20A		
98	SA21		*
102	SA21A		
114	SE3		*
131	SE3F		*
145	SE3E		
146	SE3H		
151	SE3A		
156	SE3G		*
172	SE5		*
192	SE5C		
198	SE5B		
216		S000	*
234	SE17		*
248		S999	*

APENDICE V

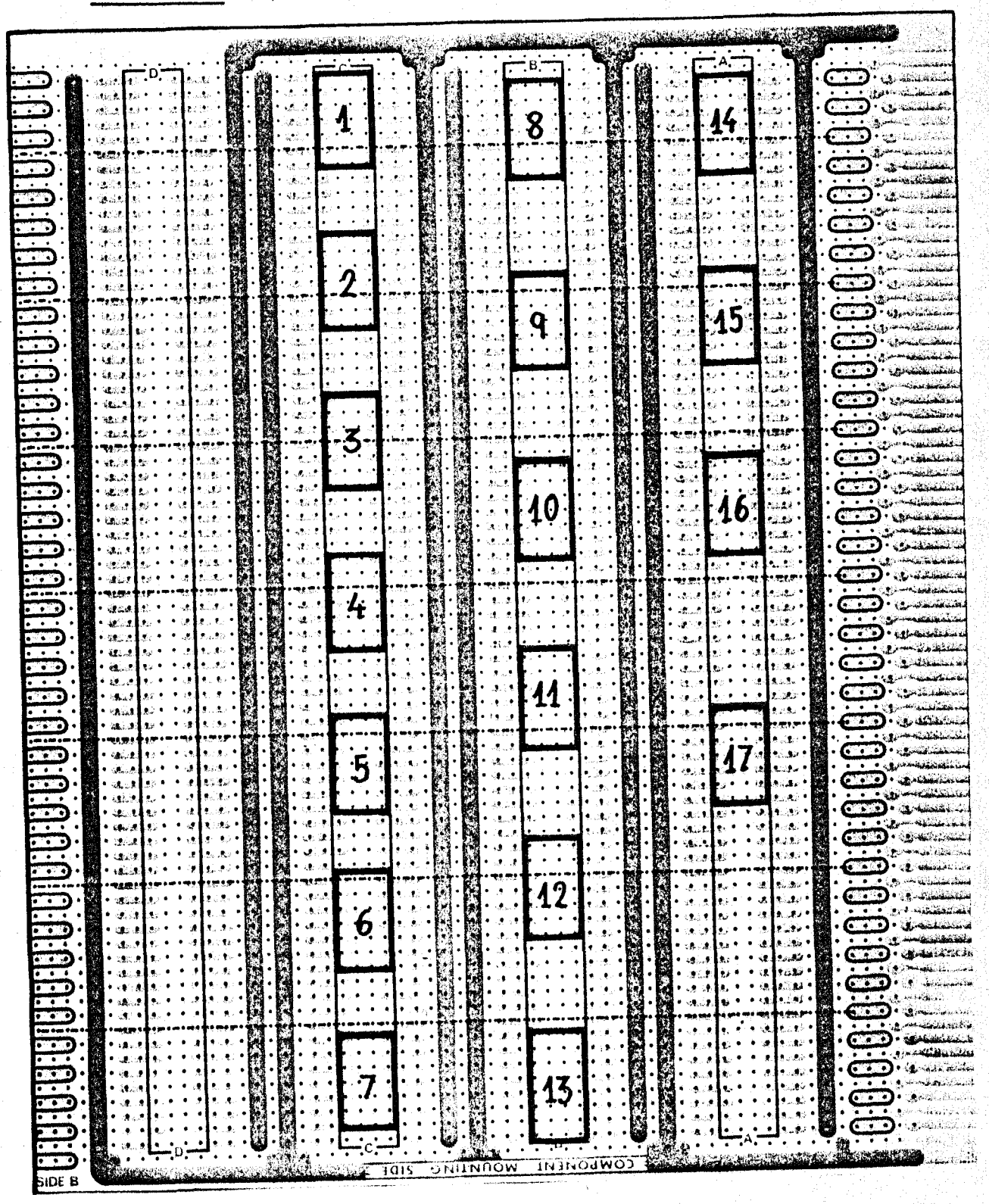
ORGANIZACION Y METODOLOGIA DE MONTAJE.

En este apéndice se indican los componentes utilizados en cada tarjeta del microordenador, y las principales señales de entrada y salida que se dan a través de los conectores para la interconexión de las distintas partes del dispositivo.

POSICION DE COMPONENTES

TARJETA Nº P

CONTENIDO: Circuitos de control de consola.



OBSERVACIONES:

COMPONENETES UTILIZADOS

TARJETA Nº P

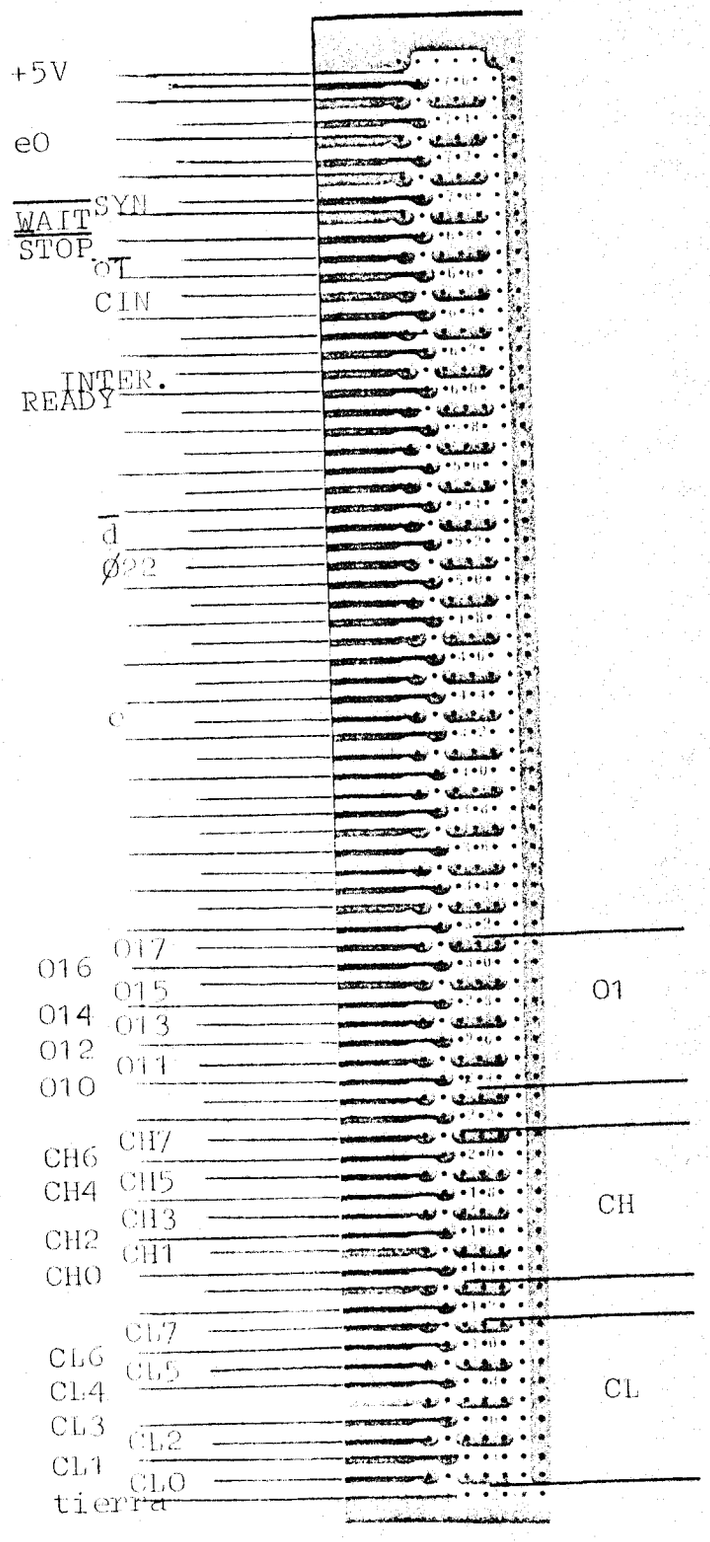
CONTENIDO: Circuitos de control de consola. Lógica de interrupción y espera.

Nº	TIPO		FUNCION
1	7400N	4	NAND
2	7400N	4	NAND
3	7405N	6	inversores colector abierto
4	7405N	6	inversores colector abierto
5	7405N	6	inversores colector abierto
6	7405N	6	inversores colector abierto
7	7405N	6	inversores colector abierto
8	74121N	1	monostable
9	7420N	2	NAND cuatro entradas
10	7472N	1	JK master-slave
11	74175N	4	báscula D
12	7400N	4	NAND
13	7404N	6	inversores
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			

OBSERVACIONES:

TARJETA P

CONTENIDO: Control de panel

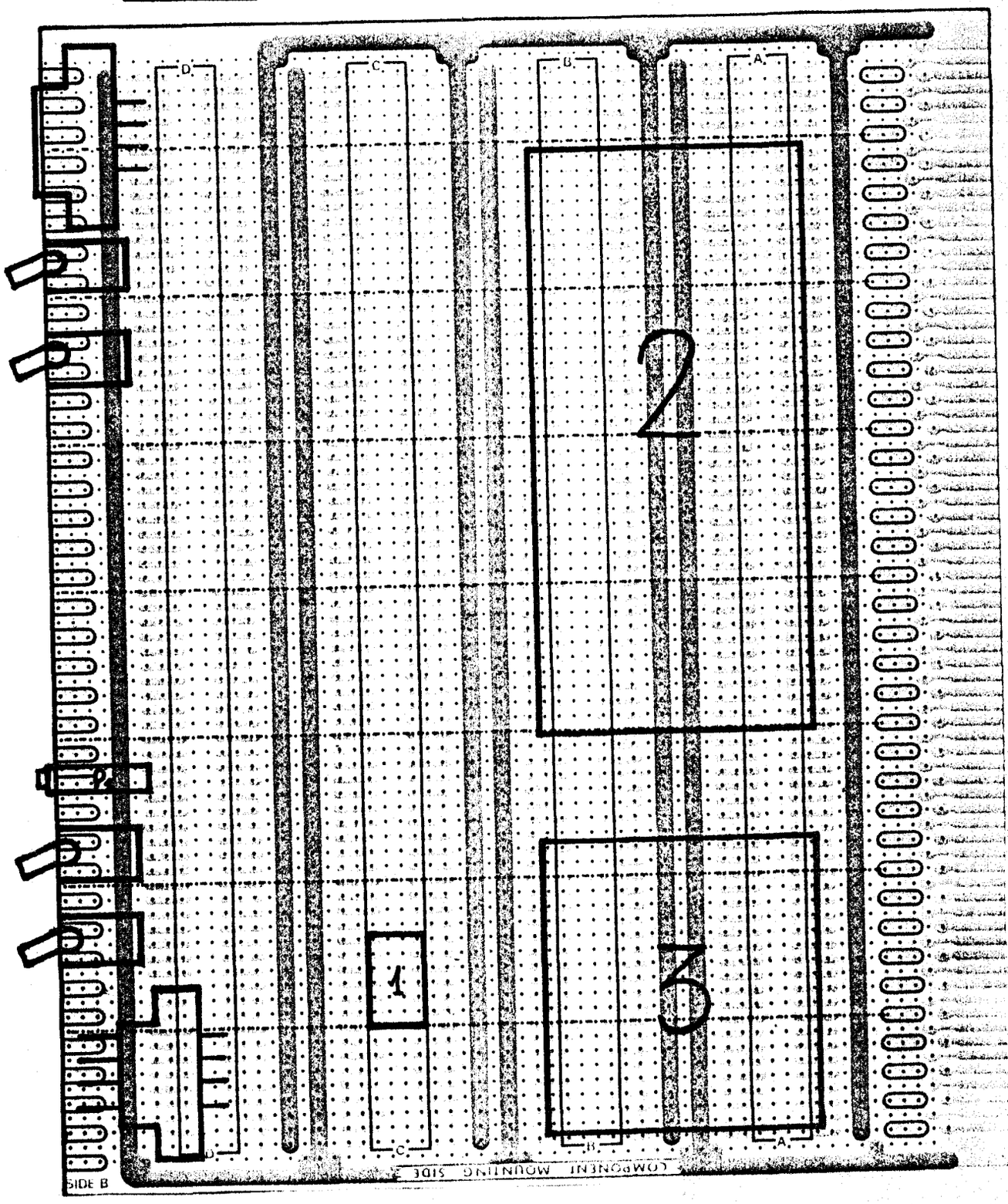


OBSERVACIONES:

P O S I C I O N D E C O M P O N E N T E S

TARJETA Nº C

CONTENIDO: Conversores A/D y D/A



OBSERVACIONES:

COMPONENETES UTILIZADOS

TARJETA Nº C

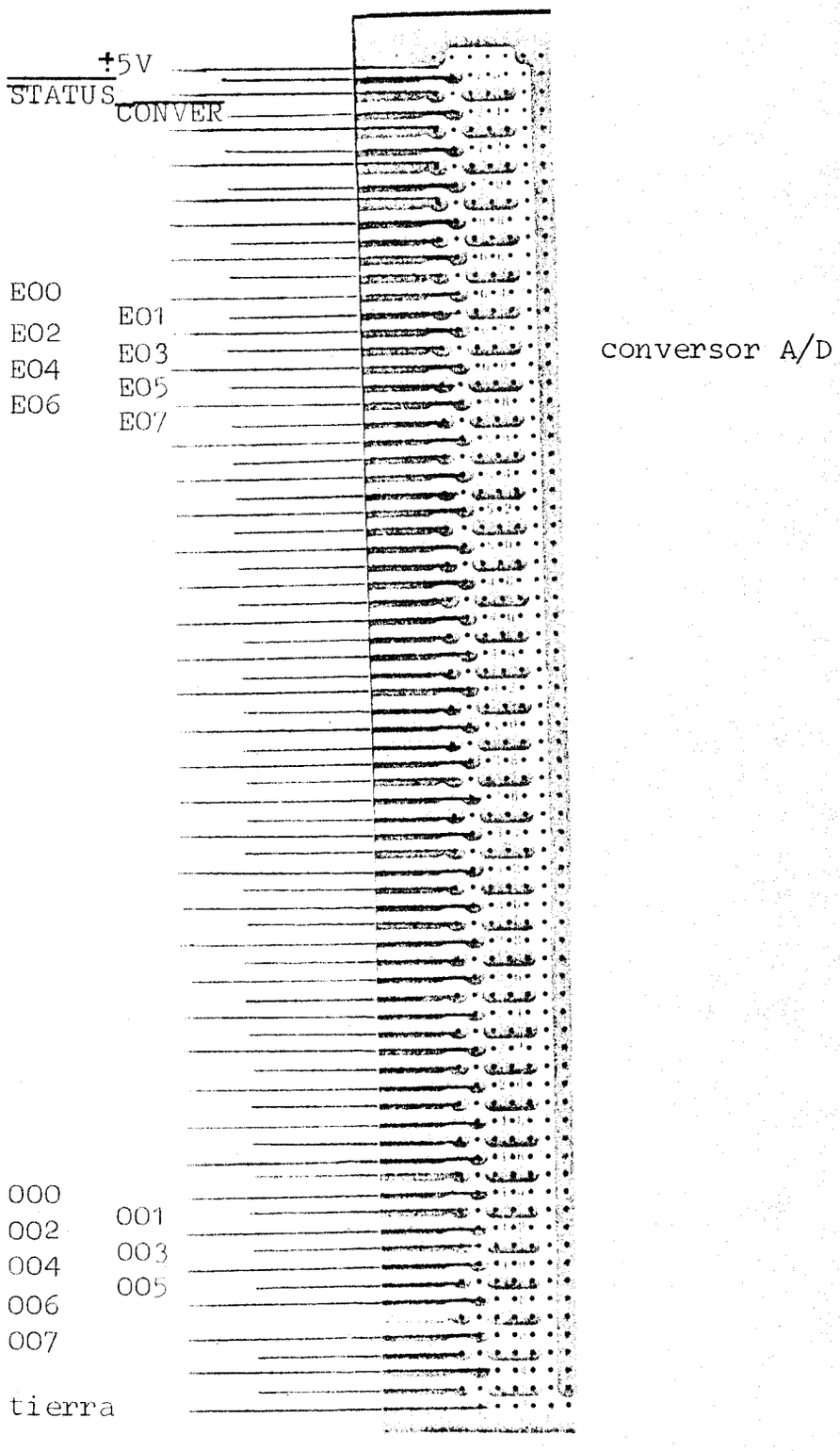
CONTENIDO: conversores A/D y D/A

Nº	TIPO	FUNCION
1	7404N	6 inversores
2	4114	1 conversor analógico/digital
3	4021	1 conversor digital/analógico
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		

OBSERVACIONES:

TARJETA C

CONTENIDO: conversores A/D y D/A

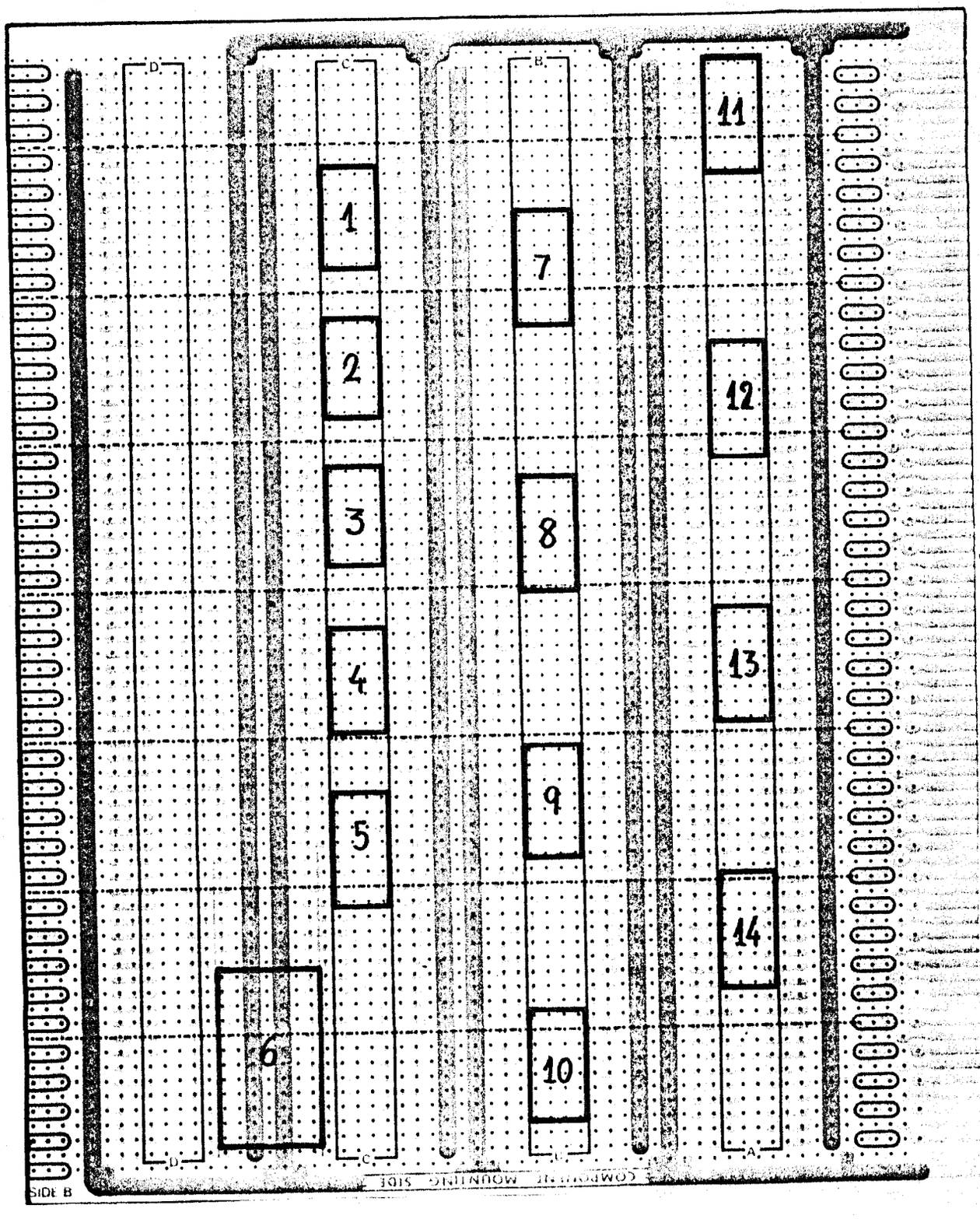


OBSERVACIONES:

P O S I C I O N D E C O M P O N E N T E S

TARJETA Nº S

CONTENIDO: Circuitos de control de salidas. Registros de salidas.



OBSERVACIONES:

COMPONENETES UTILIZADOS

TARJETA Nº S

CONTENIDO: control de salidas

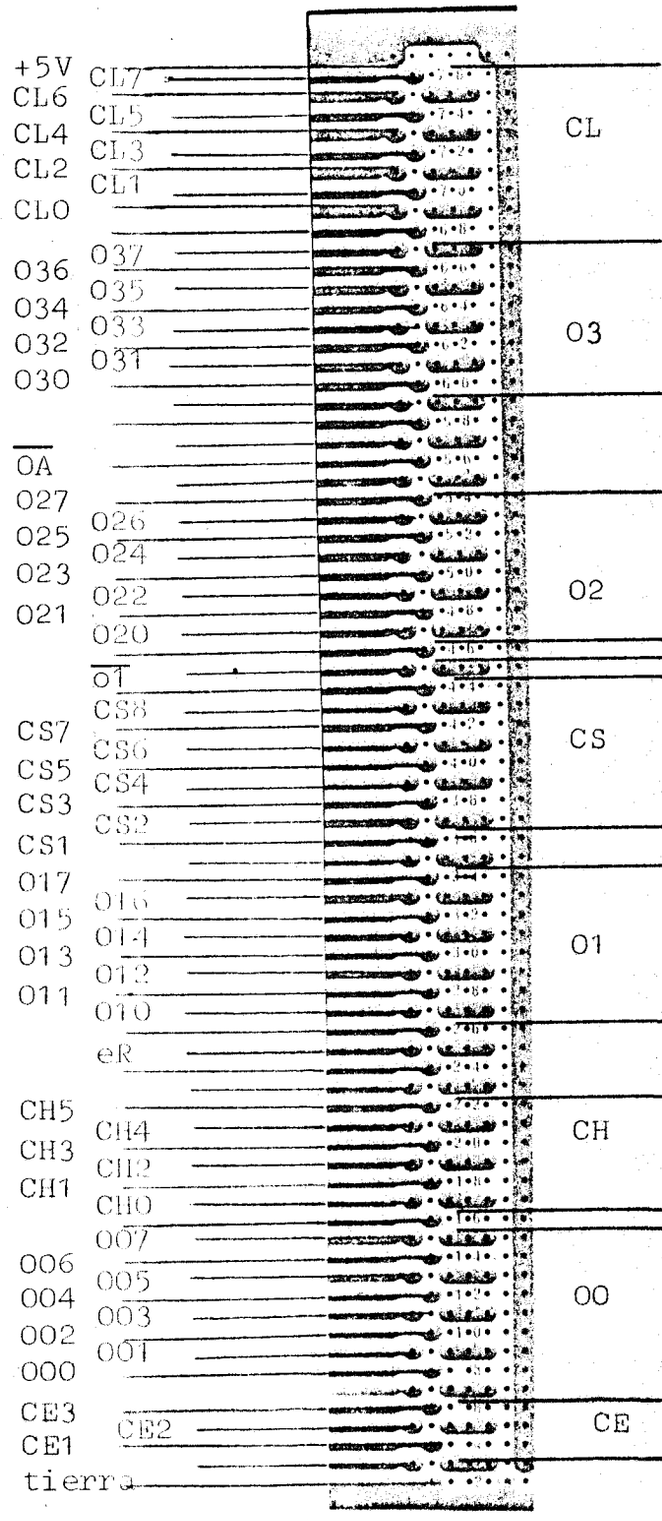
Nº	TIPO	FUNCION	
1	74154N	1	decodificador 4:16
2	7410N	3	NAND tres entradas
3	7404N	6	inversores
4	7410N	3	NAND tres entradas
5	7408N	4	AND
6	74155N	1	decodificador 3:8
7	7475N	2	latch
8	7475N	2	latch
9	7475N	2	latch
10	7475N	2	latch
11	7475N	2	latch
12	7475N	2	latch
13	7475N	2	latch
14	7475N	2	latch
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			

OBSERVACIONES:

ENTRADAS Y SALIDAS

TARJETA S

CONTENIDO: salidas.

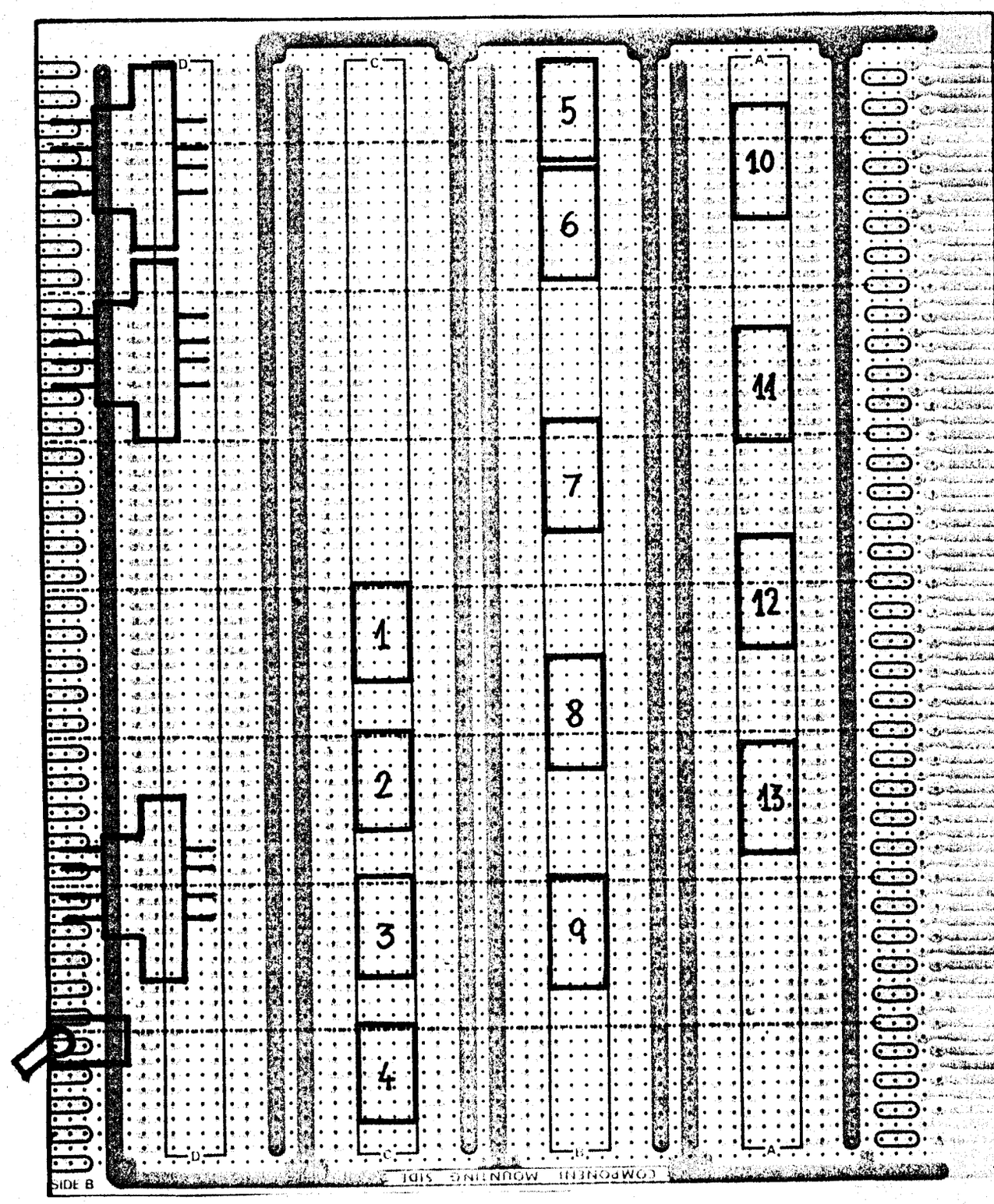


OBSERVACIONES:

P O S I C I O N D E C O M P O N E N T E S

TARJETA Nº E

CONTENIDO: control de entradas



OBSERVACIONES:

COMPONENETES UTILIZADOS

TARJETA Nº E

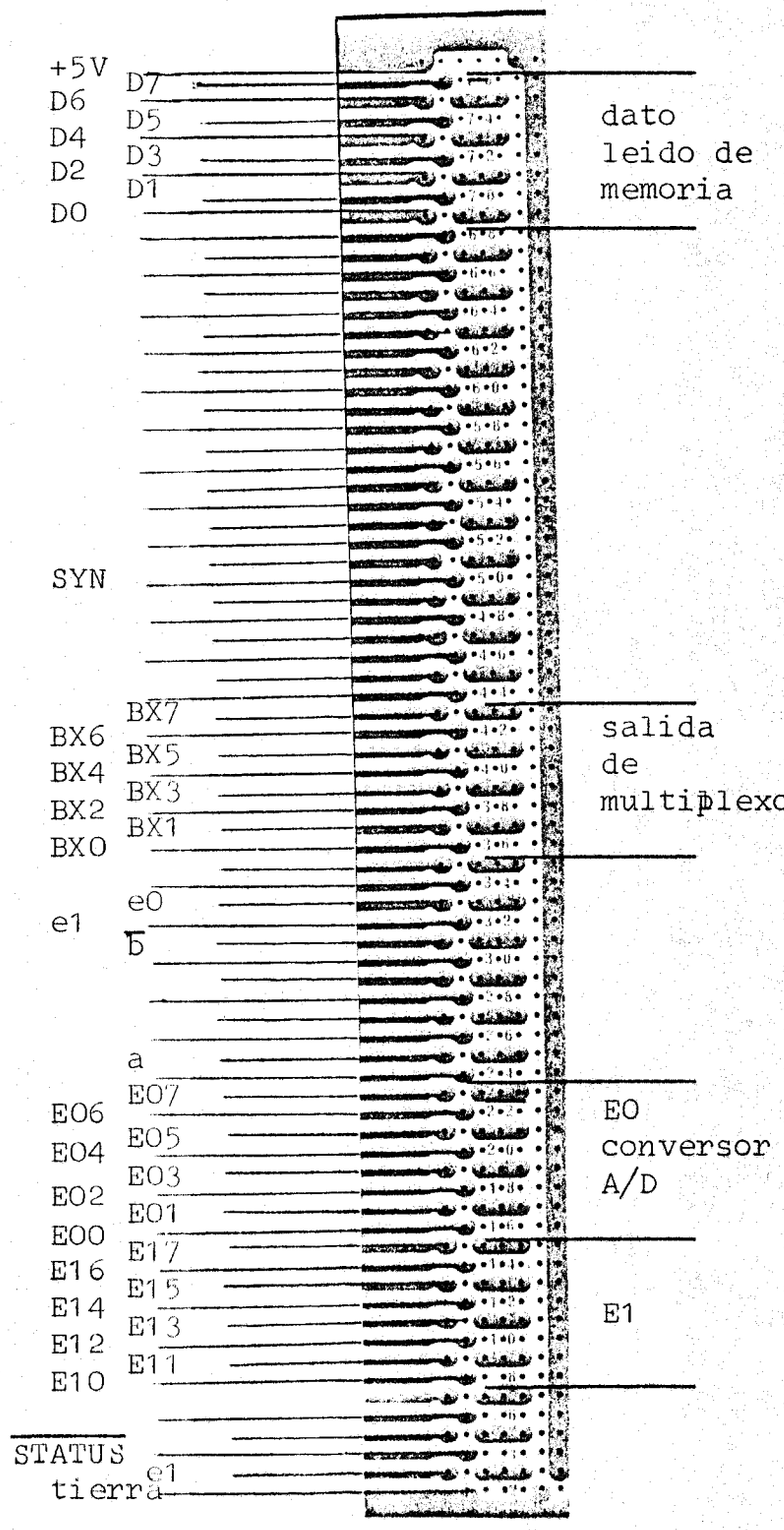
CONTENIDO: Control de entradas

Nº	TIPO	FUNCION	
1	7486N	4	exclusive-OR
2	7404N	6	inversores
3	7400J	4	NAND
4	7486N	4	exclusive OR
5	7475N	2	latch
6	7475N	2	latch
7	7475N	2	latch
8	7475N	2	latch
9	74153N	2	multiplexos 4:1
10	74153N	2	multiplexos 4:1
11	74153N	2	multiplexos 4:1
12	74153N	2	multiplexos 4:1
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			

OBSERVACIONES:

TARJETA E

CONTENIDO: Control de entradas, multiplexo.

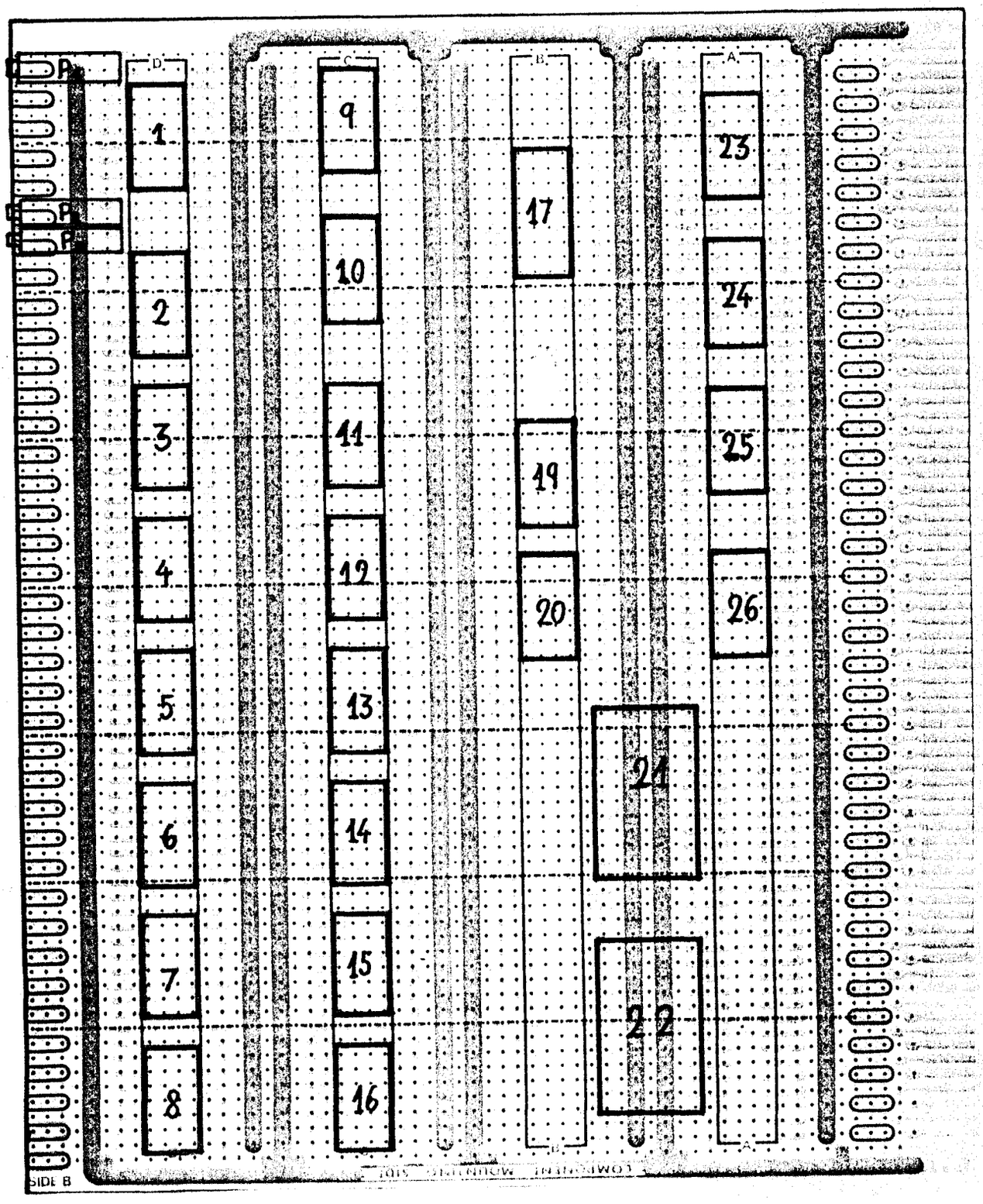


OBSERVACIONES:

POSICION DE COMPONENTES

TARJETA Nº U

CONTENIDO: Unidades aritmético-logica y de control



OBSERVACIONES:

potenciometro P1: ajuste de desfase entre Ø1 y Ø2
potenciometro P2: ajuste de anchura de Ø1
potenciometro P3: ajuste de anchura de Ø2.

COMPONENETES UTILIZADOS

TARJETA Nº U

CONTENIDO: Unidad aritmetico-lógica y unidad de control

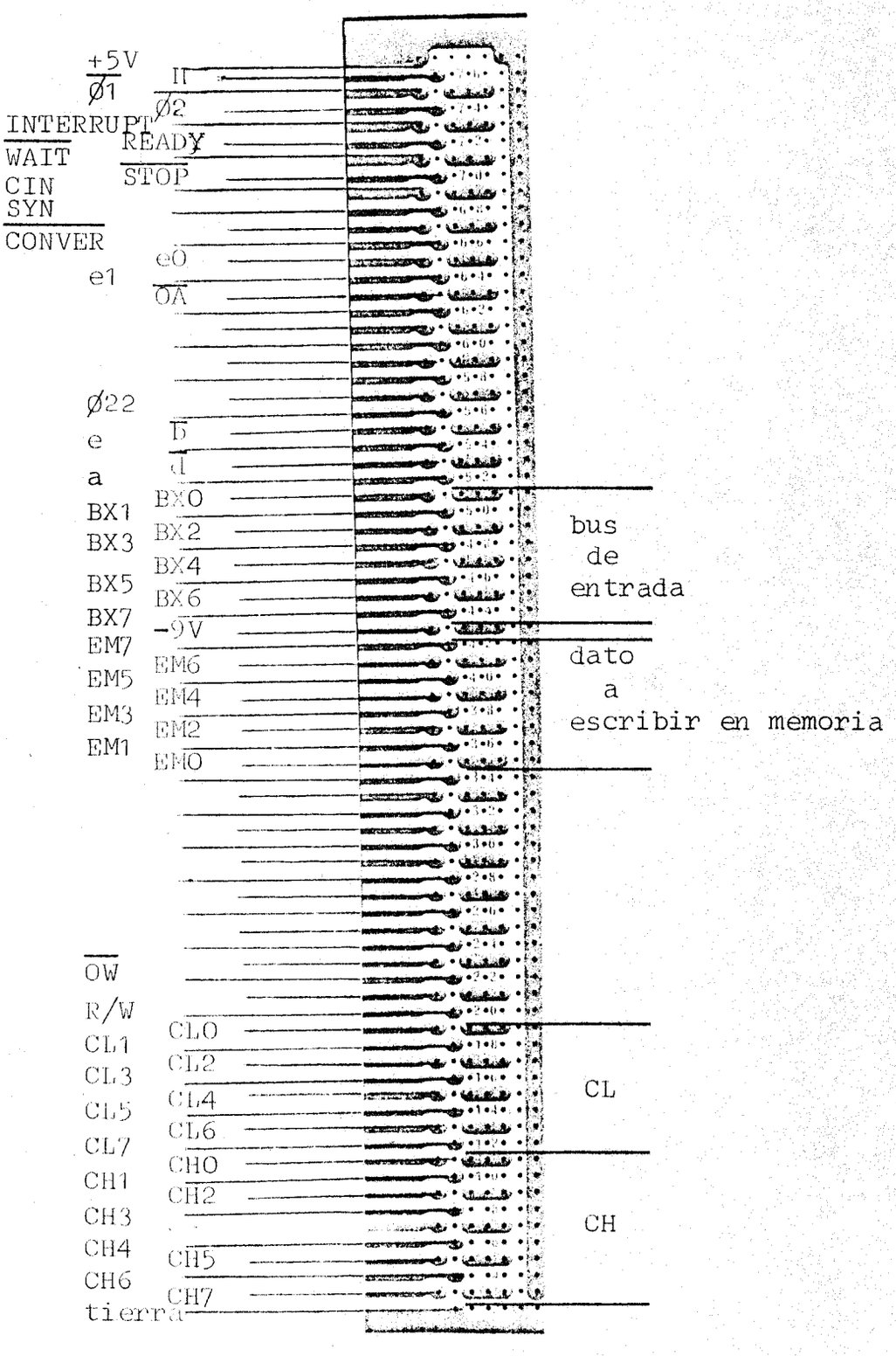
Nº	TIPO	FUNCION	
1	7400J	4	nand
2	74121N	1	monostable
3	7402N	4	NOR
4	7400H	4	NAND
5	7473N	2	JK master-slave
6	7474N	2	basculas D
7	7402N	4	NOR
8	74121N	1	Monostable
9	74121N	1	monostable
10	74L00N	4	NAND low power
11	74155N	1	decodificador 3:8
12	7402N	4	NOR
13	7404N	6	inversores
14	7400N	4	NAND
15	7410J	3	NAND tres entradas
16	C8008	1	microprocesador
17	74L00N	4	NAND low power
18	74L00N	4	NAND low power
19	74198N	1	registro 8 bits
20	74198N	1	registro 8 bits
21	74125N	4	buffer tres estados
22	74125N	4	buffer tres estados
23	7404N	6	inversores
24	7404N	6	inversores
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			

OBSERVACIONES: Las pastillas van numeradas de arriba a abajo, izquierda a derecha.

ENTRADAS Y SALIDAS

TARJETA U

CONTENIDO: Unidad aritmetico-lógica y de control

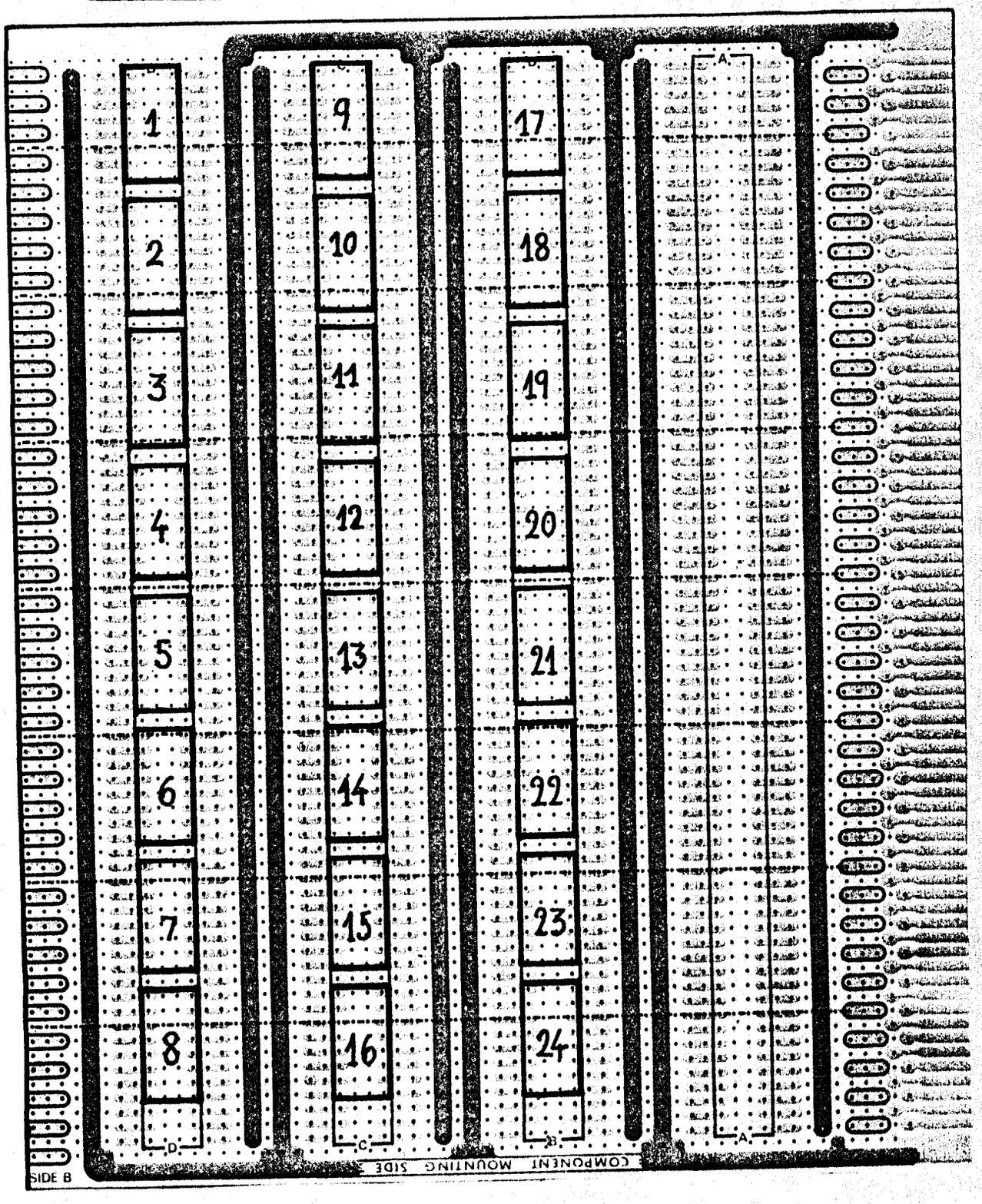


OBSERVACIONES:

P O S I C I O N D E C O M P O N E N T E S

TARJETA Nº A

CONTENIDO: .memoria RAM



OBSERVACIONES:

COMPONENETES UTILIZADOS

TARJETA Nº A

CONTENIDO: Memoria RAM

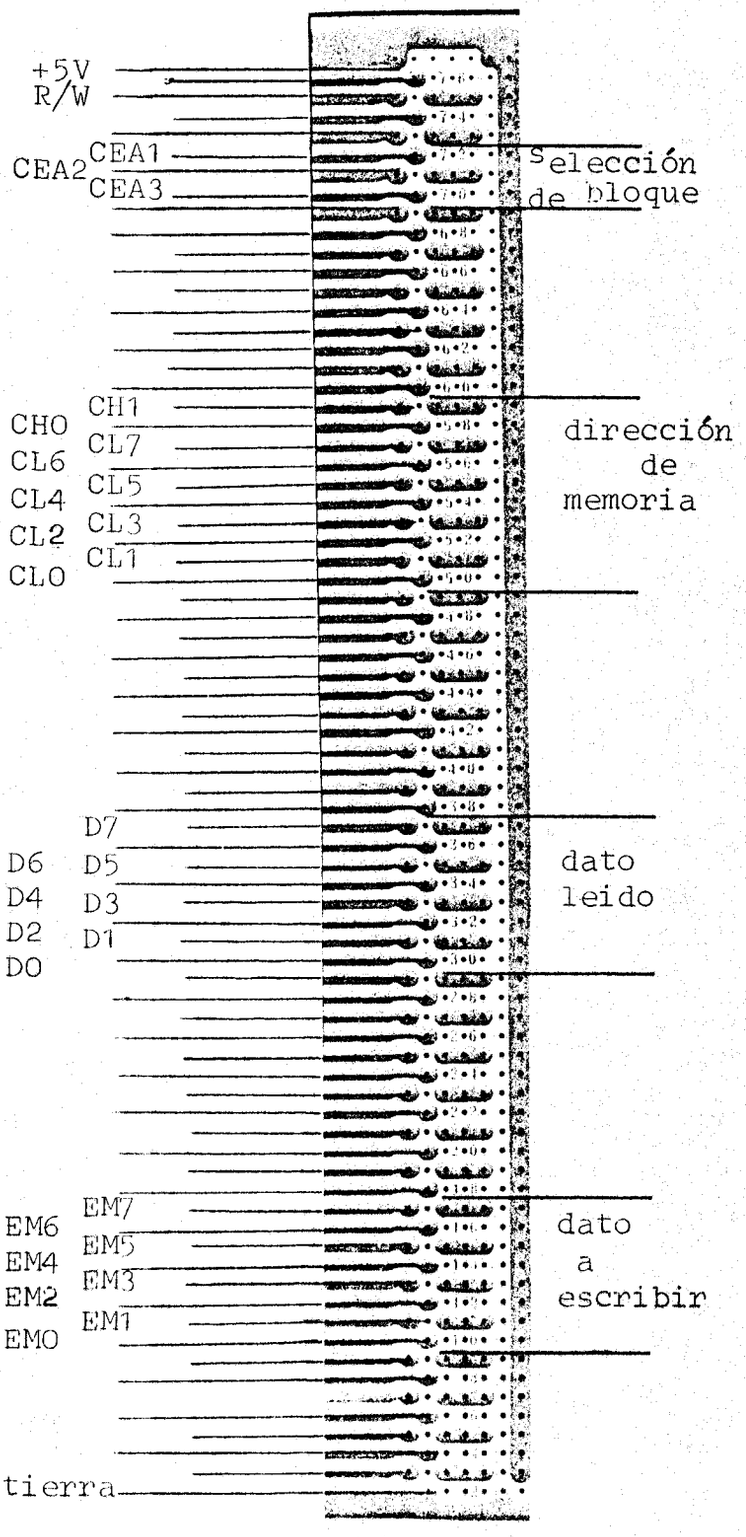
Nº	TIPO	FUNCION
1-24	C2102A	1024 bits memoria RAM
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		

OBSERVACIONES: La tarjeta contiene 24 pastillas de memoria del tipo mencionado.

ENTRADAS Y SALIDAS

TARJETA A

CONTENIDO: memoria RAM



OBSERVACIONES:

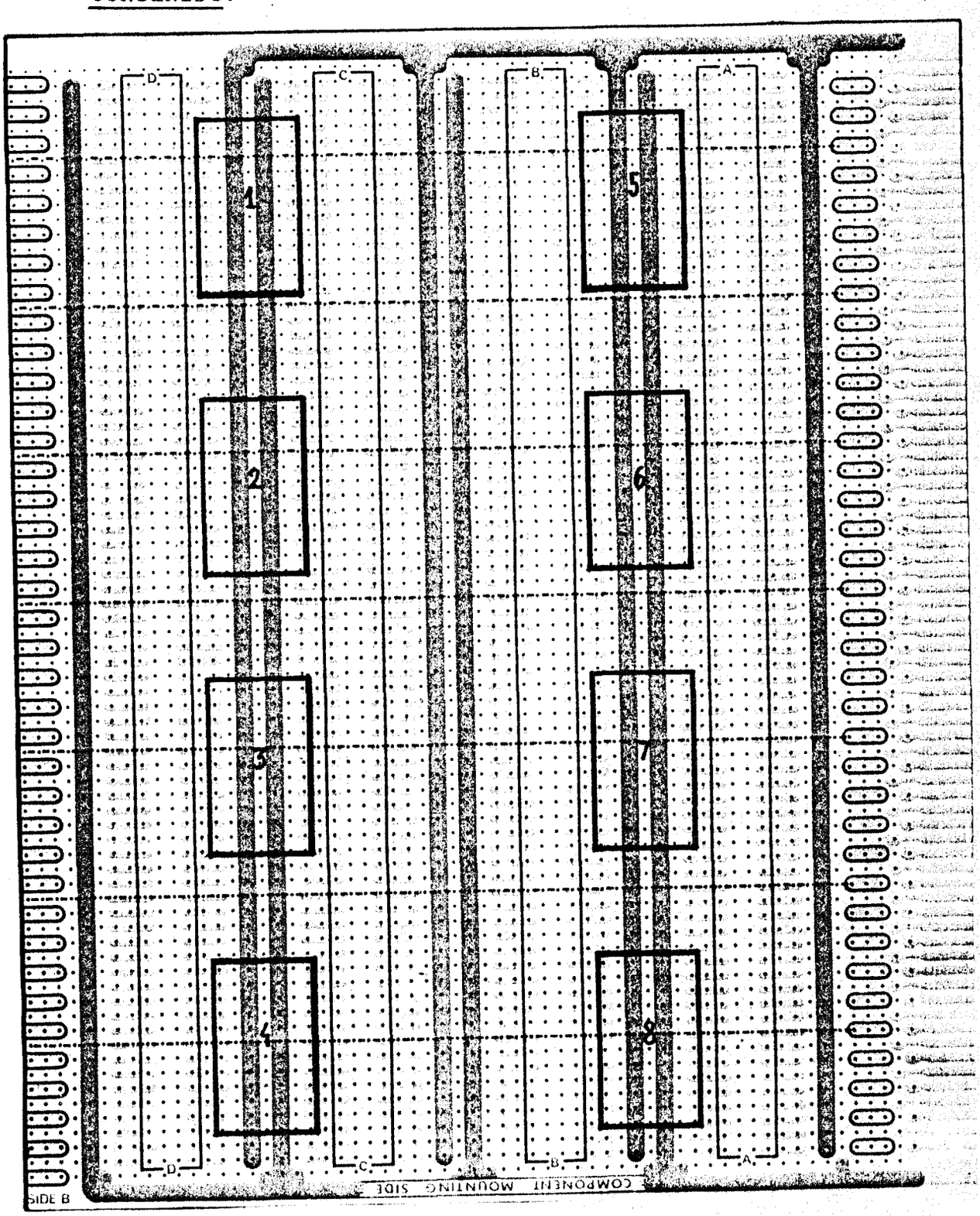
2ºK 3ºK 4ºK

8
7
...
1
0

P O S I C I O N D E C O M P O N E N T E S

TARJETA Nº O

CONTENIDO: memoria REPROM



OBSERVACIONES:

COMPONENETES UTILIZADOS

TARJETA Nº 0

CONTENIDO: Memoria REPROM

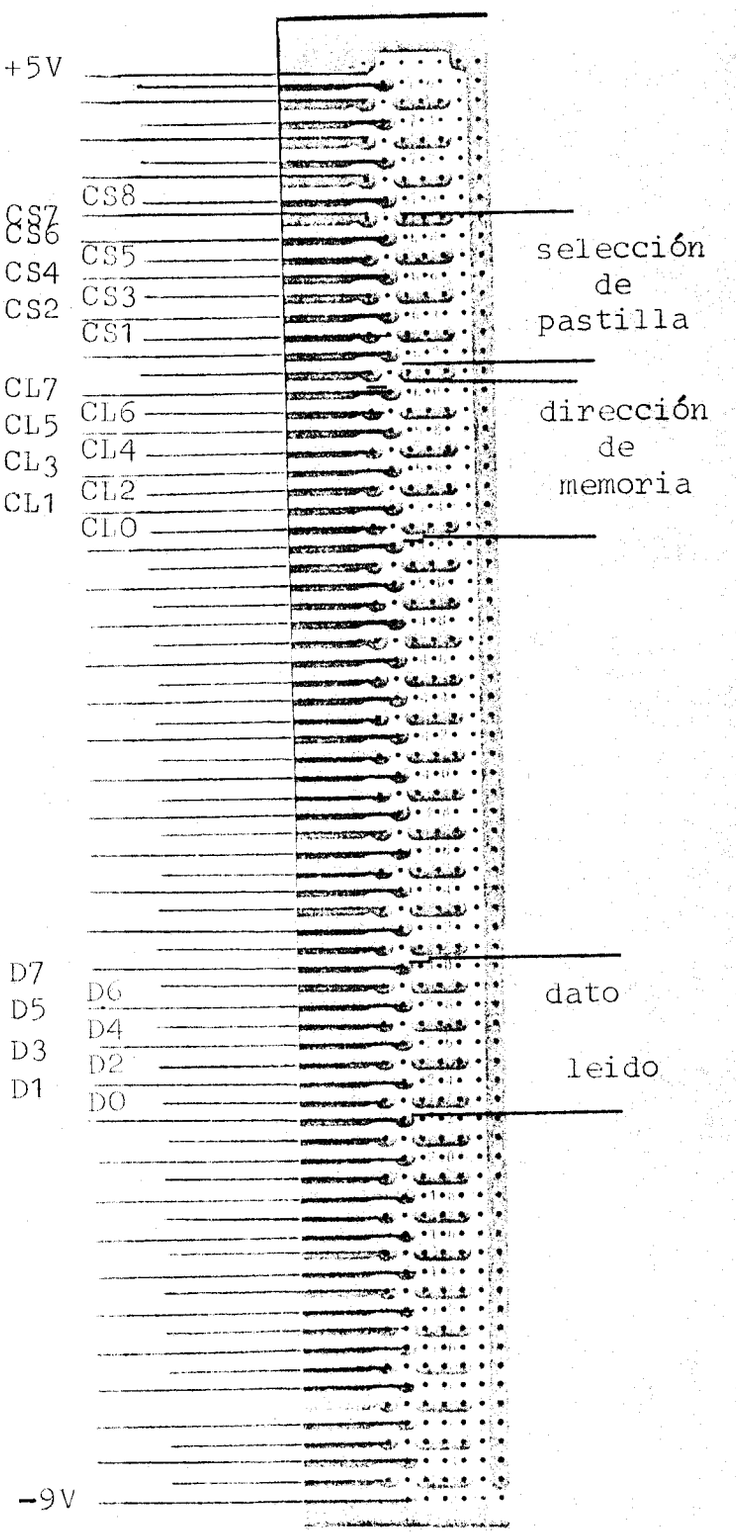
Nº	TIPO	FUNCION
1-8	C1702A	256x8 memoria REPROM
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		

OBSERVACIONES: La tarjeta contiene 8 pastillas del tipo mencionado.

ENTRADAS Y SALIDAS

TARJETA 0

CONTENIDO: Memoria REPROM



OBSERVACIONES:

8	4
7	3
6	2
5	1

APENDICE VI

RESUMENES SOBRE MENSAJES Y LENGUAJES.

CODIGO	ESTADO	MENSAJE	TIPO
1111 1111	STOP	demanda de una instrucción	DEMANDA
0000 0001	ESPERA	demanda de dirección baja de m.	DEMANDA
1000 0000	ESPERA	demanda de dirección alta de m.	DEMANDA
0000 1000		inicio de carga de un programa máquina.	DEMANDA
0000 1111	STOP	se intenta leer o escribir más allá de la capacidad de memoria	ERROR
0001 1111	STOP	se intenta escribir en m ROM	ERROR
1111 0000	STOP	el dividendo es mayor en valor absoluto que el divisor.	DIAGNOSTICO
1111 0001	STOP	desbordamiento de la mantisa	DIAGNOSTICO
1111 0010	STOP	desbordamiento del exponente.	DIAGNOSTICO
1111 0100	ESPERA	comienzo de lectura de función.	DEMANDA
1111 0101	ESPERA	final de lectura de una función.	DEMANDA
1111 0110	ESPERA	comienzo de escritura de una función.	DEMANDA
1111 0111	ESPERA	final de escritura de una función.	DEMANDA
0010 1111	STOP	leer o escribir PF más hallá de la pag.	ERROR

Relación de mensajes que puede dar el sistema operativo.

NEMOTECNICO	BINARIO	OPERACION O TAREA A REALIZAR
.LEER	0000 0101	leer de memoria
.ESCRIBIR	0000 1101	escribir en memoria RAM
.LEER PF	0001 0101	leer página programa fuente
.ESCRIBIR PF	0001 1101	escribir o cargar programa F.
.EJC PM	0010 0101	ejecutar un progr. en leng.básico
.EJC PF	0011 1101	ejecutar un programa fuente
.LEER RTROS	0010 1101	lectura del contenido de los registros

instrucciones del lenguaje de
Control.

INSTRUCCIONES DE LOS LENGUAJES ENSAMBLADOR Y MAQUINA

I. TRANSFERENCIA DE DATOS:

ENSAMBLADOR	BINARIO	Fto.	OPERACION	N. IVERSON
LOAD d,s	11DD DSSS	1	transfs. registros	d <—— s
LOAD d,(HL)	11DD D111	1	lectura memoria	d <—— (H,L)
LOAD (HL),s	1111 1SSS	1	escritura en memoria	(H,L) <—— s
LOAD d,*n	OODD D110	2	escritura inmet.rtr.	d <—— n
LOAD (HL),*n	0011 1110	2	escritura inmet.mri.	(H;L) <—— n
INR d	OODD DOOO	1	incremento de rtro.	d <—— d+1(d≠A)
DCR d	OODD DOO1	1	decremento de rtro.	d <—— d-1(d≠A)

III. SALTOS:

JUMP m	01-- -100	3	salto incondicional	PC <—— m
JUMP FC,m	010P P000	3	salto condicional	F:0;(=)->(PC<-m)
JUMP FS,m	011P P000	3	salto condicional	F:1;(=)->(PC<-m)
CALL m	01-- -110	3	llamada rtna.inconn.	ST<—PC;PC<—m
CALL FC,m	010P P010	3	llamada rtna.condic.	F:0;(=)—>(PC<-m)
CALL FS,m	011P P010	3	llamada rtna.condic.	F:1;(=) —>(PC<-m)
RET	0C-- -111	1	retorno a programa	PC <—— ST
RET FC	000P P011	1	retorno condicional	F:0;(=)—>(PC<-ST)
RET FS	001P P011	1	retorno condicional	F:1;(=)->(PC<-ST)
RST L	00AA A101	1	llamada RESTART	ST<-PC;PC<-000AAA 00

IV. OTRAS INSTRUCCIONES:

INP e	0100 MMM1	1	entrada disposit. e	A <—— e
OUT s	01MM M1M1	1	salida disposti. s	s <—— A
HLT	0000 000.	1	entrada en STOP	
HLT	1111 1111	1	entrada en STOP	
NOP	1100 0000	1	no operar	

V. PSEUDOFUNCIONES:

DIR			puntos de direccio-namiento simbolico.	
EQU			puntos de direccio-namiento simbolico.	

Formato lenguaje Ensamblador:

1 5	6 10	11 20
etiqueta	operación	operandos

II. ARITMETICO-LOGICAS:

ADD A,s	1000 0SSS	1	adicción	A <—— A +s
ADD A,(HL)	1000 0111	1		A<—— A + (H,L)
ADD A,*n	0000 0100	2		A<—— A + n
ADDC A,s	1000 1SSS	1	adicción con "carry"	A<—— A + s +c
ADDC A,(HL)	1000 1111	1		A<—— A +(H,L)+c
ADDC A,*n	0000 1100	2		A<—— A + n + c
SUB A,s	1001 0SSS	1	resta	A<—— A - s
SUB A,(HL)	1001 0111	1		A<—— A - (H,L)
SUB A,*n	0001 0100	2		A<—— A - n
SUB A,s	1001 1SSS	1		A<—— A - s -c
SUBC A,(HL)	1001 1111	1	resta con "borrow"	A<—— A- (HL)-c
SUBC A,*n	0001 1100	2		A<—— A - n -c
AND A,s	1010 0SSS	1	intersección	A<—— A s
AND A,(HL)	1010 0111	1		A<—— A (HL)
AND A,*n	0010 0100	2		A<—— A n
XOR A,s	1010 1SSS	1	exclusive OR	A<—— A ⊕ s
XOR A,(HL)	1010 1111	1		A<—— A ⊕ (H,L)
XOR A,*n	0010 1100	2		A<—— A ⊕ n
OR A,s	1011 0SSS	1	unión	A<—— A Vs
OR A,(HL)	1011 0111	1		A<—— A V (H,L)
OR A,*n	0011 0100	2		A<—— A V n
COMP A,s	1011 1SSS	1	comparación	A : s
COMP A,(HL)	1011 1111	1		A : (H,L)
COMP A,*n	0011 1100	2		A : n
RL A	0000 0010	1	rotación a izquier.	A<—— A
RR A	0000 1010	1	rotación a derecha	A<—— A
RLC A	0001 0010	1	rotación con carry iz	(c,A)<—— (c,A)
RRC A	0001 1010	1	rotación con carry d.	(c,A)<—— (c,A)

d ó s: A 000, B 001, C 010, D 011, E 100, H 101, L 110 → DDD ó SSS

F: C 00, Z 01, S 10, P 11 → PP

e: E0 000, E1 001 → MMM

s: S0 11 000, S1 11 001, S2 11 010, S3 11 011 → MM MMM

—: indiferencias.
ST<—— PC apilar direcciones
PC: contador de progrmas.
ST: memoria pila.

I. TRANSFERENCIA DE DATOS:

NEMOTECNICO	BINARIO	FORMATO	Operación	notación Iverson
LOADF FO,Fn	0101 1000	2	transf. de fcion.	FO<— Fn
LOADF Fn,FO	0101 1010	2	transf. de fcion.	Fn<— FO
LOADF Fn,A	0110 0010	2	definic. fción. cte.	Fn<—(A,.....,A)
LOADF A,Fn(C)	0110 0101	2	obtenc. punto de fcion.	A<— Fn(C)
LOADF Fn(C),A	0110 0110	2	definic. punto fcion.	Fn(C)<— A
LOADF PNP,*n	0110 0111	4	definic. Nº pág./fcion.	PNP<— n
LOADF PNP,C	0110 1010	1	"	PNP<— C
LOADF PR,*n	0110 1001	4	definic. ralentización	PR<— n
LOADF PR,C	0110 1011	1	"	PR<— C
LOADF A,PNP	0110 1100	1	obtenc. nº pág./fcion.	A<— PNP
LOADF A,PR	0110 1101	1	obtenc. ralentización	A<— PR
LOADF XFn,C	0110 1110	2	definic. de exponente	XFn<— C
LOADF A,XFn	0110 1111	1	obtenc. de expoenente.	A<— XFn
LOAD d,s	1001 DDSS	1	transf. rtros. y m.	d<— s
LOAD d,*n	1000 DD00	4	definic. de variables.	d<— n

III. SALTOS:

JUMP AN,C	1000 0001	1		A:0;(<)-CP - C
JUMP AN,*n	1000 0101	4		A:0;(<)-CP - n
JUMP AP,C	1000 0010	1		A:0;(>)—>CP<—C
JUMP AP,*n	1000 0110	4	saltos condicionales	A:0;(>)—>CP<—n
JUMP AZ,C	1000 0011	1		A:0;(=)—>CP<—C
JUMP AZ,*n	1000 0111	4		A:0;(=)—>CP<—n
CALL C	1000 1001	1		(ST)<—PC, PC-C
CALL *n	1000 1101	4	rutinas	(ST)<—PC, PC-n
RET	1000 1010	1		CP<— (SP)
JUMP C	1000 1011	1	saltos incodicionales	CP<— C
JUMP *n	1000 1111	4		CP<— n

IV. OTRAS INSTRUCCIONES:

INPF Fn	0111 0000	2	entrada de función	Fn<— Eo
INPF FC	0111 0011	1	"	FC<— Eo
OUTF Fn	0111 0100	2	salida de función	SO<— Fn
OUTF FC	0111 0111	1	"	SO<— FC
INP e	0001 110E	1	entrada de disp.E0 y E1	A<— Ee
OUT S	0001 111s	1	salida en disp. S0 yS1	Ss<— A
STOP	1111 1111	1	fin de programa	STOP
ER	1111 1011	3	Intercalar rut. L.M.	

d o s	DD o SS
A	00
B	01
C	10
(C)	11

d≠s

AN: acumulador negativo
AP: acumulador positivo
AZ: acumulador cero.

e { 0—conversor A/D ; 1—consola/teletipo }

s { 0—conversor D/A ; 1—consola }

II. ARITMETICO-LOGICAS:

ADDF FO,Fn	0010 0000	2	suma fción.	FO<— FO + Fn
SUBF FO,Fn	0010 1000	2	resta fción.	FO<— FO - Fn
MULF FO,Fn	0010 1111	2	multiplicación	FO<— FO * Fn
DIVF FO,Fn	0011 0011	2	divison.fcnes.	FO<— FO / Fn
INTF FO,Fn	0011 0111	2	integral indef.	FO<— $\int$Fn dt
DRVF FO,Fn	0011 1100	2	derivada fción.	FO<— dFn/dt
RTPF Fn,Fn	0100 0110	2	rotación+en tip	(Fn(i),Fn(o),...,Fn(i))
RTNP Fn,Fn	0100 1001	2	rotación-en t.	(Fn(1),Fn(2),...,Fn(o))
TCPF Fn,Fn	0100 1100	2	traslac. + en t	(0,Fn(o),...,Fn(i))
TCNF Fn,Fn	0100 1110	2	traslac. - en t	(Fn(1),...,Fn(i),o)
MINF A,Fn	0101 0000	2	mímimo fción.	A<—MIN(Fn(o),...,Fn(i))
MAXF A,Fn	0101 0100	2	máximo fción.	A<—MAX(Fn(o),...,Fn(i))
DRCF Fn,Fn	0101 1110	2	desplaz. d.	XFn<—XFn+C Fn—Fn/2C
DLCF Fn,Fn	0110 0000	2	despito. izda.	XFn<—XFn-C Fn—2CFn
ADD A,B		1		A<— A+B
ADD A,(n)	1010 00xx	3	Suma	A<— A+(n)
ADD A,*n		4		A<— A+n
SUB A,B		1		A<— A-B
SUB A,(n)	1010 01xx	3	Resta	A<— A-(n)
SUB A,*n		4		A<— A-n
MUL A,B		1		A<— A*B
MUL A,(n)	1010 10xx	3	Multipl.	A<— A*(n)
NUL A,*n		4		A<— A*n
DIV A,B		1		A<— A/B
DIV A,(n)	1010 11xx	3	división	A<— A/(n)
DIV A,*n		4		A<— A/n
AND A,B		1		A<— A B
AND A,(n)	1011 00xx	3	intersección	A<— A (n)
AND A,*n		4		A<— A n
XOR A,B		1		A<— A $\oplus$ B
XOR A,(n)	1011 01xx	3	unión exclusiva	A<— A $\oplus$ (n)
XOR A,*n		4		A<— A $\oplus$ n
OR A,B		1		A<— AVB
OR A,(n)	1011 10xx	3	unión	A<— AV(n)
OR A,*n		4		A<— AVn
DL A	1010 0011	1	desplazamiento	A<— A
DR A	1010 0111	1	despito. dcha.	A<— A
RL A	1010 1011	1	rotación izda.	A<— A
RR A	1010 1111	1	rotción. dcha.	A<— A

xx { A,B 00 ; A,(n) 01 ; A,*n 10 }

APENDICE VII

ALGUNAS FOTOGRAFIAS SOBRE EL DISPOSITIVO

ALGUNAS FOTOGRAFIAS DEL DISPOSITIVO:

A7-1 Vista general del dispositivo.

A7-2 Vista frontal: panel de control.

A7-3 Vista posterior: conectores y sus conexiones.

A7-4 Disposición de tarjetas en el interior del dispositivo.

A7-5 Tarjeta de control de panel.

A7-6 Tarjeta de los conversores A/D y D/A.

A7-7 Tarjeta de entradas.

A7-8 Tarjeta de salidas.

A7-9 Tarjeta de unidades aritmético-lógica y de control (CPU)

A7-10 Tarjeta de memoria RAM (3k x 8 bits)

A7-11 Tarjeta de memoria REPROM (2k x 8 bits)

A7-12 Fuentes de alimentación de +15, -15 y -8 voltios.

A7-13 Señal de sincronismo dada por el microprocesador y señal Ø1 de reloj. Base de tiempos: 1 microsegundo/centimetro.

A7-14 Programa de prueba de interfase de conversores:

```
a   INP  EO
    OUT  EO
    JUMP a
```

Parte superior del oscilograma: función de entrada.
parte inferior del oscilograma: función de salida.
Frecuencia: 400 ciclos/segundo.

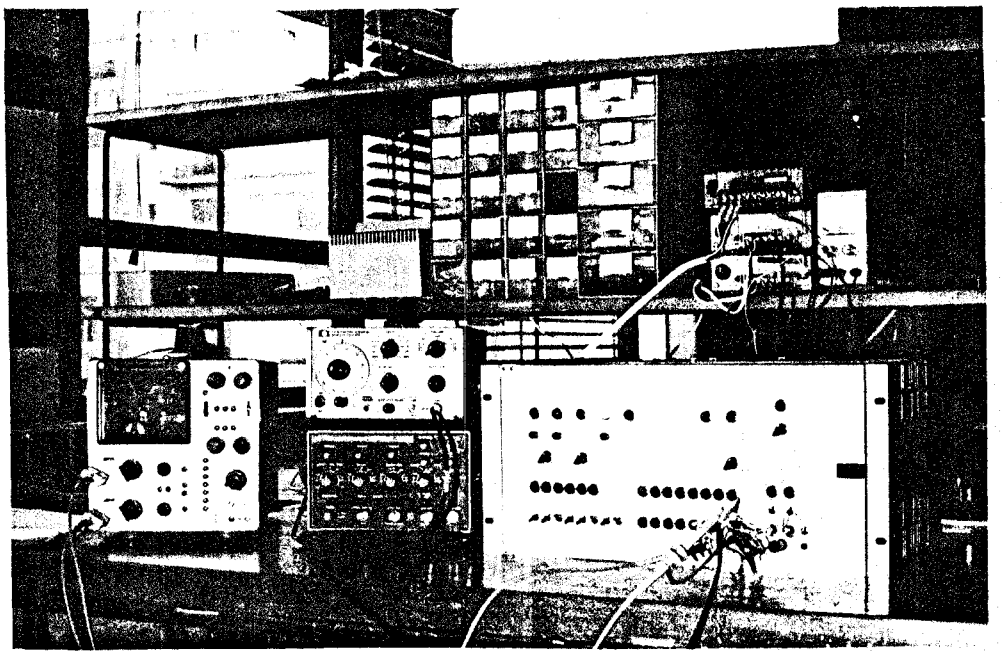


figura A7-1

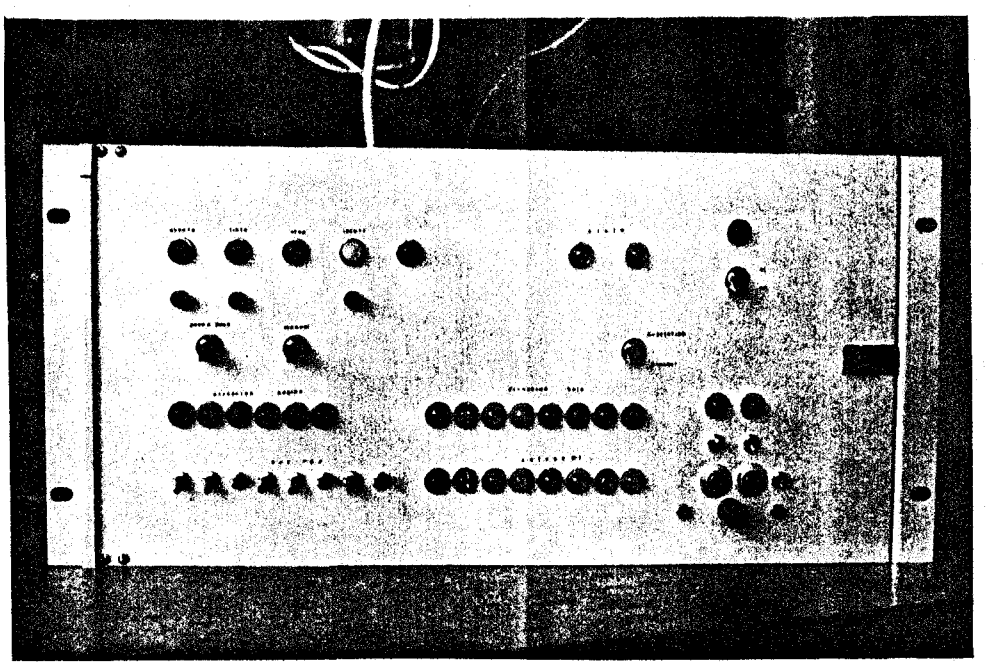


figura A7-2

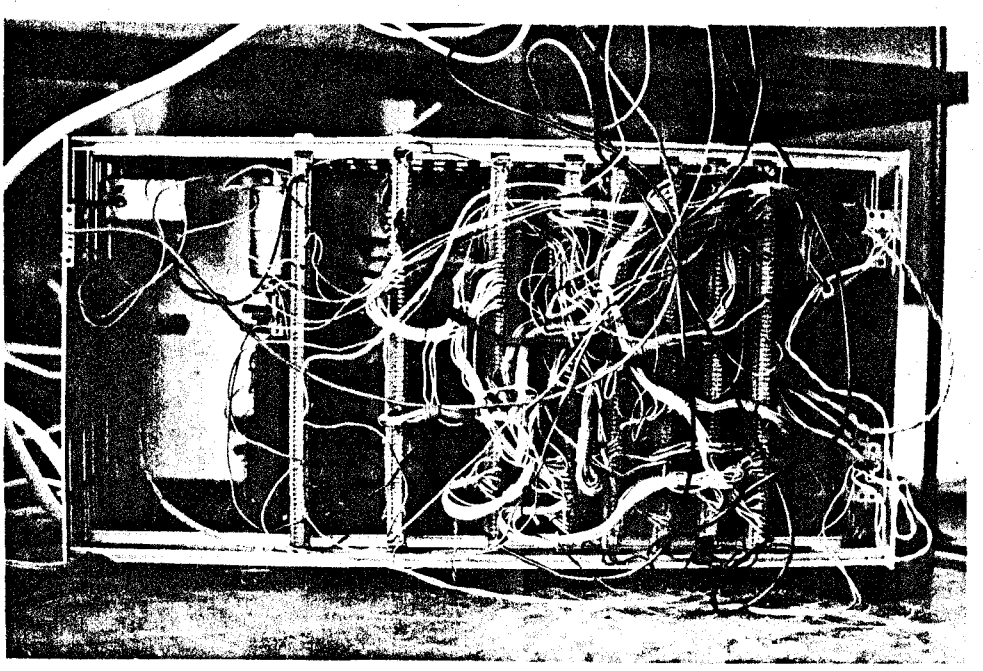


figura A7-3

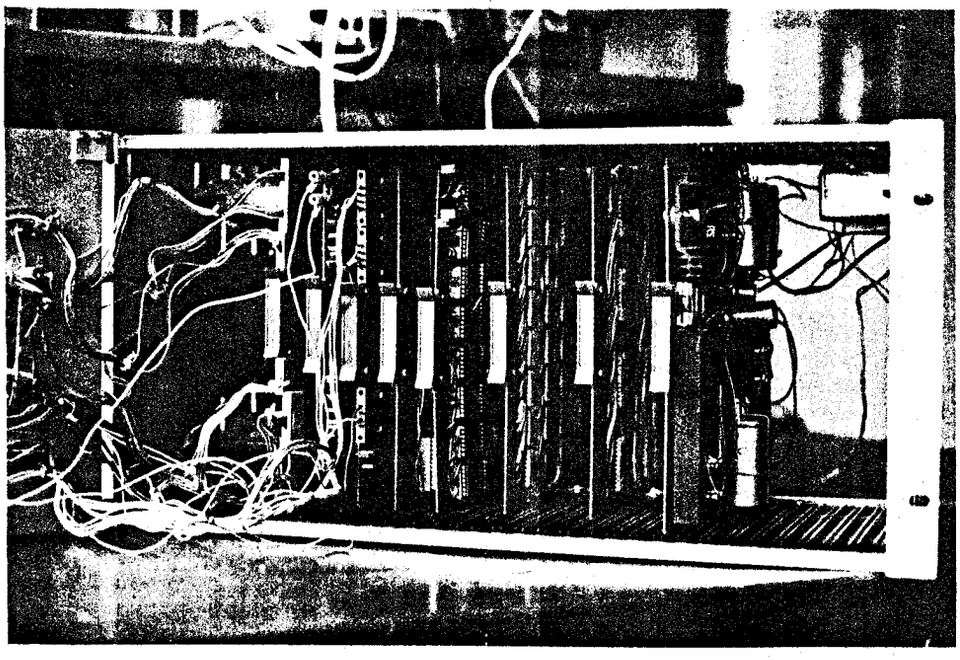


figura A7-4

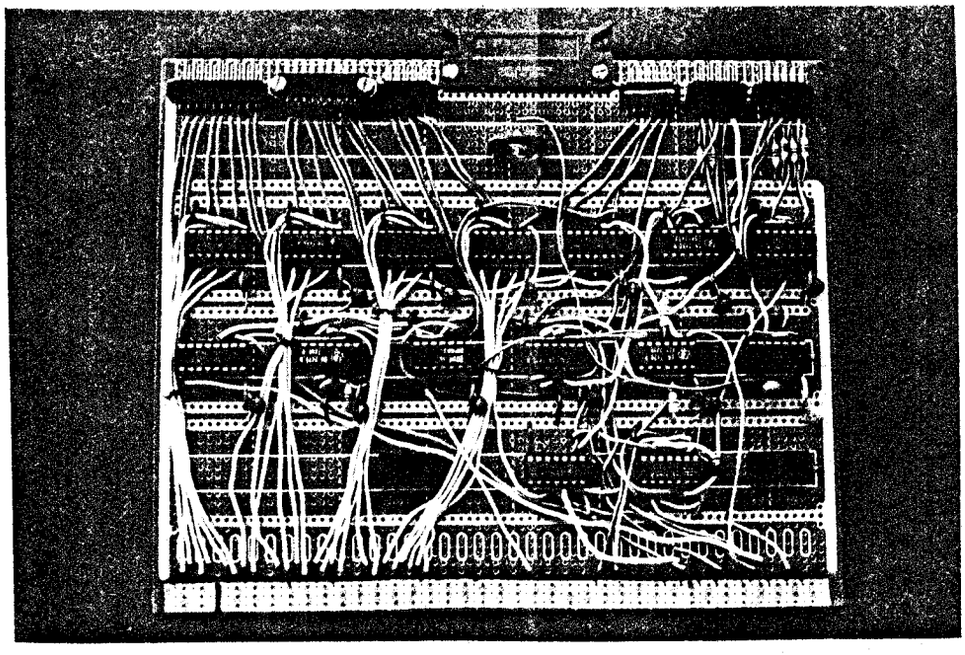


figura A7-5

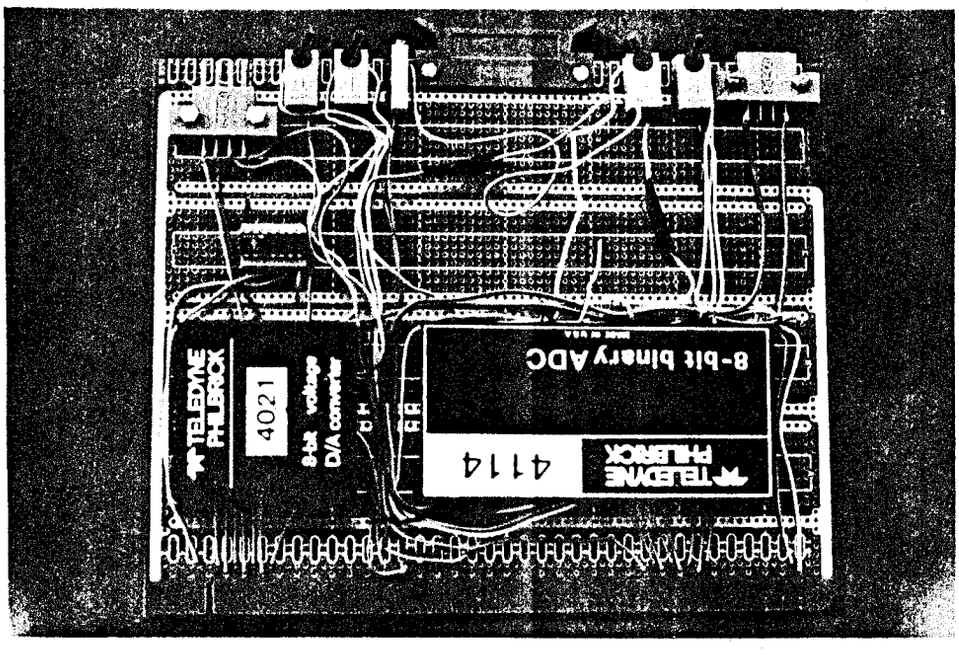


figura A7-6

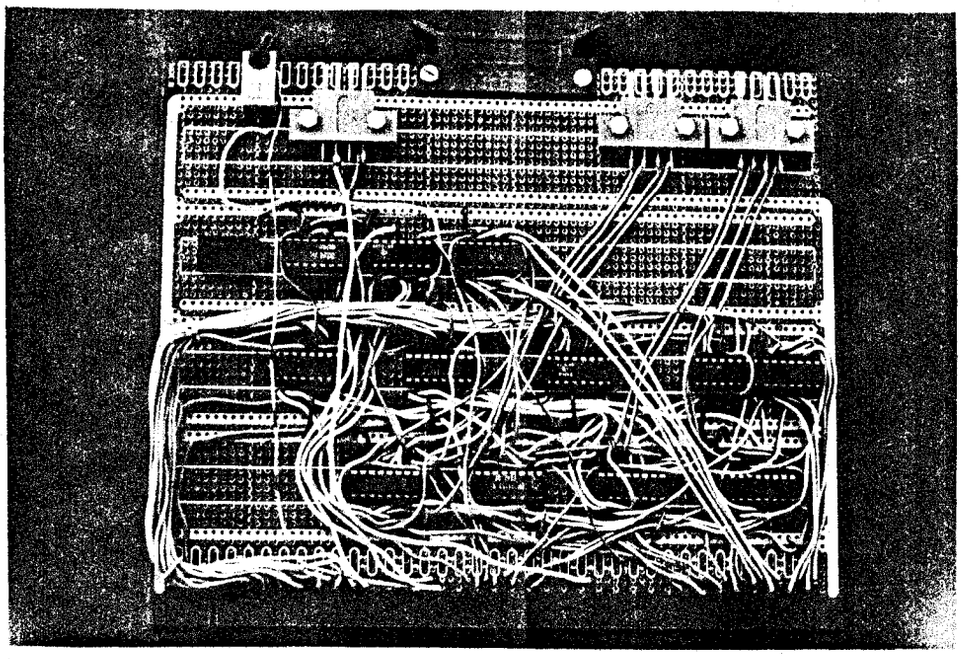


figura A7-7

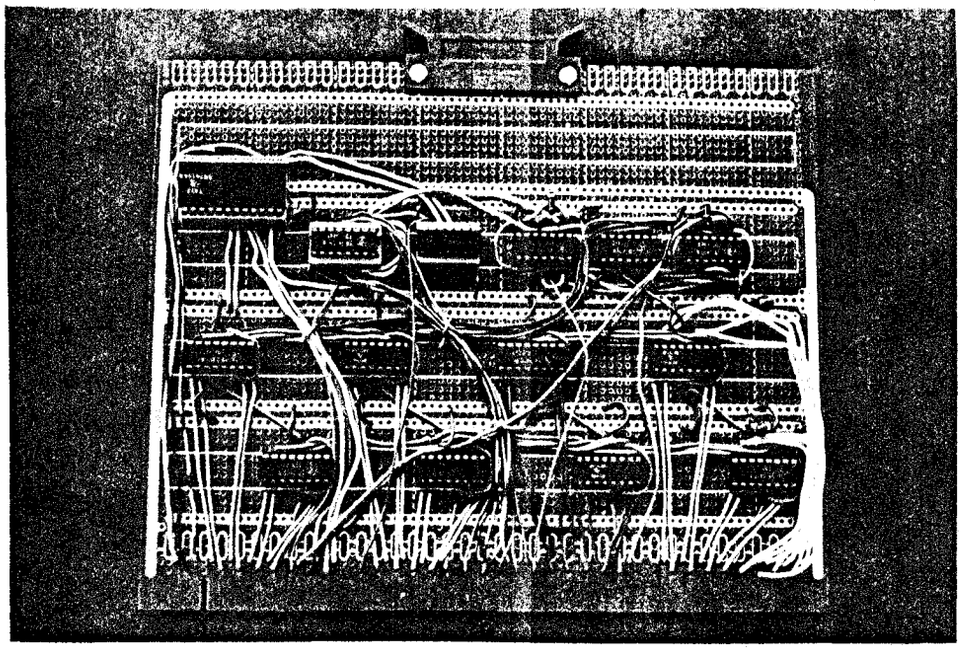


figura A7-8

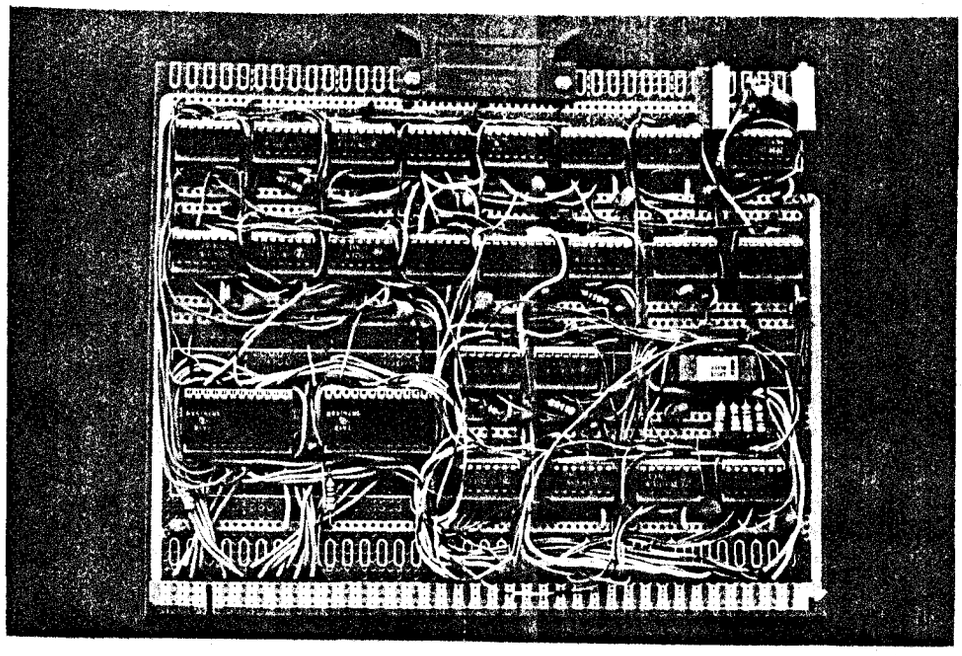


figura A7-9

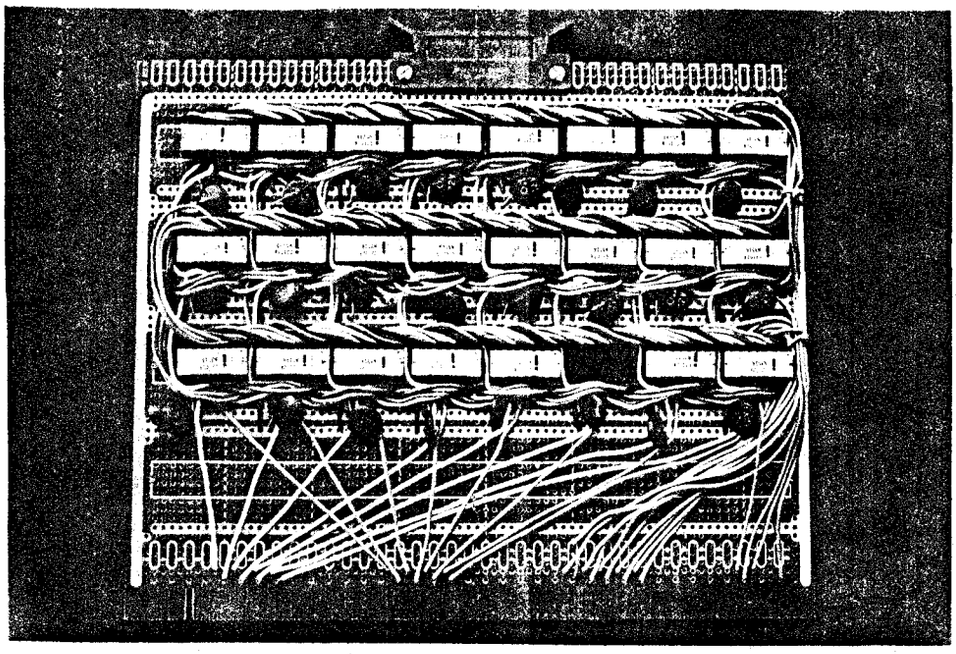


figura A7-10

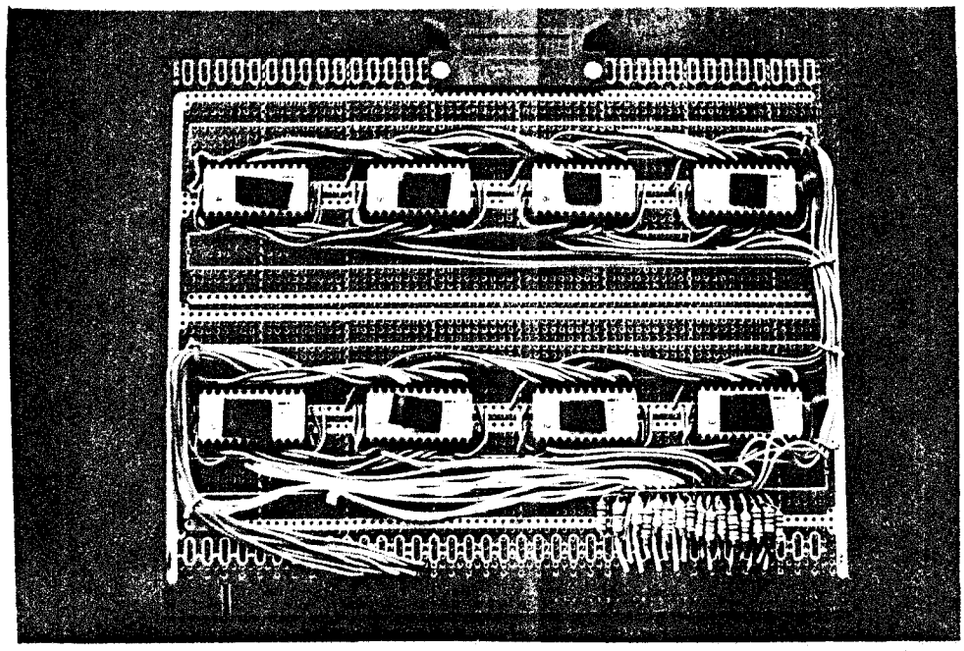


figura A7-11

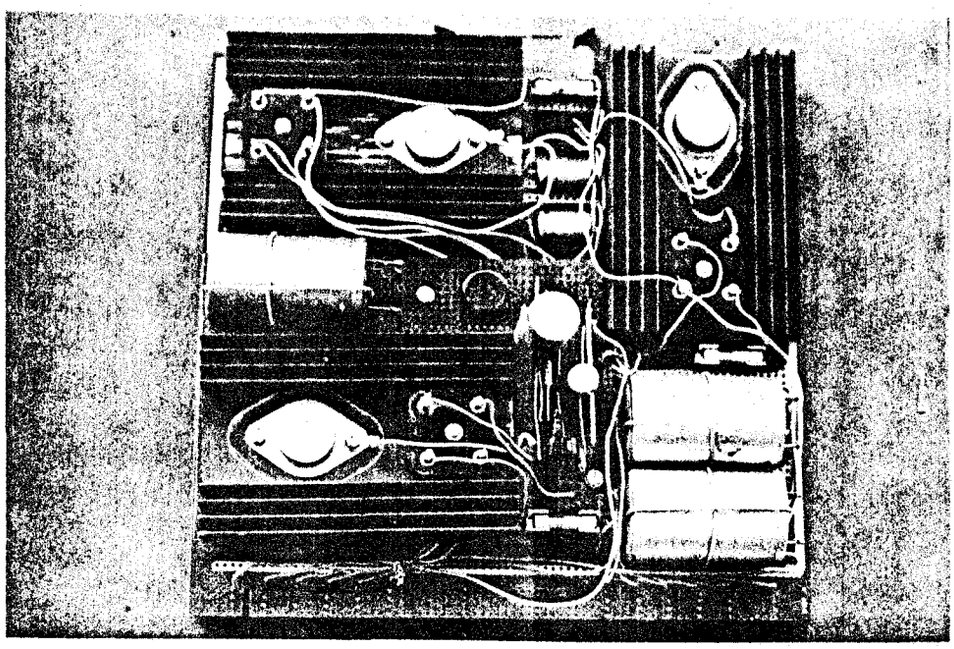


figura A7-12

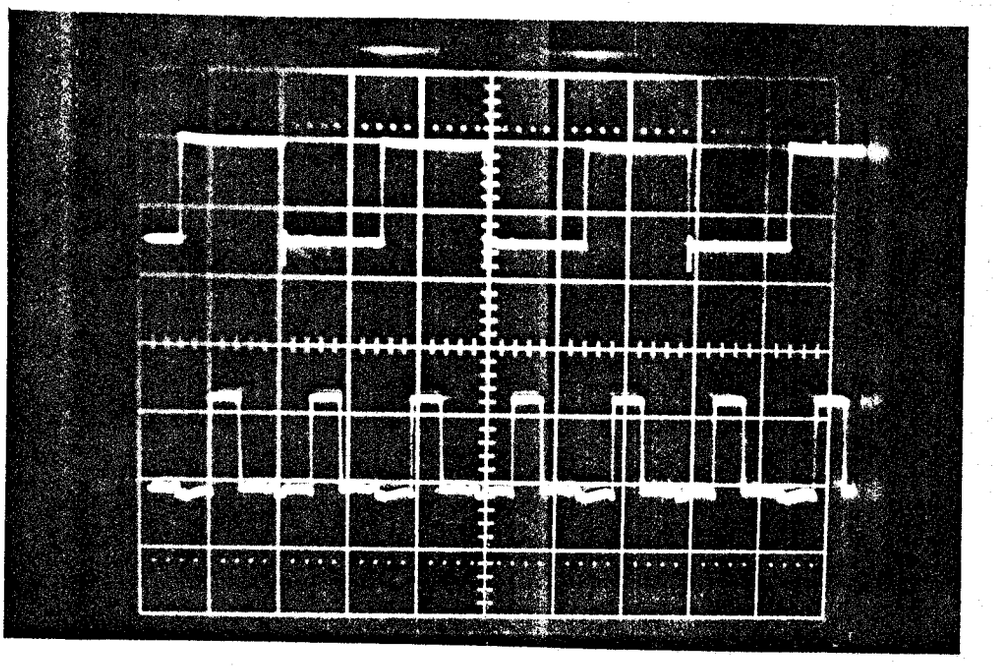


figura A7-13

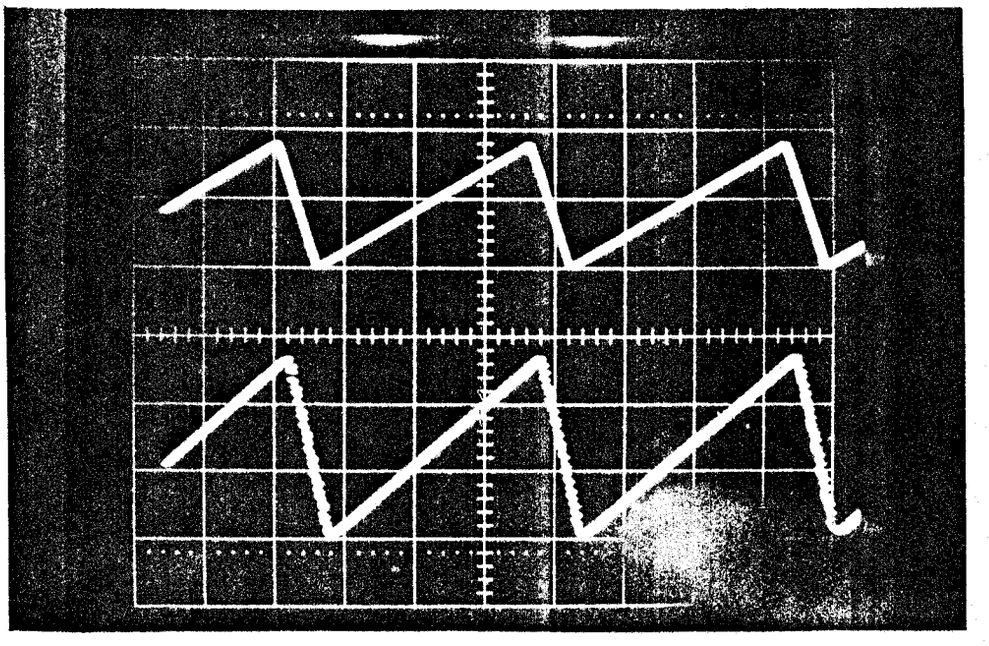


figura A7-14

BIBLIOGRAFIA

(1) ATKINS, D.E.; GARNER,H.L. "Computer Arithmetic: An Introduction and Overview".
IEEE Transation on Computers Vol. C-22, Nº 6, págs . 549-551, junio 1973.

(2) AVIZIENIS, A. "Arithmetic Algorithms for Error-Coded Operands".
IEEE Transation on Computers. Vol. C-22, nº 6, págs. 567-572, junio 1973.

(3) BARNA, A.; PORAT, D.I. "Integrated Circuits in digital Electronics".
John Wiley New York. 1973.

(4) BARRON, D.W. "Computer Operatings Systems"
Chapman and Hall LTD. Londres, 1971.

(5) BARS, JACOB, MONCHAUD, ROUSSEL "Introduction a l'étude des microprocesseurs logiques"
Institut National des Sciences Appliquees de Rennes, 1975.

(6) BARTEE T.C.; LEBOW I.L.; REED,I.S. "Theory and Design of Digital Machines"
MacGraw Hill, 1962.

(7) BAZERQUE, G.; TRULLEN C. "Informatique générale"
Dunod - Paris. 1971.

(8) BEAUCHAMP K.G.; KELLEY C.J. "Development of a Hybrid Systems for University Research"
IFIP Congress 71-Ljubljana, TA-4, págs. 97-102, Agosto 1971.

(9) BEKEY G.A.; KARPLUS W.J. "Hybrid Computation"
John Wiley & Sons, New York, 1968.

(10) BELL&HOWELL. "PM16 series-Polynomialised microcomputer"
Boletin /PM16/, Basigngstoke, abril 1975.

(11) BOOTH, A.D. "A signed Binary multiplication Technique"
Q.J. Mech. Appl. Math. Vol. 4 nº 2, págs. 236-240, 1951.

(12) BOULAYE, G.ED. "Arquitecture and Design of Digital Computers"
Dunod, 1971.

(13) BOULAYE, G. "Logique et Organes des Calculatrices Numériques".
Dunod, 1970.

(14) BOULAYE, G. "Logique et Organes des Calculatrices Numériques".
Departemente de Mathematiques et Informatique, Université de Rennes, 1974.

(15) BOULAYE, G.; MERMET J. EDS. "Microprogramming"
Hermann, 1972.

(16) BOULAYE,G. "Microprogramming"
Mac Millan-London, 1974.

(17) BOULAYE,G. "La Microprogramation"
Dunod, 1971.

(18) BOULAYE,G. "Microprogramming and structured design"
Conferences on Computer Systens and Technology, Págs. 73-84, IEE , 1974.

(19) BRENT, R.P. "On the Precision attainable with various Floating-Point Number Systems"
IEEE Transactions on Computers, Vol. C-22, nº 6, págs. 601 a 607, junio 1973.

(20) BRENTR, R.P.; KUCK D.; MARUYAMA K, "The Parallel Evaluation of Arithmetic Expressions without Division"
IEEE Transactions on Computers, págs. 532 a 534, Mayo 1973.

(21) CAPPA,M.; HAMACHER V.C. "Anda Augmented Iterative Array For High-Speed Binary Division".
IEEE Transactions on Computers, Vol. C-22, Nº 2, págs. 172 a 175, febrero 1973.

(22) CHU "Digital Computer Design Fundamentals"
McGraw Hill - New York, 1962.

(23) CHU "Computer Organization and Microprogramming"
Prentice Hall, 1972.

(24) DAGLESS E.L. "Micro-Computer Systen Structures for Instrumentation"
1974 Conference on Computer Systens and Technology, IEE London, págs. 64 a 72, octubre 1974.

(25) DAS S.R.; BANERJI D.K. CHATTOPADHYAY "On Control Memory Minimization in Microprogrammed Digital Computers"
IEEE Transactions on Computers, Vol. C-22 nº 9, págs. 845 a 848, septiembre 1973.

(26) ALLEN DURLING "Computational Techniques Analog, Digital and Hybrid Systems"
Intertex Educational, 1974.

(27) DURLING A.E.; BULLOCK T.E. "A unified method for the reconstruction of sampled data"
IEEE Trans. Computers., Vol. C-22, Nº 4, págs. 388 a 396 abril 1973.

(28) ECKHAUSE R.H. "A high-level microprogramming Language (MPL)"
Proc. SJCC, págs. 169 a 177, 1971.

(29) ERCEGOVAC M.D. "Radix- 16 Evaluation of Certain Elementary Functions"
IEEE Transactions on Computers, Vol. C-22, Nº 6, págs. 561 a 566, junio 1973.

(30) "Graphics Symbols for Electronics Diagrams"
ELECTRONICS, Vol. 48, Nº 7, 3 abril 1975.

(31) FAIRCHILD "The TTL Applications Handboock"
Fairchied Semiconductor, California, Agosto 1973.

(32) FAIRCHILD "TTL Data Book"
Fairchild Semiconductor-California, junio 1972.

(33) FALK H. "A Checkup: Minicomputer Software"
IEEE Spectrum, págs. 52 a 56, febrero 1974.

(34) FALK H. "Liking Microprocessors to the Real World"
IEEE Spectrum, págs. 59 a 67. septiembre 1973.

(35) FALK H. "Microcomputer Software Makes its Debut"
IEEE Spectrum, págs. 78 a 84 ,octubre 1974.

(36) JAURE J.C. "Emploi des ordinaterus. Introduction au Software"
Dunod, 1971.

(37) FITZGERALD J.W.; CLAPPER R.J.; HARRISON D.C. "Small Computer Processing of Ambulatory Electrocardiogramas"
Computer, Vol. 8, Nº 7, págs. 48 a 54, julio 1975.

(38) FLORES I. "Computer Organization"
Prentice-Hall, 1969.

(39) FLORES I. "El software en los ordenadores"
Ediciones Deusto-Bilbao, 1973.

(40) FLORES I. "The Logic of Computer Arithmetic"
Prentice-Hall, INC. London, 1963.

(41) FLYNN M.J.; ROSIN R.F. "Microprogramming an introduction and viewpoint"
IEEE Trnas. on Computers, Vol. C-22, Nº julio, págs. 727 a 731, 1971.

(42) FOSTER "A view of Computer Architecture"
Communication of the ACM, Nº 07, 1972.

(43) GARCIA MERAYO F. "Glosario de Informática"
Edic. Urmo. 1971.

(44) GARCIA OLMEDO B. "Un nuevo proceso en el cálculo analógico de la integral f(t) dg(t) entre 0 y TO mediante una unidad analógica con conmutación digital"
Publicaciones de la Univ. de Sevilla. 1969.

(45) FOSTER C. "Computer Architectura"
Van Nostrand- New York, 1970.

(46) GAREN E.R. "Applying microprocessors and microcomputers"
Modern Data, págs. 54 a 57, febrero 1975.

(47) GERACE G.B. "Micro-program control for computing Systems"
IEEE Transactions on Electronic Computers, Vol. EC-12,
Nº 5, págs. 733 a 747, diciembre 1963.

(48) GILOI W.; BECKERT D.; LIEBIG H,C. "A flexible standard programming system for hybrid computation"
AFIRPS Conference Proceedings, AFIPS Press New Jersey,
Vol. 34, 1969.

(49) GOMEZ R. "Dispositivo para almacenamiento de funciones en memoria y generación de función de función"
Tesis Doctoral. Dto. de Electricidad y Electrónica,
Universidad de Granada, 1973.

(50) GREBENE A.B. "Analog Integrated circuit design"
Van Nostrand Reinhold Co,. págs. 401 pp., 1972.

(51) GRIES, G. "Compiler Construction for Digital Computers"
John Wiley - New York, 1971.

(52) HAUSNER A. "Analog and Analog Hybrid Computer Programming"
Prentice-Hall N.J., 1971.

(53) HILL F.J. "Introducing AHPL"
Computer, Vol. 7, Nº 12, págs. 28 a 30, diciembre 1974.

(54) HILL F.; PETERSON G. "Digital Systems: Hardware Organization and Design"
Wiley, 1973.

(55) HILL F.J.; PETERSON G.R. "Introduction to Switching Theory and Logical Design"
Wiley-New York, 1974.

(56) HOESCHELE D.F. "Analog to Digital / Digital to Analog Conversion Techniques"
John Wiley and Sons, New York, 1968.

(57) HOFFMAN A.A.J.; FRENCH R.L.; LANG G.M. "Minicomputer interfaces: know more, save more"
IEEE Spectrum, págs. 64 a 68, febrero 1974.

(58) HOLT R.M.; LEMAS M.R. "Current Microcomputer Architecture"
Computer Design, Vol. 13, Nº 2 págs. 65 a 73, febrero 1974.

(59) HUSSON S.S. "Microprogramming-Principles and Practices"
Prentice-Hall 1970.

(60) INCERTIS F. "Diseño de una calculadora Digital Programable" tomos 1,2 y 3.
Centro de Investigaciones Técnicas de Guipuzcoa, sept. 1973.

(61) INCERTIS F. "Diseño de una calculadora Digital Programable. Tres aspectos fundamentales" Tesis Doctoral.
E.T.S.I.I. San Sebastian, Universidad de Navarra, 1973.

(62) INCERTIS F. "Métodos algorítmicos de síntesis de funciones para calculadoras digitales"
II Congreso Nacional de Automática - Barcelona, Asociación Española de Automática e Informática, págs. 1001, 1972.

(63) INTEGRATED COMPUTER SYSTEMS "Microprocessors and Microcomputers" Course Notes.
Paris 1-8- abril 1975.

(64) INTEL " Data Catalog"
Intel Corp.-California, 1975.

(65) INTEL "MCS-8 User's Manual"
Intel Corp.-California, agosto 1973.

(66) INTEL "MCS-8 Microcomputer Set-8008, 8 bit Paralell Central Processor Unit". Rev. 4.
Intel Corp.-California, noviembre 1973.

(67) INTEL "THE Intel Memory Design Handbook"
Intel Corp.- California, agosto 1973.

(68) JACOBSOHN D.H. "A Combinatori Division Algorithm for Fixed-Integer Divisors"
IEEE Transactions on Computers, págs. 608 a 610, junio 1973.

(69) JAMES M.L.; SMITH G.M.; WOLFORD J.C. "Métodos Numéricos Aplicados a la Computación Digital con FORTRAN"
International Textbook Co. Mexico. 1970.

(70) JARVIS R.A. "A General-Purpose Hybrid interface for a Minicomputer"
Simulation, Vol. 22, Nº 4, págs. 107 a 112, abril 1974.

(71) KANABY K.D.; ATKINS D.E. "A Shared-Memory Micro-Mini Computer Systems for Process Control"
IEEE Compcon Fall, págs. 5 a 10, 1974.

(70 bis) JURADO F.; PRIETO A. "Definición y realización de un ensamblador cruzado para microordenador"
(en preparación). 1976.

(72) KLAPEISH M. "Interfacing the Teletypewrinter Part1: A-D and D-A Converters"
Computer Design, Vol. 13, Nº 6, págs. 94 a 98, junio 1974.

(73) KLEIR, RAMANOORTHY "Optimization Thectniques for Microprograns"
IEEE Transactions on Computers, julio 1971.

(74) KNUTH, D.E. "The Art of Computer Programming: Seminumerical Algorithms"
Addison-Wesley, Vol. 2. 1969.

(75) E.U. KRISHNAMURTHY "On Optimal Iterative Schemes for High Speed Division"
IEEE Trasaction on Computers, págs. 227 a 231, Marzo 1970.

(76) LANGLEY F.J. "Small Computer design using microporgramming and multifunction LSI arrays"
Computer Design, Vol. 9, págs. 151 a 157, 1970.

(77) LALIOTIS T.A. "Main Memory Technology"
Computer, Vol. 6, Nº 8, págs. 21 a 27, Agosto 1973.

(78) LEWANDOWSKI R. "The Key to Success with Minoprqcessors"
Electronics, pgs. 101 a 106, marzo 1975

(79) LEWIN D. "Theory and Design of Digital Computers"
Nelson-London, 1973.

(80) LOVERING W.F.; BYWATER R.E.H. "An all Digital Hybrid Computer"
1974 Conferences on Computer Systems and Technology, IEE London 28-31 octubre 1974, págs. 49 a 57, 1974.

(81) MANO "Computer Logic Design"
Prentice-Hall, 1972.

(82) MARASA J.D.; MATULA D.W. "A Simulative Study of Correlated error propagation in varions finit-precision arithmetics"
IEEE Transaction on Computers, Vol. C-22, nº 6, págs. 587 a 597, junio 1973.

(83) McCRACKEN D.D.; DORN W.S. "Métodos Numéricos y programación FORTRAN"
Limusa Wiley - Mexico, 1973.

(84) METROPOLIS N.C. "Analyzed Binary Computing"
IEEE Transactions on Computers, Vol. C-22, Nº 6, págs. 573 a 576, junio 1973.

(85) MILLS D.L. "Executive Systems and Software Development for Minicomputers"
Proceedings of the IEEE.
Vol. 61, Nº 11, págs. 1556 a 1652, Nov. 1973.

(86) MOFFA R. "Interfacing Peripherals in Mixed Systems"
Computer Design, Vol. 14. Nº 4, págs. 77 a 84, abril 1975.

(87) MORENO E. "Generador periódico y sincrónico de funciones programables"
Tesis doctoral Dto. de Electricidad y Electrónica, Universidad de Granada, 1973.

(88) MOTOROLA SEMICONDUCTOR "The M6800 microcomputer Family M6800 Systems reference and data Sbeets"
1974.

(89) NATIONAL "Digital Integrated Circuits"
National Semiconductor Corporations-California, Agosto 1973.

(90) NICOUD J.D. "Standardized Mnemonics and Software support for Microprocessors"
IEEE Proc. 75 Iscas, págs. 1 a 4, 1975.

(91) OLIPHANT J. "Designing with Intel's Static Mos Rams"
Aplication Note AP-8 Intel-Santa Clara, 1974.

(92) PARASURAMAN B. "Aplication of LSI Processors"
73 Wescon Professional Program Proceedings, IEEE, Nº 11/2, 1973.

(93) PEATMAN J.B. "The Design of Digital Systems"
McGraw-Hill, 1972.

(94) POKOSKI, J.L.; HOLT O. "Developing Software for Microcomputer Application"
Computer Design, Vol. 14. Nº 3. págs. 88 a 90, marzo 1975.

(96) RATTNER J.C.; CORNET M.E. HOFF. "Bipolar LSI Computing elements usher in new era of digital design"
Electronics, sep. 5, 1974.

(95) POPPER C. "Smal- a Structures Macro-Assembly Language for a Microprocessor"
IEEE Compcon Fall 1974, págs. 147 a 151.

(97) RICHARDS R.K. "Digital Design"
Wiley-Interscience, 1971.

(98) ROBERTSON J.E.; TRIVEDI K.S. "Th Status of Investigations into computer Hardware Design Based on the use of continued fractions"
IEEE Transactions on computers, Vol. C-22, Nº 6, págs. 555 a 560, junio 1973.

(99) ROSIN R.F. "Contemporary Conceps of Microprogramming and Emulation"
Computins Surveys, Vol. 1, Nº 4 dec. 1969.

(100) ROSIN R.F. "The significance of microprogramming"
International Computing Symposium 1973, North-Holland Publ. Co., págs. 237 a 242, 1974.

(101) RUBIN A.I. "Analog/Hybrid-What it was, what it is, what it may be"
AFIPS Conference Proceedings, AFIPS Press-New Jersey, Vol. 37, 1970.

(102) SANCHEZ-IZQUIERDO J. "Ordenadores como elementos de un sistema"
Revista de Automática. Vol, 2, Nº 4. págs. 39 a 46, 1969.

(103) SANKAR P.V.; CHAKRABARTI S.; KRISHNAMURTHYE V. "Aritmetic Algorithms in a Negative Base"
IEEE TRansactions on Computer, Vol. C-22, Nº 2, págs. 120 a 125, febrero 1973.

(104) SCHWARTZ M. "Information Transmission, Modulation, and Noise"
McGraw-Hill Co. New York, 1970.

(105) SHANNON C. "The Philosophy of Pulse Code Modulation"
Proc. IRE, págs. 1328 a 1331, nov. 1948.

(106) SHRIVER B.D. "Microprogramming and Numerical Analysis"
IEEE Transaction on Computers, junio, 1971.

(107) SCHMID H. "Monolithic Processors"
Computer Design, Nº octubre, 1974.

(108) SCHIMID H. " Electronic Analog/Digital Conversions"
Van Nostrand Reinhold Co., 1970.

(109) SCHOEFFLER J.D.; BRONNER L.R. "Data Mangement Software for Minicomputer Production Monitoring and Control Systems"
Proceedings of the IEEE, Vol, 61, Nº 11, págs. 1563 a 1570 noviembre 1973.

(110) SIMOONS M.L.; SMALLENBURG "On-Line Analysis of Exercises Electrocardiograms"
Computer, Vol. 8, Nº 7, págs. 42 a 45, julio 1975.

(111) N. STEINBERG "Machines Analogiques et Hibrides"
Armand Colin Paris, 1969.

(112) STONE "Introduction to Computer Organization and Data Structures"
MacGraw-Hill, 1972.

(113) TELEDYNE PHILBRICK "Teledyne Philbrick Product Guide"
Massachusetts, Agosto 1973.

(114) TEXAS INSTRUMENTS FRANCE "SBP0400 Processeur Binaire 4 bits paralell"
enero 1975.

(115) TEXAS INSTRUMENTS "The Integrated Circuits Catalogue for Design Engineers"
Texas Instrumets Incorporated-Dallas, octubre 1972.

(116) THEIS "Microprocessor and Microcomputer Survery"
Datomatren, págs. 90 a 100, diciembre 1974.

(117) THOMAS L.J.; BLAINF G.J.; "Continnous Monitoring of Physiologic variables with o Dedicatece Minicomputer"
Computer, Vol. 8, Nº 7, págs. 30 a 35, julio 1975.

(118) TORRERO E.A. "Focus on Microprocessors"
Electronic Design, Vol. 18, págs. 52 a 61, sept. 1974.

(119) TRASK,M. "The Story of Cybernetics"
Dutton Pictureback New-York, 1971.

(120) WEISSBERGER A.J. "MOS/LSI Microprocessor Selection"
Electronic Design, Vol.12. págs. 100 a 104, 7 junio 1974.

(121) WIDROW B. "A Study of Rough Amplitude Quantization by Means of Nyquist Sampling Theory"
Trans. IRE. Vol. CT-3, págs. 266 a 276, diciembre 1956.

(122) WILKES, M.V.; RENWICK,W.; WHEELERD.J. "The Design of the Control Unit of an Electronic Digital Computer"
Proc. IEE London, Vol 105, págs. 121 a 128, marzo 1958.

(123) WILKES M.V. "Microprogramming"
Proc. AFIPS, Vol. 12, Nº EJCC, págs. 18 a 19, 1958.

(124) WILKES, M.V.; STRINGER J.B. "Microprogramming and the design of the control circuits in an electronic digital computer"
Proc. Cambridge Phil. Soc. pt 2.
Vol. 49, págs. 230 a 238. abril 1953.

(125) YAEGER,R. "Microprogramming: A general Design Tool"
Computer Design, págs. 150 a 157, agosto 1974.

(126) YOHE,J.M. "Roundigns in Floatins-Point Arithmetic"
IEEE Transctions on Computers, Vol. C-22, Nº 6, págs. 577 a 586, junio 1973.

(127) ZOHAR,S. "A/D Conversion for Radix (-2)"
IEEE Transactions on Computers, Vol. C-22, Nº 7 págs. 698 a 701, julio 1973.

Reunido el Tribunal examinador en el día de la fecha, constituido por:

D. Antón Civit Breu
D. Bernardo Farrés Olmedo
D. Salvador Bracho del Pino
D. José Mira Mira
D. José Miró Mirabent

para juzgar la Tesis Doctoral del Licenciado Don Alberto Prieto Espinosa

se acordó por unanimidad otorgar la calificación de sobresaliente cum Laude

y para que conste, se extiende firmada por los componentes del Tribunal, la presente diligencia.

Granada, a 14 de mayo de 1976

El Secretario,

El Presidente,

El Vocal,

El Vocal,

El Vocal,