

UNIVERSIDAD POLITECNICA DE CANARIAS

E. U. DE TELECOMUNICACIONES

PROYECTO FIN DE CARRERA:

PROGRAMADOR DE EPROMS 27XXX

AUTOR = José Manuel García Martel.

TUTOR = Sebastián Suárez Gil.

LAS PALMAS DE G.C. FEBRERO-1987

TUTOR

AUTOR

INDICE

CAPITULO I

Introducción y objeto de proyecto.....2

CAPITULO II

Condiciones iniciales de las EPROMs.....9

CAPITULO III

Características técnicas de programación.....12

CAPITULO IV

Visión general del programador.....20

CAPITULO V

Programando EPROMs con MSX.....25

CAPITULO VI

El sistema autónomo.....38

CAPITULO VII

Alimentación del sistema76

APENDICE A

Características de EPROMs.....81

LISTADOS

- Del programador MSX.....130

- Del programa monitor.....142

- Del habilitador de CI.....171

ESQUEMAS

.....175

C A P I T U L O I

INTRODUCCION Y OBJETO DEL PROYECTO.

INTRODUCCION Y OBJETO DEL PROYECTO

1.1. GENERALIDADES

La intención con que ha sido concebido este proyecto, es la de facilitar a los posibles diseñadores de sistemas de un medio económico de programar memorias EPROMS. Para ello ha sido elegido como principal medio para dicho cometido los populares ordenadores personales basados en el sistema MSX.

Este sistema se ha introducido en el hogar de muchos programadores y aficionados a la informática, debido quizás a la principal característica que presentan "la compatibilidad". Pues si, ocho fabricantes entre ellos tan importantes como SONY, SANYO, PHILIPS; se han puesto de acuerdo para contruir sus ordenadores bajo una norma común, que permita a cualquier unidad central comunicarse con periféricos de otras marcas, salvando así los problemas que presentaban al comprar un determinado ordenador, el cual se quedaba limitado en el número de sus periféricos, bien al no existir en el mercado, o bien, debido al excesivo precio de los mismos.

No solo han normalizado los buses de expansión sino que también un sistema operativo, el cual también es común al sistema, y desarrollado por MICROSOFT,

INTRODUCCION Y OBJETO DEL PROYECTO

con lo cual los programas realizados por cualquier compañía dedicada a tal fin pueden ser ejecutados en cualquier ordenador que pertenezca al sistema MSX.

Por dicha razón, los principales componentes que forman el hardware de los MSX son iguales y no sólo tienen esta característica sino que son de fácil adquisición en cualquier comercio especializado en la venta de componentes electrónicos. Los principales CI son:

- CPU: Z 80
- VDP: TMS 9929A. Procesador de video.
- AY-3-8912. Generador de sonido.

Característica esta a tener en consideración, si tenemos en cuenta que existen marcas que utilizan circuitos integrados, los cuales, son fabricados en exclusiva para ellos. Tal es e caso de los ordenadores ATARI, el cual al fracasar la fabrica, los componentes que utilizaba dicho ordenador ha dejado de ser fabricados. Con lo cual las posibilidades de reparación se encuentran mermadas.

El amplio campo de prestaciones que presenta tanto en hardware como en software los MSX ha sido quizás la razón de su fuerte implantación el mercado.

INTRODUCCION Y OBJETO DEL PROYECTO

Al estar tan introducido los MSX creo cubrir, con la realización práctica de este grabador, un gran hueco en un laboratorio de electrónica el cual se precie de tal y más dado que cuando muchos alumnos al terminar sus estudios en esta Escuela, se encuentran imposibilitados al traslado de las dependencias de este centro por razones que sobran nombrar aquí. Con lo cual espero contribuir al desarrollo de la formación aquí adquirida.

Aún en el caso de que el posible usuario no posea un ordenador personal, este mismo proyecto desarrolla un sistema autónomo el cual permite la introducción de datos, así como la grabación de los mismos. Este sistema autónomo es ajeno al ordenador MSX, tan sólo comparte el zócalo de inserción nula. Este sistema permite la grabación de las pequeñas memorias que constituye las 2716, con una capacidad de 2 Kbytes.

Con ello, éste proyecto cubre las condiciones requeridas por un buen dispositivo grabador de EPROMS y no solo por las diversas posibilidades que permite para la introducción de datos sino los los diferentes tipos de memorias que permite programar, ya que, los que se

INTRODUCCION Y OBJETO DEL PROYECTO

comercializan habitualmente sólo permiten la grabación de las 2716 y 2732.

1.2. Tipos de memorias para las cuales ha sido concebido este proyecto.-

Las memorias que permite grabar el dispositivo aquí desarrollado, son las archiconocidas 27XX y 27XXX, ya que son las que con mayor frecuencia y facilidad se pueden encontrar en cualquier comercio. Sin embargo, por las características comunes que presentan otras familias, es posible también, la grabación de dichas equivalentes. Para ello, el usuario comprobará las características técnicas de grabación que presenta la homóloga con las que aquí se estudian y si fuesen las mismas, tan solo bastaría elegir en el momento de la elección la memoria equivalente.

En la memoria de este proyecto se incluye un apéndice en el cual constan las principales características de las memorias 27XXX. Debo recomendar al posible usuario la lectura del mismo, ya que la considero de vital importancia, para una adecuada programación de la memoria. Dada la posibilidad de

INTRODUCCION Y OBJETO DEL PROYECTO

aplicar por confusión una tensión distinta, a la correcta, a las patillas de las diferentes memorias y en especial consideración la de Vpp, ya que unos pocos voltios de más aplicados en éstas patillas de control puede dañar definitivamente la memoria.

C A P I T U L O I I .

CONDICIONES INICIALES DE LAS EPROMS

CONDICIONES INICIALES DE LAS EPROMS

Quiero realizar en este capítulo una descripción de las condiciones previas que ha de reunir las EPROMS para su correcta grabación. Dichas condiciones son principalmente:

1) Que el integrado que constituye la memoria se encuentre limpia de residuos y principalmente los grasientos, que dificultan en gran medida la correcta grabación.

2) Todas las celdas que constituyen la memoria han de estar en el estado lógico " 1 ", ello se consigue con un buen borrado. Si como se ha dicho, inicialmente y después de cada borrado, todos los bits de las diferentes EPROMS están en el estado " 1 " lógico, la información es introducida colocando un cero en el bit seleccionado. De esta forma un " 0 " (grabado), sólo puede ser cambiada a " 1 " mediante la aplicación de luz ultravioleta.

Esta característica de borrado, es tal que, el borrado empieza a ocurrir cuando es expuesta, las diferentes memorias a una luz de longitud de onda corta de aproximadamente 4.000 Angstroms.

Tanto la luz solar como las de tubos fluorescentes tienen su longitud comprendida en un rango

CONDICIONES INICIALES DE LAS EPROMS

entre 3.000 y 4.000 Angstroms, si la EPROM es expuesta a esta radiación con su ventana abierta el borrado dura aproximadamente tres años, si bien, si fuese expuesta directamente a las radiaciones solares el dispositivo puede estar borrado en una semana.

Las recomendaciones para un efectivo borrado es que la EPROM sea expuesta por su ventana a las radiaciones de una luz ultravioleta de longitud de onda 2.537 Angstroms. La radiación mínima requerida es de 15 W sec / cm.².

El tiempo empleado por una lámpara que proporcione un poder de radiación de 12 uW sec. / cm.² es entre 15 y 20 minutos.

Si se aplicasen lámparas con un poder de radiación de 7.258 W sec. / cm.² o superiores podrían dañar definitivamente la EPROM.

C A P I T U L O I I I
C A R A C T E R I S T I C A S T E C N I C A S D E P R O G R A M A C I O N

CARACTERISTICAS GENERALES DE PROGRAMACION

Según el tipo de EPROM a programar habrá un número variable de pines en el encapsulado de las distintas EPROMS. Con lo que variará de lugar ciertos pines de control, así como la polarización y los pines correspondientes al direccionamiento.

Para una mejor y más fácil comprensión del desarrollo del hardware adicional a la programación de las diferentes EPROMS, considero necesario una descripción exhaustiva de sus características generales, si bien en el apéndice final vienen reflejadas las características para cada tipo en concreto.

Las características generales de la programación son las siguientes, si bien, según el modelo, aún teniendo la misma capacidad, puede variar en sus características.

Tenemos dos métodos o algoritmos diferentes de programación según el tipo de memoria elegida. Dichos métodos son los siguientes:

Método 1) Los datos de programación se introducen en palabras de ocho bits, a través del bus de datos, un tiempo antes de efectuarse la escritura (Tset up), aproximadamente 10 nanosegundos. Después del

CARACTERISTICAS GENERALES DE PROGRAMACION

direccionamiento y estando presente los datos se aplica un pulso de programación de 50 msg. en el pin CE.

Para la programación de cada dirección se aplica un pulso de programación por dirección a grabar.

Normalmente, después de un ciclo de programación, se tiene un ciclo de verificación, en el cual se comprueba la dirección y el dato grabado. En este ciclo de verificación la tensión V_{pp} es de 5 V.

En los cronogramas que se adjuntan en el apéndice final vienen reflejados los diferentes tiempos de cada tipo, si bien, el tiempo que debe permanecer una dirección, en el pin correspondiente viene determinada principalmente por la anchura de pulso de programación que puede oscilar entre 45 y 55 msg.

Sin embargo como puede verse en los cronogramas existen otros tiempos que han de respetarse para obtener una programación correcta, como son, los tiempos de establecimiento, del bus de datos y direcciones antes del pulso de programación, así como los tiempos que tienen que estar los datos y

CARACTERISTICAS GENERALES DE PROGRAMACION

direcciones estables, después del pulso de programación.

| <u>Sím.</u> | <u>Parámetros</u> | <u>min.</u> | <u>tipico</u> | <u>unidad</u> |
|-------------|---|-------------|---------------|---------------|
| Tps | Tiempo (Tset up) que debe permanecer estable el dato antes del pulso de programación. | 10 | - | ns. |
| Tpd | Tiempo que ha de permanecer estable el dato, después del pulso de programación. | 0 | - | ns. |
| Ts | Tiempo de set up del pulso de programación. | 2 | - | us. |
| Tpr | Tiempo de subida del pulso de programación. | 0.01 | 2 | us. |
| Tpf | Tiempo de bajada del pulso de programación. | 0.01 | 2 | us. |
| Tpw | Anchura del pulso de programación. | 45 | 50 | msg. |

Las memorias que emplean este método de programación corresponden a las 2716 y 2732A las cuales emplean un mayor tiempo en el pulso de programación respecto del método que a continuación vamos a

CARACTERISTICAS GENERALES DE PROGRAMACION

desarrollar, conocido por el método del algoritmo inteligente.

Metodo 2) "Algoritmo inteligente"

Al igual que el método anterior los datos de programación se introducen en palabras de ocho bits a través del bus de datos, un tiempo antes de efectuarse la escritura (Tset up), aproximadamente a 10 usg. de duración en la patilla CE o bien en la PGM.

En el diagrama 3.1. se muestra el flujograma de este algoritmo inteligente que es estudiado en este capítulo como modelo para su estudio, si bien, en el apéndice final se puede comprobar que existen ligeras modificaciones entre los diferentes tipos de EPROMS en lo que respecta al algoritmo a emplear. Las diferencias entre dichos algoritmos radica fundamentalmente en el número de implulsos a aplicar.

El algoritmo inteligente de programación es el método que se aplica para un ahorro de tiempo. Es significativamente más rápido que el algoritmo estándar de aplicar pulsos de 50 msg. En el caso de una 2764 el tiempo empleado por el algoritmo estándar suele ser de minuto y medio que puede ser reducido a 1/5 si se emplease el algoritmo inteligente. Este algoritmo

CARACTERISTICAS GENERALES DE PROGRAMACION

emplea dos tipos diferentes de pulsos. Inicialmente y durante un número prefijado como se aplican impulsos de 1 msg. (\overline{CE} , \overline{PGM}) en la correspondiente patilla a aplicar el impulso. A continuación se aplica un impulso de una duración de 4X msg. Donde X es un contador interactivo que equivale al número de impulsos aplicados. Inicialmente tiene el valor 1 y como tope 15; es decir a un bite se le puede aplicar como máximo 15 impulsos de 1 msg. de duración para continuar con un impulso largo que tendría como duración 60 msg. Si todavía al verificar el bite, éste no obtuviera la correcta verificación, el algoritmo pararía la programación para indicar que la EPROM se encuentra en malas condiciones.

Durante la programación la tensión V_{pp} se mantiene en unos valores relativamente altos (21 , 12.5 V.) y baja en el modo de lectura a niveles TTL, al igual que en el método anterior. Pero con el empleo de este algoritmo la tensión de polarización sube de 5 a 6 V. Tal como puede observarse en el flujograma de la figura 3.1.

Una vez hayan sido grabados los datos y verificados, las tensiones V_{cc} y V_{pp} se sitúan a

CARACTERISTICAS GENERALES DE PROGRAMACION

niveles TTL y se realiza una lectura para la verificación del correcto funcionamiento de la EPROM en el modo normal de lectura.

Los valores de los diferentes tiempos que han de ser tenidos en cuenta para la correcta grabación son los siguientes:

| <u>Sím.</u> | <u>Parámetros</u> | <u>min.</u> | <u>tipico</u> | <u>unidad</u> |
|-------------|---|-------------|---------------|---------------|
| Tps | Tiempo (Tset up) que debe permanecer estable el dato antes del pulso de programación. | 10 | - | ns. |
| Tpd | Tiempo que ha de permanecer estable el dato, después del pulso de programación. | 0 | - | ns. |
| Ts | Tiempo de set up del pulso de programación. | 2 | - | us. |
| Tpr | Tiempo de subida del pulso de programación. | 0.01 | 2 | us. |
| Tpf | Tiempo de bajada del pulso de programación. | 0.01 | 2 | us. |
| Tpw | Anchura del pulso de programación. | 0.95 | 1 | msg. |

CARACTERISTICAS GENERALES DE PROGRAMACION

El algoritmo inteligente puede ser aplicado a las memorias 2764, 27128, 27256 y en sus diferentes versiones. Aunque las EPROMS 2764 y 27128 y no sus respectivas versiones, pueden ser programadas aplicando el algoritmo estándar de 50 msg. Este proyecto, tal posibilidad ha sido eliminada, eligiéndose el algoritmo que presenta mayores ventajas.

Para una mayor visión de las características de grabación, o bien, de lectura, se recomienda consultar el apéndice final y más concretamente la versión del tipo de EPROM en cuestión.

C A P I T U L O I V
V I S I O N G E N E R A L D E L P R O G R A M A D O R

Como introducción a los capítulos siguientes, en los cuales se hace un estudio por separado de los dos métodos de grabación que permite este programador. Así mismo se hace un análisis de los elementos comunes a ambos sistemas de programación.

En el diagrama de bloques del programador, podemos diferenciar los siguientes elementos:

- Bloque de alimentación.
- Programador MSX.
- Zócalo de 28 pines.
- Sistema autónomo.
- Selectores de EPROMs.

En este mismo diagrama incluyo el ordenador personal, del cual nos valemos para la programación con un MSX, queriendo con ello clarificar las conexiones del programador.

El programador para su normal funcionamiento requiere distintos niveles de tensión. Pueden observarse en el diagrama de bloques que las diferentes tensiones requeridas son: +5 V, -6 V, +26 V +15 V, +22 V.

Esto es debido a las necesidades de cada tipo de EPROM para su correcta programación. En el capítulo VII se desarrolla debidamente este bloque.

El medio que permite la transferencia de señales entre el ordenador personal y el zócalo es el bloque "programador MSX". Este bloque viene desarrollado en el capítulo V.

Cuando queramos realizar la programación con el ordenador personal hemos de comunicar la placa de montaje con el computador por medio de una cinta de 50 hilos, con sus respectivos conectores hembras a los extremos.

El zócalo de 50 pines es de inserción nula. Permitiendo la introducción de la EPROM en un dispositivo enlazando las señales de ambos programadores.

Estos zócalos poseen una palanquita que en posición vertical la EPROM queda sin presión alguna, por la patillas pertenecientes al zócalo, mientras que si la abatimos los pines de la EPROM se aprisionados por los contactos metálicos.

El bloque perteneciente al sistema autónomo es el medio del cual nos valemos para realizar

programaciones de memorias 2716. Dicho sistema está controlado por un programa monitor, que permite la comunicación de usuario con el mencionado sistema por medio de una serie de 8 displays y un teclado. Este "sistema autónomo" viene estudiado en el capítulo VI.

En el bloque con epígrafe "selectores de EPROMs" se encuentran los diversos conmutadores que son necesarios situar en la posición adecuada, según el tipo de EPROM, para la aplicación de los correctos niveles de tensión, necesarios para la adecuada grabación.

En el diagrama que lleva por título "esquema de conmutadores" podemos apreciar un zócalo de 28 pines, el cual es utilizado conjuntamente por el programador MSX como por el sistema autónomo.

A de tenerse cuidado en la correcta introducción de los pines en la EPROM con los correspondientes al zócalo.

En el mismo diagrama se aprecian una serie de conmutadores (S1,S2.S3) que mediante la selección de los mismos podemos adecuar el tipo de EPROM que va ha realizar la grabación a las señales procedentes de los sistemas de control.

En el diagrama "selector de EPROM" se encuentran representados las pocisiones a seleccionar en los interruptores. En la primera columna se encuentra el modelo de EPROM; en la columna central la posición a seleccionar en el interruptor 1 para el modelo en cuestión; asi mismo en la tercera columna están para el conector dos.

Durante la grabación y lecturas de las diferentes EPROMs es necesario controlar tanto la tensión de polarización como la habilitación de datos (\overline{OE}), el Chip Enable, y \overline{PGM} . Este control está realizado por las líneas correspondientes a los puertos del 8255 (J1), PB2,PB1,PB4 y PB6 que con diferentes valores de tensión a las entradas del controlador de tensiones situan la correspondiente tensión a la salida.

C A P I T U L O V
PROGRAMANDO EPROMS CON MSX

En este capítulo quiero describir en primer lugar las características del Hardware y Software de los ordenadores personales MSX. Descripción ésta necesaria para lograr comprender la implementación de este programador, objeto de este proyecto.

En lo que se refiere al Hardware, decir que los ordenadores MSX tienen como CPU un Z-80A, microprocesador como todos sabemos de 8 bits, y que viene a ser una versión de los populares microprocesadores 8085, si bien el Z-80A incorpora nuevas instrucciones.

Se aporta en este capítulo una fotocopia del esquema electrónico de la configuración interna de uno de estos ordenadores, que ha servido como base del desarrollo de este proyecto. El ordenador personal MSX, en el cual me he apoyado para la realización de este programador es un SPECTRAVIDEO-728. En este diagrama (5.1) ha sido empleado, para conocer las características de las diferentes señales del bus de expansión, queriendo lograr con ello una reducción en los elementos externos del ordenador, que configuran el programador.

Ha de observarse en dicho esquema, que

las señales referentes al direccionamiento, así como señales de habilitación de dispositivos (IORQ, RD, WR) se encuentran todas ellas bufferadas por medio de 74LS367. Sin embargo las señales de datos van directamente desde las patillas del Z-80A al slot de expansión. Todas las señales mencionadas anteriormente son obtenidas por medio del bus de expansión del cual vienen previstos los MSX. Conector (macho) de 50 pines. El diagrama 5.2 nos muestra la distribución de los pines de dicho conector. Los cuales no comentamos aquí, dado que las señales son de sobra conocidas para cualquier microprocesador de este tipo.

Sin embargo creo conveniente la descripción de los puertos de salida que posee el ordenador. La figura 5.3. nos muestra la distribución de los 256 puertos que lo componen.

Los puertos comprendidos entre &H00 a &H78 no están especificados en la información técnica aportada por el fabricante. Aunque con las pruebas realizadas dichas direcciones están utilizadas para la selección de los dispositivos internos. Los puertos comprendidos entre las direcciones &H79 a &H80 están destinadas para la utilización de la tarjeta de 80

columnas. Los comprendidos entre &HB1 al &HB9 son destinados para el control de cassette, siendo los demás puertos, comprendidos desde la última dirección hasta la &HBO ocupadas por el procesador de VIDEO (VDP) que en los MSX es el TMS 9929A, el generador de sonido (AY-3-8912 PSG) y por último un PPI (8255).

Desde la dirección &HBO hasta la &HFF pueden ser utilizadas por el usuario para la ampliación de sistemas.

Los elegidos para el direccionamiento de los puertos que nos ponen en comunicación con el programador son los siguientes.

Para el 8255 (J1):

| | | |
|------|------|-------------------|
| &HE2 | 226D | puerto de control |
| &HE1 | 225D | puerto C |
| &HE0 | 224D | puerto B |
| &HDF | 223D | puerto A |

Para el 8255 (J2):

| | | |
|------|------|-------------------|
| &HD3 | 211D | puerto de control |
| &HD2 | 210D | puerto C |
| &HD1 | 209D | puerto B |
| &HD0 | 208D | puerto A |

Los puertos listados anteriormente cumplen los requisitos que a continuación se describen:

El puerto &HE2 es utilizado para programar al 8255 (J1), realizando por medio de este puerto la programación de los puertos que configuran el 8255, como entrada y de salida.

Los puertos comprendidos entre &HDF al &HE1 son todos programados como de salida, realizando las funciones que a continuación se detallan:

&HDF presenta el byte alto de direcciones en el zócalo, que tiene la finalidad de ubicar la memoria a grabar.

&HE0 presenta el byte bajo de dirección.

&HE1 proporciona las necesarias señales de control que permiten la programación.

De la misma forma los puertos comprendidos entre &HD3 y &HD0 realizan las siguientes funciones:

&HD3 programa al PIO como entrada y salida de los diferentes puertos, que lo configuran.

&HD2 tiene por finalidad la transferencia de datos entre la EPROM y el ordenador.
&HD1 destinado para futuras ampliaciones.
&HDO destinado para futuras ampliaciones.

Descripción de la distribución de la memoria de los MSX.

Los ordenadores personales MSX. para los cuales esta pensado este programador de EPROM posee una capacidad de memoria de 80Kb. distribuidos de la siguiente manera:

16Kb. destinados para el procesador de video.
32Kb. destinados para la ROM.
32Kb. destinados para la RAM del sistema.

El diagrama 5.4 es el Mapa de la distribución de la memoria de los MSX.

La RAM del sistema va desde la posición &HF380 hasta la posición &HFFFF. De la dirección ZHF380, inclusive, hay rutinas en código máquina, que son llamadas por la BIOS de tratamiento de cartuchos. Siendo esta la zona de trabajo del sistema que es empleada para sus operaciones internas.

El área que es destinada como bloque de control de ficheros realiza las operaciones de entrada y salida. El tamaño de este bloque puede ser variado con la función MAXFILES. El límite superior de este bloque esta en la dirección &HF380. Si se reduciere este bloque, la zona liberada puede ser destinada para el almacenaje de programas en código máquina y de datos. La orden CLEAR reduce este área.

En el área de cadena de caracteres se guardan los contenidos de las variables de tipo cadena de caracteres. El tamaño que toma por defecto es de 200 caracteres de longitud. Para la variable de este tipo, sin embargo, podemos aumentar dicha longitud mediante el empleo de las funciones CLEAR.

El área libre, destinada a guardar los programas BASIC, lugar donde se ubicará el programa objeto de este estudio.

La capacidad de esta área es de 28815 bytes, si bien por ser un área libre, puede ser destinada para diferentes fines. A medida que crece el programa BASIC esta área se verá reducida.

El área del programa BASIC es el lugar donde empieza a localizarse el programa.

El área de matrices es el lugar donde se pueden encontrar a las matrices declaradas mediante la orden DIM. En nuestro caso utilizamos esta área para almacenar los datos que serán introducidos en la EPROM a programar o bien donde quedará la información leída de dicha EPROM, que deseamos copiar o leer. La variable utilizada para tal fin la llamo V1 y está dimensionada de la forma V1 (7,255) y declarada como variable entera. El volumen de esta matriz es de 8*256 bytes (2Kb.), teniendo en cuenta que la matriz comienza por 0 y acaba en 7 y 255 respectivamente.

Los circuitos integrados que componen el Hardware que permite la transferencia de información entre la memoria del MSX y la EPROM a grabar es la siguiente:

| | | |
|----|--------|--------------------------------------|
| J1 | 8255 | puerto programable de entrada/salida |
| J2 | 8255 | puerto programable de entrada/salida |
| J3 | 74LS20 | Dos puertas NAND de cuatro entradas. |
| J4 | 74244 | Buffer de salida. |
| J5 | 74LS06 | Seis inversores. |

El diagrama 5.5 es el esquema electrónico de las conexiones de los CI anteriormente citados. Se han omitido en dicho esquema los conectores, selectores y condensadores, con el fin de no hacer muy engorroso el esquema. Los condensadores tienen por capacidad 2.2 F en los diferentes integrados, conectados entre las patillas Vcc y GND, con el fin de mantener estable la tensión de polarización.

La selección de los PIO se realizan con las señales de direccionamiento A7, A6, A5 y A4, siendo estas señales activas a nivel alto. La señal IORQ indica el momento en el cual el microprocesador realiza la operación de entrada o salida, habilitando con ello el dispositivo en el momento oportuno para la comunicación

La selección se realiza de la siguiente forma: Supongamos que queremos seleccionar el PIO indicado en el esquema como J1 que es un 8255. Este integrado tiene como líneas de direccionamiento A7, A6 y A5; y como selección del puerto las señales A1 y A0.

Llevando las señales A7, A6, A5 y IDRQ a una puerta NAND (J3) tendremos a la salida la señal que habilita al PIO.

Las señales de direccionamiento pueden ser aplicadas directamente desde el ordenador a esta puerta NAND, sin embargo, la señal IDRQ por ser activa a nivel bajo debe pasar previamente por un inversor de los que componen el integrado J5. De igual forma, se selecciona el PIO (J2), excepto que en lugar de aplicar A5 aplicaremos A4.

Junto a la habilitación de los PIO deben añadirse las señales WR y RD procedentes del ordenador y conectadas en paralelas a ambos PIO, indicando el ciclo en el que se encuentra ya sea de escritura o de lectura.

El puerto C de J1 que proporciona las señales de control para la programación de la EPROM. Las funciones que realizan cada uno de las ocho líneas que configuran dicho puerto son las siguientes:

- PC7 Sin definir.
- PC6 Proporciona pulsos de programación en la.

grabación de las 2764 y 27128.

- PC5 Habilita las salidas de las EPROM, es decir controla la entrada OE.
- PC4 Controla la función de polarización.
- PC3 Sin definir.
- PC2 Destinada a proporcionar la tensión adecuada a cada ciclo en la patilla Vpp., por tanto controla tres tensiones distintas. Además realiza la función OE la 2732A.
- PC1 Idem. al anterior.
- PC0 Realiza la función CE, que tiene por finalidad el envío de pulsos de programación para las 2716, 2732A y 27256. Y la selección en el modo de lectura de todas las EPROM.

El Software que controla la programación de la EPROM así como el programa monitor está desarrollado en BASIC MSX VERSION 1.0. El programa ocupa un volumen de 16.5 Kb.

Una vez cargado el programa y habiéndolo ejecutado nos presenta un menú con cinco posibles.

opciones:

1. PROGRAMAR MEMORIA
2. VERIFICAR PROGRAMACION
3. LECTURA DE MEMORIA
4. COPIAR MEMORIA
5. EXIT

La opción 1 permite la toma de datos desde el teclado. Una vez completada la introducción de datos que deseamos programar, pasa a la fase de programación. Utilizando para ello el procedimiento Stándar para las 2716 y 2732 o el algoritmo inteligente para las 2764, 27128 y 27256.

Los procedimientos aquí señalados son descritos en la debida medida en el apéndice final.

La opción 2 realiza la verificación del contenido de la EPROM. Inicialmente los datos son introducidos a través del teclado. Una vez cubierta las direcciones a verificar o el bloque de 2Kb. éste pasa a la fase de lectura y comparación de los datos introducidos con los existentes en la EPROM. En caso de

no ser verificado como de serlo lo indicará oportunamente.

La opción 3 realiza la lectura de datos de la EPROM entre dos direcciones definidas previamente y los visualiza en la pantalla en series de 16 bytes. Apretando cualquier tecla nos representa los siguientes 16 bytes y así sucesivamente hasta la representación de todos los datos comprendidos entre las direcciones.

La opción 4 permite copiar una EPROM y grabar el contenido de esta en otra.

La opción 5 nos devuelve al BASIC MSX.

En cualquiera de las opciones anteriores los datos leídos no serán superiores a 2Kb. debiendo ser realizada la lectura, escritura o grabación en distintos bloques dependiendo ello de la capacidad de la EPROM.

C A P I T U L O V I

E L S I S T E M A A U T O N O M O

Introducción al sistema autónomo

Independientemente del sistema constituido por el MSX, tenemos otra posibilidad de programar memorias. Aunque su orientación esta predispuesta a pequeñas memorias (2716), dado que el sistema está limitado por su capacidad de memoria RAM puede realizar la programación de modelos mayores si realizamos la introducción de datos en distintos bloques. Este método estaría dado a cometer errores por la gran cantidad de datos a introducir. El listado se encuentra al final de este capítulo.

El sistema está provisto de un programa monitor que permite la comunicación de dicho sistema con el posible usuario.

Como todo sistema tiene la posibilidad de ser destinado a otros menesteres. Se ha tenido la precaución de dotar en la parte intermedia del programa monitor una zona libre para los posibles cambios que pudiera requerir el usuario.

Este programa monitor esta ubicado en una memoria 2716 la cual posee 2Kb. sin embargo el programa monitor ocupa 1.3 Kb. libres, lo que posibilita unos programas de 0.7 Kb.

También esta prevista la expansión del sistema por medio de cuatro puertos de ocho bits y dos de 6 bits siendo dichas puertas programables. Estos puertos están implementados en dos chips 8155.

En los capítulos siguientes vamos a desarrollar las cuestiones de interés para el manejo del sistema. Para ello estudiaremos el ámbito del Hardware y la situación de los elementos importantes para el usuario. Más tarde se estudiará el uso de los diferentes comandos.

Estudio del Hardware

A diferencia de los computadores MSX que utiliza como CPU un Z-80, el sistema autónomo emplea un 8085A. La compatibilidad de ambos sistemas se consigue utilizando las entradas que dejan a los dispositivos en alta impedancia. En el caso que nos aborda para habilitar el sistema situaremos el interruptor SA, correspondiendo esta a dejar a los 8255 en estado de alta impedancia.

Como el controlador del teclado y los display empleo un 8279 que permite la intercomunicación entre el sistema autónomo y el usuario.

Una serie de 8 displays, un teclado compuesto por 21 teclas, una EPROM 2716 donde se ubica el programa monitor; una nueva 2716 permite la habilitación de los diferentes dispositivos, el listado que compone la información de dicha EPROM se adjunta al final del capítulo. Esta memoria EPROM está bajo el control del 8085.

Las 21 teclas que componen el teclado están dispuestas de la siguiente forma:

- 16 que constituyen un teclado hexadecimal.
- 2 para el control del cursor.
- 2 para el cierre de línea. Las cuales permiten incrementar y decrementar las posiciones de memoria que visualizan los displays.
- 1 tecla destinada para inicializar el sistema (Reset) y de la cual ya se hablará más adelante.

La memoria RAM que compone el sistema autónomo está compuesta por dos memorias 2114, y por la memoria que aportan los 8155.

Por último componentes pasivos son los que completan el Hardware.

Distribución del espacio de direcciones

En el sistema autónomo existen dos zonas de direccionamiento: la zona correspondiente a las instrucciones relativas a memorias y la zona de direccionamiento de entrada/salida.

La zona de direccionamiento de memoria comprende 64Kb. En el mapa de memoria que a continuación se ilustra pueden verse las zonas ocupadas y las zonas libres. El espacio comprendido desde el 0000 hasta 07FF está ocupado por el programa monitor. A este espacio se le une el espacio reservado para la EPROM que se desea grabar. Para la 2716 por tanto comprenderá las direcciones 0800 a 0EFF. El espacio comprendido entre la dirección 1800 a 1BFF que esta reservada a la RAM. Existe por otra parte una dirección entre la dirección FF00 y FFFF, que esta destinada para la RAM de extensión.

La parte de esta zona indicada en el diagrama está destinada para el monitor y no puede ser utilizada por el usuario. Así mismo hay una zona disponible entre las direcciones 1C00 a la 1CFF.

El espacio de entrada/salida comprende 256 direcciones. Una parte de ellas es utilizada por los

circuitos integrados que componen el aparato. En el mapa de memoria puede observarse dicha distribución.

La zona entre EE y EF corresponde al B279. La zona 7F a F8 corresponden a los puertos de la RAM. La zona de direccionamiento de puertos no ocupada asi como la de memoria pueden ser utilizadas para futuras ampliaciones.

A continuación hacemos una descripción de los comandos que permiten el empleo de este sistema.

DESCRIPCION DE LOS COMANDOS

DESCRIPCION GENERAL DE LOS COMANDOS

El programa monitor puede ejecutar cuatro funciones bajo orden del usuario. Estas funciones son introducidas a través del teclado mediante comandos.

Los comandos relativos a la memoria permiten la lectura y escritura en la memoria interna. Para ello, la dirección se puede incrementar o decrementar a voluntad. Además, una sola memoria puede llenarse con una constante y el contenido de una zona de la memoria puede ser trasladada a otras zonas. Un comando adicional ofrece una sensible ventaja con respecto al traslado de zonas de memorias. Después de la translación las direcciones de salto en relación a la zona trasladada ya no coinciden. El programa monitor contiene un comando que comprueba si las direcciones de salto están acorde con la zona trasladada. Para las órdenes que se refieren al salto, esta dirección de salto es renovada automáticamente; todos los comandos de este grupo son utilizados para la programación de EPROMS si el correspondiente interruptor de tensión está en la posición indicada en este mismo proyecto.

Para la rápida orientación de los comandos disponibles ver el diagrama.

a) representación de los comandos y sus funciones.

| Símbolo del comando | Función |
|---------------------|--------------------------------------|
| 0 | Leer y escribir en memoria |
| 1 | Llenar zona de memoria con constante |
| 2 | Traslado de bloques de memoria |
| 3 | Cambio de direcciones |

b) Símbolos del campo de dirección y su función.

Símbolos para parámetros

AA Dirección de comienzo
EA Dirección final
BA Dirección de destino
SA Dirección de inicio
EU Dirección final de la zona de la memoria con direcciones a modificar.

CAMBIO AUTOMATICO DE DIRECCION.

Después del traslado de la información en un programa del usuario en una zona de memoria a otra puede producirse errores en la ejecución del programa si no ejecutan cambios adicionales. Vamos a explicarlo con un ejemplo:

El diagrama contiguo muestra dos bloques de programas A y B en memoria, ambos tienen órdenes de salto en una dirección del bloque B. Si los traslados del bloque B a otra zona de la memoria indicada por el cuadro dado: b) la orden de salto sigue señalando a la dirección de la antigua situación del bloque B. Esto no es deseado porque la zona de la memoria en la cual se encontraba el bloque B debe ser usada para operaciones internas, y por su traslado el orden lógico de las órdenes no debe ser alterado. Para conseguir esto deben ser cambiadas todas las direcciones de salto en todos los bloques de programas que contengan órdenes de salto. En el ejemplo esto no es exigible tanto para el bloque A como para el bloque B. Después de la modificación las direcciones de salto correctas se encuentran en el diagrama.

Una ejecución manual de este cambio de direcciones exige mucho tiempo y muestra facilidad para cometer errores. El sistema ofrece un comando que ejecuta esta tarea. Para ello hay que suministrar la dirección de la zona a la que se quiere trasladar. Además se necesitan las direcciones de comienzo y final de la zona de memoria en la cual debe ser modificada las direcciones de salto. El comando debe ser aplicado a todas las zonas de memoria que contengan órdenes de salto a la zona de memoria trasladada.

a) Posición primitiva

| Dirección | instrucción | |
|-----------|-------------|----------|
| 1 8 0 0 | | Bloque A |
| 1 8 0 1 | | |
| 1 8 0 4 | JMP 180DH | |
| 1 8 0 5 | | |
| 1 8 0 6 | | Bloque B |
| 1 8 0 7 | JMP 180DH | |
| 1 8 0 A | | |
| 1 8 0 D | MOV A, B | |
| 1 8 0 E | | |

b) Posición trasladada con direcciones de salto incorrectas.

Dirección instrucción

| | | |
|---------|--|-----------|
| 1 8 0 0 | | JMP 180DH |
| 1 8 0 1 | | |
| 1 8 0 4 | | |
| 1 8 0 5 | | |

Bloque A

| | | |
|---------|---|---|
| . | . | . |
| . | . | . |
| 1 8 0 D | . | . |
| . | . | . |

Posición anterior
bloque B

| | | |
|---------|--|-----------|
| 1 8 1 6 | | JMP 180DH |
| 1 8 1 7 | | |
| 1 8 1 A | | MOV A,B |
| 1 8 1 D | | |
| 1 8 1 E | | |

Nueva
posición
del bloque
B

c) Posición trasladada con direcciones de salto correctas.

Dirección instrucción

| | | |
|---------|--|-----------|
| 1 8 0 0 | | JMP 181DH |
| 1 8 0 1 | | |
| 1 8 0 4 | | |
| 1 8 0 5 | | |

Bloque A

.
 . 1 8 0 D

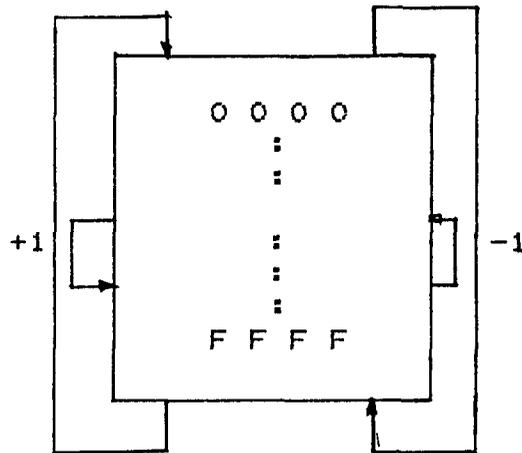
Posición anterior
 bloque B

| | | |
|---------|--|-----------|
| 1 8 1 6 | | JMP 181DH |
| 1 8 1 7 | | |
| 1 8 1 A | | MOV A,B |
| 1 8 1 D | | |
| 1 8 1 E | | |

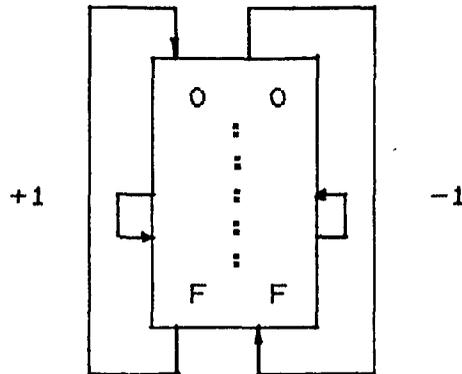
Nueva
 posición
 del bloque
 B

Incremento y decremento de direcciones

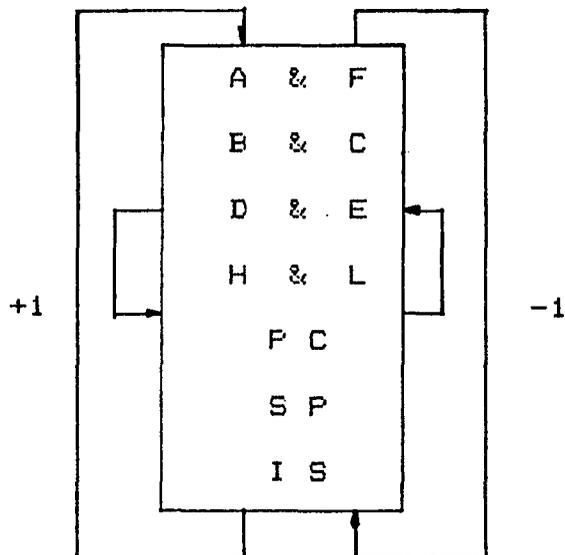
a) Direcciones de memoria:



b) Direcciones de E / S:



c) Nombre de registros:



INTRODUCCION DE UN COMANDO

Después de conectar la alimentación o de apretar la tecla reset, el display aparece apagado, excepto el punto del campo de comando. El sistema espera la introducción de un símbolo de comando 0,1,2,3. Si se pulsa una tecla no permitida para símbolo de comando, ésta es ignorada y no tiene consecuencias. Si se introduce un símbolo de comando válido a través del teclado, éste aparece en el campo de comandos. Si se introdujeran a continuación otros símbolos de comandos válidos, estos serían sobre escritos. El símbolo de comando no es tomado como válido hasta que sea pulsada la tecla de cierre ↑ o ↓.

Un comando en ejecución puede revocarse en cualquier momento mediante la tecla R. A continuación puede introducirse un nuevo comando. También se puede revocar un comando llevando el cursor al campo de comandos mediante una de las teclas de desplazamiento (←, →), y sobrescribiendo el comando ya existente. En cuanto se introduce un símbolo de comando se oscurece el campo de dirección y de datos. Apretando una de las teclas de cierre, el símbolo de comando se toma como válido. La diferencia entre las dos posibilidades de

revocar un comando reside fundamentalmente en que al apretar la tecla R se interroga al interruptor de selección de programas; se deshabilita la interrupción del programa del usuario y que los valores propuestos para parámetros son tomados como fijos; mientras que en el otro caso esto no sucede.

ESTUDIO DE LOS DIFERENTES COMANDOS

El grupo estudiado a continuación contiene cuatro comandos que permiten la escritura de nueva información o la modificación de la misma ya existente en la memoria del sistema. Puede ser modificada tanto la información como su lugar en la memoria. Junto con la escritura o lectura de posiciones de memoria también es posible llenar una determinada zona con una constante, el traslado de información entre distintas zonas de la memoria y el cambio automático de las direcciones en las órdenes de salto.

La información no solo puede escribirse en la memoria de escritura / lectura, sino también en el chip de memoria introducida en el lugar de conexión de EPROM n.2, si el interruptor de programación se encuentra en la posición ON. Aunque se usen los mismos

comandos que para escribir en la RAM esta posibilidad es estudiada en un capítulo aparte, ya que observar algunas particularidades.

ESCRITURA Y LECTURA EN MEMORIA

COMANDO O.

Función.

El comando O sirve para leer información de la memoria o también inscribir nueva información. Las direcciones de memoria pueden ser incrementadas o decrementadas automáticamente a partir del último valor, o ser modificada manualmente; la información puede modificarse de la misma manera. En el marco de este comando aparecen al mismo tiempo una dirección de memoria y el dato contenido en ella en el display.

Ejecución.

Para la ejecución normal del comando descrita a continuación se presupone que la información sólo intentará ser escrita en direcciones en las que se encuentra la memoria de escritura / lectura.

El cursor debe encontrarse al principio en el campo de comandos. Se introduce el símbolo de comandos O y se pulsa cualquiera de las teclas de cierre. A

continuación aparecen en el campo de direcciones una posición de memoria de cuatro cifras y al campo de datos su contenido de dos cifras correspondiente. La dirección de memoria inicial es un valor propuesto, que después de conectar la alimentación tiene un valor aleatorio, y en el desarrollo siguiente el último valor relacionado con el comando 0. El cursor se encuentra a la izquierda del campo de direcciones.

Si el cursor se encuentra en el campo de direcciones, la dirección puede ser modificada pulsando las teclas de desplazamiento del cursor. Tras la modificación de la dirección, aparece inmediatamente el contenido correspondiente de la memoria en el campo de datos.

Para modificar el contenido de la memoria hay que desplazar el cursor al campo de datos. Esto puede conseguirse de dos maneras:

a) Después de escribir en la parte derecha de la dirección, el cursor se traslada automáticamente a la parte izquierda del campo de datos.

b) La misma posición puede ser alcanzada con las teclas de desplazamiento del cursor. Tras cualquier modificación de una cifra de la palabra de datos, el

nuevo valor es asimilado inmediatamente.

En este comando las direcciones y los datos no están concluidos con la tecla de cierre. Esta tiene otra función. Inmediatamente de la posición actual el cursor en el campo de direcciones y datos. La pulsación de la tecla ↓ conduce a que la dirección del campo de direcciones se incrementen en uno y que en el campo de datos aparezca su valor correspondiente. Apretando la tecla ↑ se decrementa en uno la dirección del campo de direcciones, e igualmente aparece el contenido de la memoria en el campo de datos. Después de cualquier pulsación de la tecla ↓ o ↑, el cursor se encuentra en la parte izquierda del campo de datos.

El comando 0 es en principio interminable y solo puede ser revocado con la introducción de un nuevo comando (para ello trasladar el cursor al campo de comandos), o pulsando la tecla R.

Particularidades:

1) Después de conectar el aparato se encuentran en la RAM valores aleatorios, que aparecerán al utilizar el comando.

2) El programa monitor lee al ejecutar el comando 0 , la información contenida en la memoria y la muestra en el campo de datos del display. Una cifra introducida por el teclado como información aparece solamente en el campo de datos que ha podido ser escrita en la memoria, es decir, si bajo la dirección se encuentra una zona de memoria utilizable. En todos los demás casos, por ejemplo para memorias no conectadas en la dirección, aparece en el campo de datos del display la información encontrada en las dos últimas cifras de la dirección.

Ejemplo:

Supongamos que queremos introducir en la dirección 1800H de la memoria de escritura / lectura la palabra de datos. A continuación queremos comprobar el contenido de la dirección siguiente: 1801H. El desarrollo esta representado en la página siguiente.

Ejemplo para el comando 0:

| TECLA | ENTRADA | | SALIDA | |
|-------|--|----------------------------------|-------------|--|
| | SIGNIFICADO | DISPLAY | SIGNIFICADO | |
| | | (Cursor en el campo de comando). | | Listo para la introduccion de un simbolo de comando. |
| 0 | Simbolo de comando. | O. ^ | | |
| ↓ | Cierre del simbolo de comando | O. XXXX = YY ^ | | Simbolo de comando; direccion de memoria y contenido. |
| 1 | Nueva cifra para el primer lugar de la direccion de memo. | O. 1XXX = YY ^ | | Simbolo de comando; direccion de memo.(modificada)y contenido de memo. |
| 8 | Nueva cifra para el segundo lugar de la direccion de memo. | O. 18XX = YY ^ | | Simbolo de comando; direccion de memo.(modificada)y contenido de memo. |
| 0 | Nueva cifra para el tercer lugar de la direccion de memo. | O. 180X = YY ^ | | Simbolo de comando; direccion de memo.(modificada)y contenido de memo. |

LLENAR UNA ZONA DE MEMORIA CON UNA
CONSTANTE. COMANDO 1.

Función.

El comando 1 sirve para inscribir la misma palabra de información (constante) en todas las direcciones de memorias. Las direcciones de principio y final, así como la palabra información, a escribir, hay que introducirlos como párrafos.

Ejecución.

Para el desarrollo normal del comando se presupone que en la zona a escribir se encuentra una memoria de escritura / lectura.

El cursor debe de encontrarse al principio en el campo de comandos. Se introduce el símbolo de comando 1 y se pulsa una de las teclas de cierre. A continuación aparece en el campo de direcciones el símbolo para la dirección de comienzo (AA), y en el campo de datos la cifra 00, como valor presupuesto de la dirección. El cursor se encuentra al principio en la parte izquierda del campo de datos.

El valor presupuesto de la dirección puede ser ahora modificado. Apretando una de las teclas de cierre el valor es tomado como válido. A continuación aparece

en el campo de direcciones el símbolo de la dirección final (EA), y en el campo de datos la cifra 00 como valor presupuesto para la dirección. La introducción de un valor y el cierre ocurre de la misma manera. Seguidamente aparece en el campo de direcciones el símbolo del Byte de datos (db), y en el campo de datos la cifra 00 como valor presupuesto para la palabra de información.

Después de la introducción de un valor y su cierre se da por concluida la introducción de parámetros.

Posteriormente el aparato ejecuta el comando. Para ello aparece continuamente en el campo de direcciones la dirección actual, y en campo de datos la palabra información. Después de la ejecución del comando el cursor se encuentra en el campo de comandos. El campo de direcciones y el de datos se encuentran apagados.

Particularidades.

1) Si la dirección de comienzo introducida es mayor que la dirección final se escribirá en las dos zonas parciales que comprenden dichas direcciones (dirección de comienzo . . . FFFFH) y (0000H . . . dirección final).

Ejemplo:

Vamos a llenar la zona de memoria comprendida entre la dirección de comienzo 1B10H y la dirección final 1BFFH con la constante 76. El desarrollo viene representado a continuación.

Ejemplo para el comando 1:

| TECLA | ENTRADA SIGNIFICADO | SALIDA DISPLAY (Cursor en el campo de comando). |
|-------|---|---|
| 1 | Simbolo del comando | 1. ^ |
| ↓ | Cierre del simbolo de comando | 1. A A = 0 0 0 0 ^ |
| 1 | Nueva cifra para el primer lugar de la direccion de comienzo | 1. A A = 1 0 0 0 ^ |
| B | Nueva cifra para el segundo lugar de la direccion de comienzo | 1. A A = 1 B 0 0 ^ |
| 1 | Nueva cifra para el tercer lugar de la direccion de comienzo | 1. A A = 1 B 1 0 ^ |
| ↓ | Cierre de la direccion de comienzo. | 1. E A = 0 0 0 0 ^ |
| 1 | Nueva cifra para el primer lugar de la direccion final. | 1. E A = 1 0 0 0 ^ |
| B | Nueva cifra para el segundo lugar de la direccion final. | 1. E A = 1 B 0 0 ^ |

Ejemplo para el comando 1:

| TECLA | ENTRADA | | SALIDA | |
|-------|---|-------------------|---|--|
| | SIGNIFICADO | DISPLAY | SIGNIFICADO | |
| 0 | Nueva cifra para el cuarto lugar de la direccion de memo. | 0. 1800 = YY ^ | Simbolo de comando, direccion de memo. (modificada) y contenido de memo. | |
| 3 | Nueva cifra para el primer lugar del contenido de memo. | 0. 1800 = 3Y ^ | Simbolo de comando, direccion de memo. y contenido de memo. (modificado). | |
| 1 | Nueva cifra para el segundo lugar del contenido de memo. | 0. 1800 = 31 ^ | Simbolo de comando, direccion de memo. y contenido de memo. (modificado). | |
| ↓ | Incremento de la direccion de memo. | 0. 1801 = YY ^ | Simbolo de comando, direccion de memo. (incrementada). Contenido de memo. | |

^ POSICION DEL CURSOR

XXXX VALOR PROPUESTO PARA LA DIRECCION DE MEMO.

YY ANTIGUO CONTENIDO DE LA POSICION DE MEMORIA SEÑALADA

| TECLA | ENTRADA SIGNIFICADO | SALIDA DISPLAY |
|-------|--|----------------------------|
| F | Nueva cifra para el cuarto lugar de la direccion final | 1. E A = 1 B F F ^ |
| ↓ | Cierre de la direccion final | 1. d b = 0 0 ^ |
| 7 | Nueva cifra para el primer lugar del Byte de datos | 1. d b = 7 0 ^ |
| 6 | Nueva cifra para el segundo lugar del Byte de datos | 1. d b = 7 6 ^ |
| ↓ | Cierre del comando. | 1. 1 * * * = 7 6 1. (+) |

(+) COMANDO EJECUTADO

(^) POSICION DEL CURSOR

(*) DISPLAY CORRIENDO EN ESA POSICION

TRASLADO DEL CONTENIDO DE MEMORIA

COMANDO 2

Función.

El comando n.2 sirve para trasladar información de una zona original de memoria a una zona final de memoria. La zona original de memoria viene definida por una dirección de comienzo y de final; la zona final de memoria viene definida por su dirección de comienzo; en este caso se llama dirección de destino. Estas tres cifras hay que darlas como parámetros.

Ejecución.

Para el desarrollo del comando descrito a continuación se presupone que la dirección de destino se encuentra en la zona de memoria RAM. Recuérdese que la dirección de comienzo no puede ser mayor que la dirección final.

El cursor debe encontrarse al principio en el campo de comandos. Se introduce el símbolo de comando n.2 y se pulsa una de las teclas de cierre. A continuación aparece en el campo de direcciones el símbolo para la dirección de la zona original (AA) y en el campo de datos la cifra 0000 como valor propuesto para la dirección el cursor se encuentra al principio

en la parte izquierda del campo de datos.

El valor propuesto como dirección puede ser ahora modificado apretando una de las teclas de cierre; su valor será tomado como válido. A continuación aparece en el campo de direcciones el símbolo para la dirección final de la zona original (EA) y en el campo de datos la cifra 0000, como valor propuesto como dirección. La entrada de un valor y el cierre ocurren de la misma forma. A continuación se introduce la dirección de destino a la que le corresponde el símbolo ba y un valor propuesto 0000. Con el correspondiente pulsado de una de las teclas de cierre termina la entrada de parámetros.

Si la dirección de destino es más pequeña que la dirección de comienzo de la zona original, se traslada primeramente el contenido de la dirección de comienzo a la dirección de destino, y a continuación las siguientes direcciones en orden creciente. Si por el contrario la dirección de destino es mayor que la dirección de comienzo, se traslada el contenido de la dirección original a la última dirección de la zona de destino y las direcciones siguientes en orden creciente.

Mientras corre el comando aparece continuamente en el display la dirección de destino actualizada a cada instante, y en el campo de datos la palabra de información. Después de la ejecución del comando el cursor se encuentra en el campo de comandos y los campos de direcciones y de datos están apagados.

Particularidades.-

1) Si la dirección de comienzo introducida es mayor que la dirección final, o si se sobrepasa la dirección FFFFH no se asegura una correcta ejecución del comando.

Ejemplo.-

Se desea trasladar el contenido de la memoria entre la dirección de comienzo 0000H y la dirección final 00FFH en la zona de memoria a partir de la dirección 1800H. El desarrollo viene representado a continuación:

Ejemplo para el comando 2:

| TECLA | SIGNIFICADO | ENTRADA | SALIDA |
|-------|--|---------|----------------------------------|
| | | | DISPLAY |
| | | | (Cursor en el campo de comando). |
| 2 | Simbolo del comando | 2. | 2. ^ |
| ↓ | Cierre del simbolo de comando | 2. | 2. A A = 0 0 0 0 ^ |
| ↓ | Cierre de la direccion de comienzo de la zona original | 2. | 2. E A = 0 0 0 0 ^ |
| → | Aceptacion del primer lugar de la direccion final de la zona original | 2. | 2. E A = 0 0 0 0 ^ |
| → | Aceptacion del segundo lugar de la direccion final de la zona original | 2. | 2. E A = 0 0 0 0 ^ |
| F | Nueva cifra para el tercer lugar de la direccion final de la zona original | 2. | 2. E A = 0 0 F 0 ^ |
| F | Nueva cifra para el cuarto lugar de la direccion final de la zona original | 2. | 2. E A = 0 0 F F ^ |

Ejemplo para el comando 2:

| TECLA | ENTRADA SIGNIFICADO | SALIDA DISPLAY |
|-------|---|--|
| ↓ | Cierre de la dirección final de la zona original | 2. b A = 0 0 0 0 ^ |
| 1 | Nueva cifra para el primer lugar de la dirección destino | 2. b A = 1 0 0 0 ^ |
| 8 | Nueva cifra para el segundo lugar de la dirección destino | 2. b A = 1 8 0 0 ^ |
| ↓ | Cierre del comando. | 2. 1 8 * * = * * ^ (+) |

(+) COMANDO EJECUTADO

(^) POSICION DEL CURSOR

(*) DISPLAY CORRIENDO EN ESA POSICION

CAMBIO DE DIRECCIONES. COMANDO 3

Función:

El comando 3 sirve para cambiar las posiciones de salto de las instrucciones referentes a saltos en las CPU 8080-8085. Cuando las instrucciones de salto se encuentran en una zona determinada y las direcciones de salto en otra zona también conocida.

En general, éste comando se usa conjuntamente con el comando 2 con el que se realiza el traslado del contenido de una zona de memoria a otra. El comando 3 no realiza este traslado, pero hay que suministrarle los parámetros usados con el comando 2 (dirección de comienzo, dirección final y dirección de destino).

La zona de memoria en la que hay que cambiar las direcciones de salto vienen determinadas por la dirección de comienzo y la dirección final de la zona, que esta señalado a la zona original de tal forma que después señalen un bloque de información de la zona de destino.

Ejecución:

Para el desarrollo normal del comando se presupone que en la zona de destino se encuentra una zona de RAM y que la dirección de comienzo señala al

código de operación de una instrucción. Además la dirección de comienzo no debe ser mayor que la dirección final.

El cursor debe encontrarse al principio del campo de comando. Se introduce el símbolo de comando 3 y se pulsa una de las teclas de cierre. A continuación aparece el campo de direcciones con el símbolo para la dirección de comienzo de la zona original AA, y en el campo de datos el último valor de AA relacionado con el comando 2 ó 3. El cursor se encuentra al principio de la parte izquierda del campo de datos.

El valor propuesto de la dirección puede ser ahora modificada. Pulsando una de las teclas de cierre, el valor es tomado como válido. De la misma forma se introduce a continuación la dirección final de la zona original (EA) y la dirección de destino (bA).

A continuación aparece en el campo de direcciones el símbolo para la dirección de comienzo de la zona a modificar (AU) y en el campo de datos la cifra 0000, como valor propuesto. Después se introduce la dirección final de la zona a modificar (EU) con el valor propuesto 0000. A continuación ejecutaremos el comando encontrando al cursor en el campo de comandos.

CAMBIO DE DIRECCIONES

mientras que el campo de datos y direcciones se encuentra apagado.

Ejemplo para el comando 3:

| TECLA | ENTRADA SIGNIFICADO | SALIDA DISPLAY (Cursor en el campo de comando). |
|-------|---|---|
| 3 | Simbolo del comando | 3. ^ |
| ↓ | Cierre del simbolo de comando | 3. A A = 0 7 E 3 ^ |
| ↓ | Cierre de la direccion de comienzo de la zona original (Valor propuesto aceptado). | 3. E A = 0 7 F F ^ |
| ↓ | Cierre de la direccion de destino (Valor propuesto aceptado). | 3. A U = 0 0 0 0 ^ |
| 1 | Nueva cifra para el primer lugar de la direccion de comienzo de la zona de memoria a modificar | 3. A U = 1 0 0 0 ^ |
| 8 | Nueva cifra para el segundo lugar de la direccion de comienzo de la zona de memoria a modificar | 3. A U = 1 8 0 0 ^ |
| ↓ | Cierre de la direccion de comienzo (Valor propuesto aceptado). | 3. E U = 0 0 0 0 ^ |
| 1 | Nueva cifra para el primer lugar de la direccion final de la zona de memoria a modificar | 3. E U = 1 0 0 0 ^ |

Ejemplo para el comando 3:

| TECLA | ENTRADA SIGNIFICADO | SALIDA DISPLAY |
|-------|--|-----------------------|
| 8 | Nueva cifra para el segundo lugar de la direccion final. | 3. E U = 1 8 0 0 ^ |
| 1 | Nueva cifra para el tercer lugar de la direccion final. | 3. E U = 1 8 1 0 ^ |
| C | Nueva cifra para el cuarto lugar de la direccion final. | 3. E U = 1 8 1 C ^ |
| ↓ | Cierre el comando. | 3 ^ (+) |

(+) COMANDO EJECUTADO

(^) POSICION DEL CURSOR

(*) DISPLAY CORRIENDO EN ESA POSICION

Escritura de EPROMS (comandos 0 a 2)

Con este sistema autónomo podemos escribir las memorias 2716. Para meter o sacar EPROMS del lugar de conexión n.2 hay que poner la palanca de presión en posición vertical. Después de introducirla hay que establecer contacto bajando la palanca a la posición horizontal.

La EPROM puede escribirse al igual que una RAM, con la única diferencia que un bit pasado de 1 a 0 no puede ser modificado. Ante un error de programación no queda más remedio que borrar toda la memoria. Por ello se recomienda no escribir la EPROM mediante el comando 0. Es más ventajoso escribir primero en la RAM con ayuda del comando 0, y comprobar el contenido. Si todo está en orden todo el bloque de información puede ser trasladado a la EPROM mediante el comando 2. En caso de ser necesario puede llenarse una zona de la EPROM con una constante, mediante el comando 1. El uso del comando 0 para la EPROM solo tiene sentido cuando se desee escribir células aisladas de memoria.

C A P I T U L O V I I
A L I M E N T A C I O N D E L S I S T E M A

Buscando en lo posible la independencia de otros elementos, como es el caso de las fuentes de alimentaciones exteriores, dotando un sistema de alimentación permitiendo trabajar con las diferentes exigencias de la programación de la familia 27XX.

En el caso de la 2716 las tensiones necesarias para la programación son 25 V. para Vpp y 5 V. para la entrada Vcc. Sin embargo, para la 2764 se requiere 21 V. en Vpp y 6 V. en Vcc. Por ello con las diferentes tensiones que requiere el sistema considero necesaria una descripción de fuente de alimentación.

El diagrama 7.1. muestra el esquema electrónico, en el cual podemos apreciar los diferentes elementos que lo componen, estos son:

- Un transformador de tensión.
- Tres fuentes rectificadores.
- Un regulador de tensión positiva 7815.
- Un regulador de tensión positiva LM323.
- Un regulador de tensión negativa 7906.
- Condensadores y resistencias varias.

El transformador de tensión permite tensiones de entrada (240 V-220 V-125 V-110 V) y para tensiones de salidas de (24 V-14 V-10 V). En la entrada del transformador se ha tomado correspondiente a 220 V si bien, esta puede cambiarse según se requiera, cambiando las actuales conexiones.

El transformador puede proporcionar una corriente de salida de tres amperios, cantidad esta superior a la consumida por el circuito pero previendo las posibles ampliaciones del sistema se ha creído conveniente que dicho transformador proporcione dicha corriente.

Recordemos, que el sistema MSX tiene previsto posibles ampliaciones por medio de los puertos A y B del 8355/2. El sistema autónomo puede también ampliarse por medio de los seis puertos que proporcionan los 8155.

Las salidas del transformador se dividen en cuatro rutas estas son:

Una toma de 34 V de tensión eficaz, cuya finalidad es suministrar a los reguladores de tensión uA723 la tensión para estabilizar las tensiones de polarización de los amplificadores operacionales uA323.

que suministran la tensión adecuada a las exigencias de programación.

Las tensiones de salida de estos estabilizadores $\mu A723$ son de 26 y 21 V como indicamos en el diagrama 7.1.

La segunda salida del transformador proporciona mediante su respectivo rectificador 20 V de tensión eficaz y conectado a través de un condensador de filtrado de 1000 $\mu F/40$ V al regulador de tensión positiva 7815 destinado a la polarización de los LM724.

Por último la tercera salida proporciona 14 V de tensión eficaz que por medio de un condensador de filtrado de 1000 μF a dos reguladores de tensión el primero positivo (LM323) destinado a suministrar la tensión de polarización para la mayoría de CI que configuran el programador. Dicho LM323 es capaz de proporcionar 3 A.

Como puede observarse el 7906 emplea parte de la tensión de esta toma para su funcionamiento. Uniendo las patillas GND de ambos integrados a la masa del sistema podemos obtener por medio de un diodo zener de 8.2 V la conveniente tensión negativa para el 7906 al cual se le ha previsto a su entrada del

correspondiente condensador (22 uF) para evitar el rizado. Esta tensión negativa está destinada a la polarización de los integrados LM724.

Todos los integrados que componen el programador están salvaguardados de posibles cortocircuitos mediante un fusible de 2 A. previsto en la entrada del transformador permitiendo una mayor tranquilidad al usuario.

A P E N D I C E A

CARACTERISTICAS DE LAS EPROMS 27XXX

2716

FROM BORRABLE POR RAYOS ULTRAVIOLETAS PROGRAMABLE
ELECTRICAMENTE CON CAPACIDAD DE 16K (2K*8)

La 2716 de Intel es una memoria de solo lectura, borrrable por rayos ultravioletas y programable eléctricamente (EPROM). La tensión de polarización es de 5V., tiene un modo de permanecer llamado standby del cual ya hablaremos, tiene un rápido y fácil modo de programación. La realización de sistemas con esta EPROM disminuye el coste de producción sin perder eficacia por ello.

La 2716 con 5V. y con un tiempo de acceso de 350 nsg., es ideal para el uso de sistemas basados en microprocesadores y más concretamente con los 8085 y 8086.

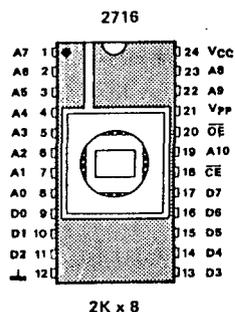
Las elección de las posibles versiones 2716-55 y 2716-65 es una buena elección para aplicaciones que exijan una rápida respuesta. El modo standby de la 2716 reduce considerablemente la potencia disipada sin incrementar por ello el tiempo de acceso. Cuando esta al máximo poder de disipación la potencia consumida es de 520 mW. mientras que la máxima potencia en modo standby es de tan solo de 132 mW., un ahorro del 75 %.

CARACTERÍSTICAS 2716

La 2716 presenta un simple método de programación, un simple pulso a nivel TTL. El tiempo total de programación de los 16.384 bits es tan solo de 100 μ s.

diagrama A.1.

patillaje de la 2716



MODOS DE OPERACION

Los modos de programación de la 2716 son seis, listados en el diagrama A.2. En cualquier modo en el que se trabaje, los niveles requeridos en las entradas son TTL. Las tensiones requeridas para el normal funcionamiento son $V_{CC}=5V$. y $V_{PP}=25V$., durante los tres modos que requiere la programación y cinco voltios para los modos normales de trabajo.

| Mode \ Pins | CE (18) | OE (20) | V _{PP} (21) | V _{CC} (24) | Outputs (9-11, 13-17) |
|-----------------|---|-----------------|----------------------|----------------------|-----------------------|
| Read | V _{IL} | V _{IL} | +5 | +5 | D _{OUT} |
| Output Disable | V _{IL} | V _{IH} | +5 | +5 | High Z |
| Standby | V _{IH} | X | +5 | +5 | High Z |
| Program | Pulsed V _{IL} to V _{IH} | V _{IH} | +25 | +5 | D _{IN} |
| Verify | V _{IL} | V _{IL} | +25 | +5 | D _{OUT} |
| Program Inhibit | V _{IL} | V _{IH} | +25 | +5 | High Z |

NOTES: 1. X can be V_{IL} or V_{IH}

CARACTERISTICAS 2716

D.C. AND A.C. OPERATING CONDITIONS DURING READ

| | 2716 | 2716-1 | 2716-2 | 2716-5 | 2716-8 |
|-----------------------------------|----------|----------|----------|----------|----------|
| Temperature Range | 0°C-70°C | 0°C-70°C | 0°C-70°C | 0°C-70°C | 0°C-70°C |
| VCC Power Supply ^(1,2) | 5V ±5% | 5V ±10% | 5V ±5% | 5V ±5% | 5V ±5% |
| Vpp Power Supply ⁽²⁾ | Vcc | Vcc | Vcc | Vcc | Vcc |

Modo de lectura

La 2716 posee dos funciones de control; ambas funciones han de ser realizadas en el orden lógico para obtener en las correspondientes salidas los datos. La función Chip Enable (\overline{CE}) es usado para la selección del dispositivo. La salida Output Enable (\overline{OE}) es una función de control de las salidas; asume el control para la presentación de datos en los correspondientes pines de salida; independientemente de la selección de salida.

Asumiendo que las direcciones son estables, el tiempo de acceso (T_{acc}) es igual a la espera de la activación del \overline{CE} en la salida (T_{ce}). Los datos son presentados a la salida un tiempo T_{oe} ; después de caer la señal \overline{OE} . Asumiendo que \overline{CE} se encuentra en posición baja y las direcciones permanezcan estables por un tiempo $T_{acc} - T_{oe}$.

Modo standby

La 2716 posee un modo de espera; el cual, reduce en un 75%; desde 525 mW. a 132 mW. el consumo. La 2716 permanece en este estado mientras no es seleccionada; es decir, mientras el valor lógico a la entrada de \overline{CE} sea high. En este modo, la salida se

encuentra en estado de alta impedancia, independientemente de la entrada \overline{OE} .

Output OR-tieing

Como las 27165 son usualmente usadas en diferentes configuraciones, Intel las ha provisto de dos lineas de control que facilita el uso de las mismas, cuando se conectan multiples memorias. Las dos lineas de control son usadas para:

- a) Una reducci3n en la potencia disipada.
- b) El uso de un mismo bus de comunicaci3n para distintos dispositivos.

Modo de programaci3n

Las consideraciones previas para la programaci3n de la 2716 fueron estudiadas en el capitulo II, por lo cual, lo omito aqu3.

La 2716 se encuentra en modo de programaci3n cuando la entrada V_{pp} tiene 25V. y \overline{OE} se encuentra en V_{IH} . Los datos ser3n aplicados en serie de 8 bits, en paralelos, a los pines de salida de datos. Los niveles requeridos tanto para las entradas de direcciones como de datos son TTL.

Cuando las direcciones se encuentran en modo estable, un pulso alto de 50 msg. es aplicado a la

entrada del pin \overline{CE} . Un pulso habrá de ser aplicado a cada posición de memoria para ser programado. El valor máximo del pulso es de 55 msg. La 2716 no debe ser programada con un nivel de señal continua aplicada a la entrada.

La programación de múltiples 2716, en paralelo para un mismo dato, puede ser fácilmente realizado si se dispone de los zócalos conectados en paralelos para la inserción de las 2716 y su posterior grabación.

Inhabilitación de programación

La programación de múltiples 2716 en paralelo, con diferentes datos es también fácilmente realizable. Exepto la entrada \overline{CE} , todas las demás entradas incluidas \overline{OE} pueden ser conectadas en paralelo. Un pulso de programación a nivel TTL aplicado al \overline{CE} de la 2716 y con V_{pp} a 25 V. programa el dato presentado, mientras que un nivel bajo en la entrada \overline{CE} inhibe las otras 2716 que están siendo programadas.

Modo de verificación

Para estar seguro de la correcta programación de las determinadas palabras, el dispositivo puede ser

verificado mientras que permanece los 25 V. en la entrada Vpp. Exepto durante la programación y el ciclo de verificación, la tensión en Vpp ha de ser de 5 V.

En los diagramas A.3 y A.4 se presentan los diferentes tiempos que requiere este dispositivo para el normal fucionamiento, así como, para la correcta programación.

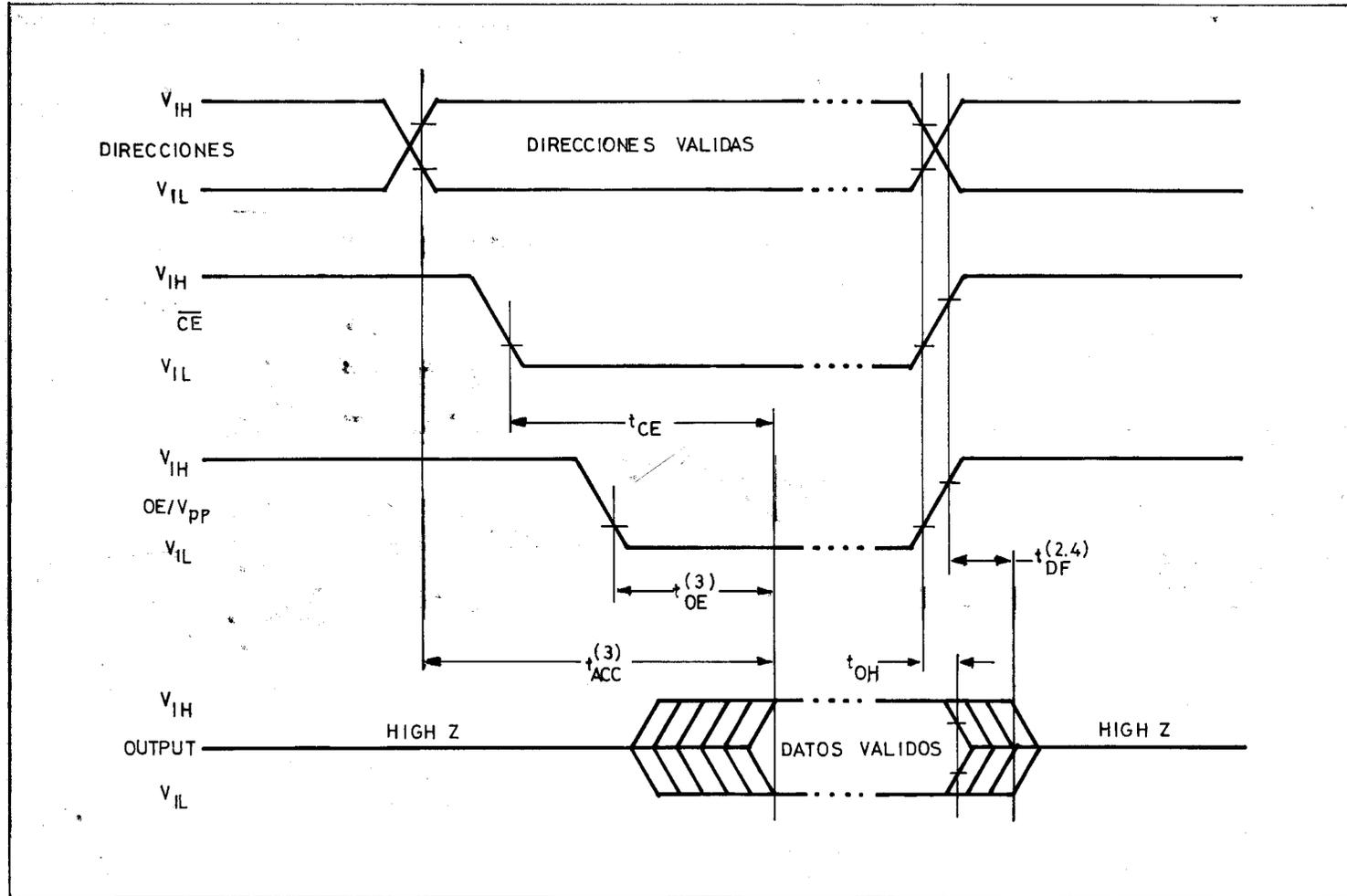
En los diagramas A.5 y A.6 se presentan los cronogramas de esta EPROM. Observese que ha diferencia de las EPROMS que a continuación se estudian, ésta, en particular para se grabada es necesario un pulso a nivel alto en la entrada \overline{CE} , mientras que las demás es a nivel bajo.

Las características para el correcto borrado fueron ya descritas en el capítulo II, para su lectura remitirse a dicho capítulo.

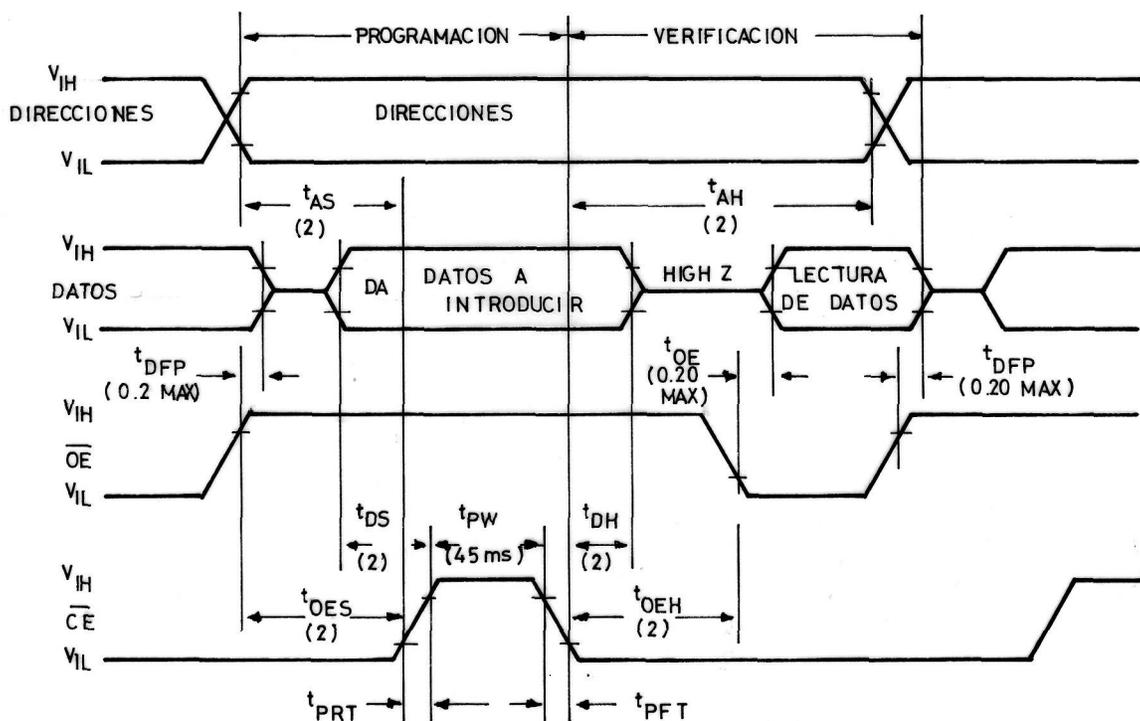
A.C. CHARACTERISTICS

| Symbol | Parameter | Limits (ns) | | | | | | | | | | Test Conditions† |
|----------------------------------|---|-------------|------|--------|------|--------|------|--------|------|--------|------|--|
| | | 2716 | | 2716-1 | | 2716-2 | | 2716-5 | | 2716-6 | | |
| | | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. | |
| t _{ACC} | Address to Output Delay | | 450 | | 350 | | 390 | | 450 | | 450 | $\overline{CE} = \overline{OE} = V_{IL}$ |
| t _{CE} | \overline{CE} to Output Delay | | 450 | | 350 | | 390 | | 490 | | 650 | $\overline{OE} = V_{IL}$ |
| t _{OE} ^[4] | Output Enable to Output Delay | | 120 | | 120 | | 120 | | 160 | | 200 | $\overline{CE} = V_{IL}$ |
| t _{DF} ^[4,6] | \overline{CE} or \overline{OE} High to Output Float | 0 | 100 | 0 | 100 | 0 | 100 | 0 | 100 | 0 | 100 | $\overline{CE} = V_{IL}$ |
| t _{OH} | Output Hold from Addresses, \overline{CE} or \overline{OE} Whichever Occurred First | 0 | | 0 | | 0 | | 0 | | 0 | | $\overline{CE} = \overline{OE} = V_{IL}$ |

MAPA LECTURA



CRONOGRAMA PROGRAMACION.



1. TODOS LOS TIEMPOS ENTRE () SON MINIMOS Y EN μ S SI NO SE ESPECIFICA LO CONTRARIO
2. LOS NIVELES DE LAS ENTRADAS SON 0.8 V. PARA V_{IL} Y 2 V. PARA V_{IH}

CARACTERISTICAS 2732A

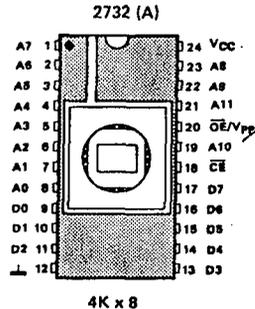
2732A

FROM BORRABLE POR RAYOS ULTRAVIOLETA, PROGRAMABLE ELECTRICAMENTE CON CAPACIDAD DE 32K (4K*8).

La Intel 2732A requiere para su funcionamiento una sola tensión de polarización, 5V. Memoria de solo lectura, borrable y programable eléctricamente (EPROM).

El tiempo medio de acceso de la 2732A es de 250 ns. Si se toma la versión 2732A-2 de rápida selección, el tiempo es de 200 ns. Estos tiempos son compatibles con los exigidos en los sistemas de microprocesadores que trabajan a 8 Mhz. sin que tengan que emplear el estado WAIT.

Patillaje de la
2732 A



Una importante característica de la 2732A es la separación del control de salidas, Output Enable

CARACTERISTICAS 2732A

(\overline{OE}) del \overline{CE} , lo cual, permite que las salidas puedan ser desconectadas mediante el \overline{OE} del bus de conecciones del sistema.

La 2732A tiene un modo standby que reduce la potencia disipada sin incrementar por ello el tiempo de acceso. La máxima corriente consumida en modo activo es de 125 mA., mientras que en modo standby la mayor corriente consumida es de 35 mA., con lo cual, se reduce en un 70% la disipación. Este modo de trabajo: standby es seleccionado cuando se presentan en su entrada \overline{CE} un nivel lógico de TTL alto.

La fabricación de la 2732A está realizada con tecnología HMOS. lo cual presta a esta memoria, de Intel, un alto grado de rapidez de respuesta a los dispositivos.

Las características de borrado fueron expuestas en el capítulo II. Remítanse a él en caso de consulta.

MODOS DE OPERACION

Los seis modos de operación de la 2732A son listados en el diagrama A.10. En el modo de lectura son requeridos cinco voltios solamente. Todas las entradas

CARACTERISTICAS 2732A

son a niveles TTL, excepto para \overline{OE} / V_{pp} durante la programación y de 12.5V. en pin de direccionamiento A9 para el identificador inteligente. En el modo de programación las entradas son también a niveles TTL, excepto en \overline{OE} / V_{pp} al cual se le aplica 21V. a la entrada

| MODE | PINS | \overline{CE} (18) | \overline{OE}/V_{pp} (20) | A_9 (22) | V_{CC} (24) | OUTPUTS (9-11,13-17) |
|------------------------|------|-------------------------|--------------------------------|---------------|------------------|-------------------------|
| Read | | V_{IL} | V_{IL} | X | +5 | O_{OUT} |
| Output Disable | | V_{IL} | V_{IH} | X | +5 | High Z |
| Standby | | V_{IH} | X | X | +5 | High Z |
| Program | | V_{IL} | V_{pp} | X | +5 | D_{IN} |
| Program Inhibit | | V_{IH} | V_{pp} | X | +5 | High Z |
| Intelligent Identifier | | V_{IL} | V_{IL} | V_H | +5 | Code |

Modo de lectura

La 2732A posee dos funciones de control, ambas funciones han de ser realizadas en el orden lógico para obtener en las correspondientes salidas los datos. La función Chip Enable (\overline{CE}) es usado para la selección del dispositivo. La salida Output Enable (\overline{OE}) es una función de control de las salidas, asume el control para la presentación de datos en los correspondientes pines de salida, independientemente de la selección de salida.

Asumiendo que las direcciones son estables, el tiempo de acceso (T_{acc}) es igual a la espera de la activación del \overline{CE} en la salida (T_{ce}). Los datos son presentados a la salida un tiempo T_{oe} , después de caer la señal \overline{OE} . Asumiendo que \overline{CE} se encuentra en posición baja y las direcciones permanezcan estables por un

tiempo $T_{acc} - T_{oe}$.

Modo standby

La 2732A posee un modo de espera, el cual, reduce en un 75%, desde 525 mW. a 132 mW. el consumo. La 2732A permanece en este estado mientras no es seleccionada, es decir, mientras el valor lógico a la entrada de CE sea high. En este modo, la salida se encuentra en estado de alta impedancia, independientemente de la entrada \overline{OE} .

Output OR-tieing

Como las 2732A son usualmente usadas en diferentes configuraciones, Intel las ha provisto de dos líneas de control que facilita el uso de las correcciones en múltiples memorias. Las dos líneas de control son usadas para:

- a) Una reducción en la potencia disipada.
- b) El uso de un mismo bus de comunicación para distintos dispositivos.

Modo de programación

Las consideraciones previas para la programación de la 2732A fueron estudiadas en el capítulo II, por lo cual, lo omito aquí.

La 2732A se encuentra en modo de programación cuando la entrada V_{pp} tiene 21V. Los datos serán aplicados en serie de 8 bits en paralelos a los pines de salida de datos. Los niveles requeridos tanto para las entradas de direcciones como de datos son TTL.

Cuando las direcciones se encuentran en modo estable, un pulso bajo de 50 msg. es aplicado a la entrada del pin \overline{CE} . Un pulso habrá de ser aplicado a cada posición de memoria para ser programado. El valor máximo del pulso es de 55 msg. La 2732A no debe ser programada con un nivel de señal continua aplicada a la entrada.

La programación de múltiples 2732A en paralelo, para un mismo dato, puede ser fácilmente realizado si se dispone de los zócalos conectados en paralelos, para la inserción de las 2732A y su posterior grabación.

Inhabilitación de programación

La programación de múltiples 2732A en paralelo, con diferentes datos es también fácilmente realizable. Excepto la entrada \overline{CE} , todas las demás entradas incluidas \overline{OE} pueden ser conectadas en paralelo. Un pulso de programación a nivel TTL aplicado al \overline{CE} de la 2732A y con V_{pp} a 21 V. programa el dato presentado, mientras que un nivel bajo en la entrada CE inhibe las otras 2732A que están siendo programadas.

Modo de verificación

Para estar seguro de la correcta programación de las determinadas palabras, el dispositivo puede ser verificado, si la tensión de 21 V. disminuye a cero voltios en la entrada V_{pp} . Excepto durante la programación y el ciclo de verificación, la tensión en V_{pp} ha de ser de 5 V.

En los diagramas A.3 y A.4 se presentan los diferentes tiempos que requiere este dispositivo para el normal funcionamiento, así como para la correcta programación.

En los diagramas A.5 y A.6 se presentan los cronogramas de esta EPROM. Observese que ha diferencia de las EPROMS que a continuación se estudian: ésta, en particular para ser grabada es necesario un pulso alto en la entrada \overline{CE} , mientras que las demás es a nivel bajo.

Las características para el correcto borrado fueron ya descritas en el capítulo II.

FORMA INTELIGENTE DE IDENTIFICACION

La forma inteligente de identificación es un método que permite su identificación y por que operario fue manufacturado. Este método tiene la finalidad de permitir la programación por un equipo automático mediante un adecuado algoritmo. Este modo de identificación puede ser usado en un rango de temperatura entre 20 y 30 grados centígrados.

El método consiste en la identificación de un código binario procedente de los pines de salida del dispositivo.

Para activar este modo el equipo programador debe forzar el pin de direccionamiento A9 a 11.5 a

CARACTERISTICAS 2732A

12.5. Dos bytes pueden ser obtenidos secuencialmente aplicando un nivel alto y otro bajo en el pin de direccionamiento A0 (pin 8). Todas las demás líneas de dirección debe permanecer en estado lógico bajo, mientras dure la indentificación.

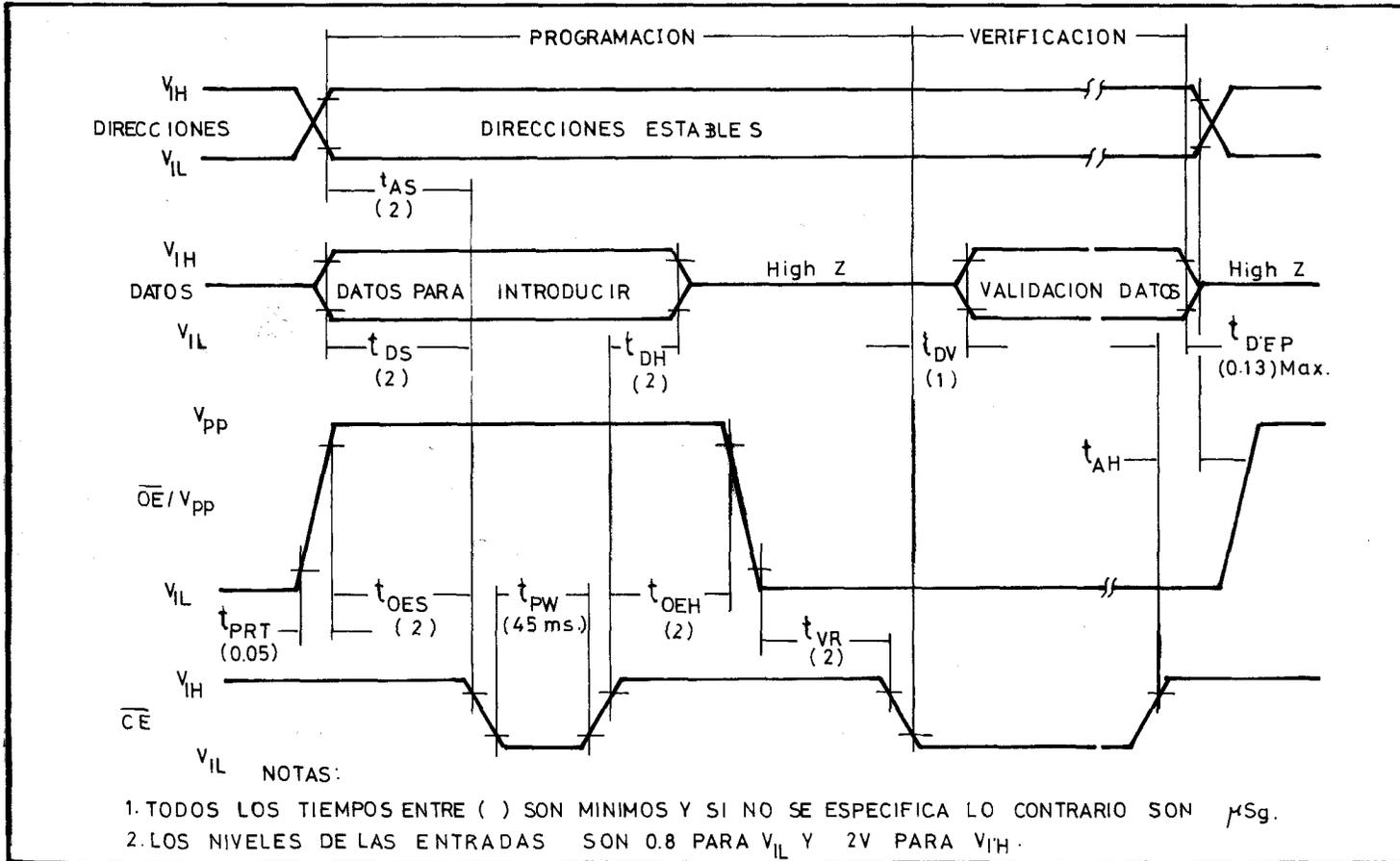
El byte 0 (A0 = V) representa el código de manufacturación. El byte 1 (A0 = V) identifica al dispositivo. Para la INTER 2732A, estos dos bytes identificadores vienen representados en la tabla A.11.

$T_A = 25 \pm 5^\circ\text{C}$, $V_{CC} = 5V \pm 5\%$, $V_{PP} = 21V \pm 0.5V$

| Symbol | Parameter | Limits | | | Units | Test Conditions† |
|------------------|--|--------|------|------|-------|---|
| | | Min. | Typ. | Max. | | |
| t _{AS} | Address Setup Time | 2 | | | μs | |
| t _{OES} | \overline{OE} Setup Time | 2 | | | μs | |
| t _{DS} | Data Setup Time | 2 | | | μs | |
| t _{AH} | Address Hold Time | 0 | | | μs | |
| t _{OEH} | \overline{OE} Hold Time | 2 | | | μs | |
| t _{DH} | Data Hold Time | 2 | | | μs | |
| t _{DFP} | \overline{OE} High to Output Not Driven | 0 | | 130 | ns | |
| t _{DV} | Data Valid from \overline{OE} | | | 1 | μs | $\overline{CE} = V_{IL}$, $\overline{OE} = V_{IL}$ |
| t _{PW} | \overline{CE} Pulse Width During Programming | 20 | 50 | 55 | ms | |
| t _{PRT} | \overline{OE} Pulse Rise Time During Programming | 50 | | | ns | |
| t _{VR} | V _{PP} Recovery Time | 2 | | | μs | |

| Identifier | Pin | A ₀ (8) | O ₇ (17) | O ₆ (16) | O ₅ (15) | O ₄ (14) | O ₃ (13) | O ₂ (11) | O ₁ (10) | O ₀ (9) | Hex Data |
|-------------------|-----------------|-----------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|-----------------------|-------------|
| Manufacturer Code | V _{IL} | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 89 |
| Device Code | V _{IH} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 01 |

CRONOGRAMA DE PROGRAMACION



MAPA LECTURA

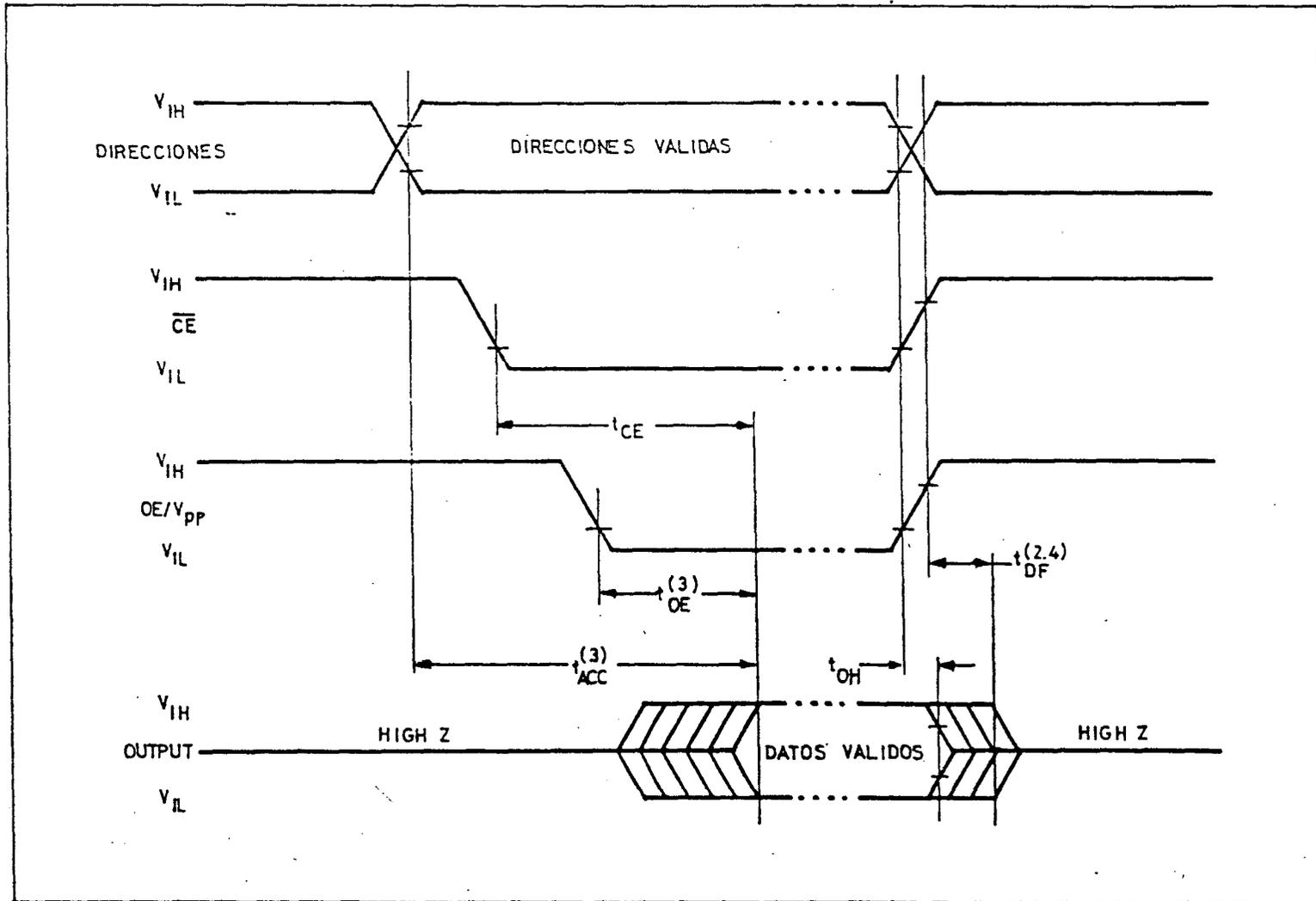


Diagrama n.º

2 7 6 4

PROM BORRABLE POR RAYOS ULTRAVIOLETA Y PROGRAMABLE
ELECTRICAMENTE CON CAPACIDAD DE 64K (8K*8).

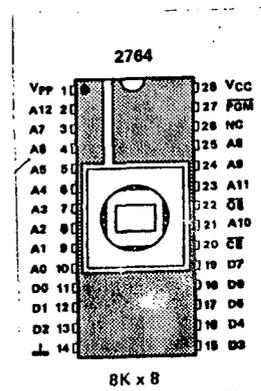
La Intel 2764 con solo una tensión de polarización ($V_{cc} = 5V.$), es una memoria de solo lectura programable eléctricamente y borrable por rayos ultravioleta con capacidad de 65.536 bits. La 2764 estándar tiene un tiempo de acceso de 250 ns. y una rápida selección de 200 ns. El tiempo de acceso es compatible con las configuraciones de sistemas microprocesadores que trabajan a frecuencias de 8 Mhz. La 2764 puede operar sin la necesidad de estados WAIT. La 2764 también es compatible con la familia 8051 que trabaja con 12Mhz.

Una importante característica de la 2764 es la separación de la entrada de control Out Enable (\overline{OE}) del Chip Enable (\overline{CE}). La \overline{OE} permite la desconexión de las salidas del bus del sistema.

La 2764 tiene un modo Standby que reduce la potencia consumida sin que por ello se incremente el tiempo de acceso. La máxima corriente que consume en modo activo es de 100 mA., mientras que en modo Standby

la corriente consumida es de solo 40 mA. Este modo es seleccionado por una señal alta TTL en la entrada \overline{CE} .

patillaje de la 2764



MODOS DE OPERACION

Los ocho modos de operación de la 2764 están contenidos en el diagrama A.25. Una sola tensión es necesaria para el modo de lectura. Todas las entradas son a niveles TTL, excepto para Vpp y para el identificador inteligente, en el cual deben aplicarse 12V. al pin A9.

Modo de lectura

La 2764 tiene dos funciones de control, ambas deben seguir una secuencia lógica para obtener los datos en la salida. El \overline{CE} es el que permite la selección del dispositivo para ser usado. La función de control \overline{OE} puede ser usado para habilitar los datos en sus salidas correspondientes, independientemente de la selección del dispositivo. Asumiendo que las

| MODE \ PINS | CE (20) | OE (22) | FGM (27) | A9 (24) | Vpp (1) | VCC (28) | Outputs (11-13, 15-19) |
|-------------------------|-----------------|-----------------|-----------------|----------------|-----------------|-----------------|------------------------|
| Read | V _{IL} | V _{IL} | V _{IH} | X | V _{CC} | V _{CC} | DOUT |
| Output Disable | V _{IL} | V _{IH} | V _{IH} | X | V _{CC} | V _{CC} | High Z |
| Standby | V _{IH} | X | X | X | V _{CC} | V _{CC} | High Z |
| Program | V _{IL} | V _{IH} | V _{IL} | X | V _{pp} | V _{CC} | DIN |
| Verify | V _{IL} | V _{IL} | V _{IH} | X | V _{pp} | V _{CC} | DOUT |
| Program Inhibit | V _{IH} | X | X | X | V _{pp} | V _{CC} | High Z |
| Intelligent Identifier | V _{IL} | V _{IL} | V _{IH} | V _H | V _{CC} | V _{CC} | Code |
| Intelligent Programming | V _{IL} | V _{IH} | V _{IL} | X | V _{pp} | V _{CC} | DIN |

direcciones son estables, el tiempo de direccionamiento (t_{ACC}) es igual al tiempo de espera para seleccionar el \overline{CE} para la salida (t_{CE}). Los datos son válidos a la salida después de un tiempo t_{OE} desde la presentación de la señal \overline{OE} , asumiendo que \overline{CE} este a nivel bajo y las direcciones sean estables por un tiempo $t_{ACC} - t_{OE}$.

Modo Standby

La 2764 tiene un modo de funcionamiento que reduce el consumo de corriente desde 100 mA. a 40 mA. La 2764 se encuentra en este modo cuando hay una señal alta TTL en la entrada \overline{CE} . Estando en este modo, las salidas se encuentran en estado de alta impedancia, independientemente de la entrada \overline{OE} .

Output OR-Tieing

La razón de que estas EPROMS sean usadas en configuraciones que requieren varias unidades 2764 es que poseen dos líneas de control que permiten la conexión de múltiples memorias. Las dos líneas de control permiten:

- a) Disminuir en lo posible la disipación de la memoria.

b) Asegurar la independencia de las salidas del bus del sistema.

Estas dos funciones de control permiten una mayor eficacia a la hora de seleccionar los diferentes dispositivos. La señal \overline{OE} puede ser conectada a las señales READ que configuran el bus del sistema.

Modo de programación

La 2764 se encuentra en modo de programación cuando V_{pp} está a 21V., \overline{CE} y \overline{PGM} a nivel bajo TTL. Los datos serán programados cuando se apliquen ocho bits en paralelo a los pines de salida de datos. Los niveles requeridos para las señales de direcciones y de datos son TTL.

Programación Stándar

Para la programación, \overline{CE} debe estar a nivel bajo durante todo el proceso, mientras que V_{pp} debe estar a 21V. Cuando las direcciones y los datos son estables, un pulso a nivel bajo de 50 ms. aplicado a la entrada \overline{PGM} . Un pulso debe ser aplicado para cada posición de memoria a programar. El periodo de duración del pulso no debe durar más de 55 ms.

La programación de múltiples 2764 en paralelo

con un mismo dato puede ser fácilmente realizado. Como las entradas están conectadas en paralelo para la programación con un mismo dato, un pulso aplicado a $\overline{\text{PGM}}$ realiza la grabación del dato.

Verificación.

Para la verificación de que la información programada es la correcta, el modo de operación es el siguiente: la señal $\overline{\text{CE}}$ y la $\overline{\text{OE}}$ al nivel V_{IH} , mientras que la señal $\overline{\text{PGM}}$ está a V_{IL} y V_{pp} a 21 voltios.

Algoritmo inteligente de programación.

El algoritmo de programación que nos permite ahorrar un tiempo importante en la grabación respecto al método standar. Dicha reducción se encuentra en el orden de un minuto y medio. Un flujograma de dicho método está descrito en este mismo apartado siendo compatible con el algoritmo inteligente empleado por la 27128.

Este algoritmo emplea dos pulsos distintos de programación. En primer lugar realizará un pulso de 1 ms. A continuación verificará si la información ha sido grabada. En caso contrario volverá a lanzar un nuevo pulso repitiendo este proceso hasta un total de 15

veces. Si llegado a este tope aún la información introducida no ha sido recogida por el dispositivo, se enviará un pulso de una duración de $4 \times$ "número de veces que se han lanzados los pulsos de 1 ms". Si aún así persistiese la no verificación de byte, la EPROM sería desechada por considerarla estropeada.

En el algoritmo inteligente la tensión a aplicar en la patilla de polarización V_{cc} va a cambiar de 5 a 6 voltios y la entrada V_{pp} se encontrará permanentemente a 21 voltios.

Reconocedor inteligente

La 2764 viene provista de un modo de identificación mediante programación. Dicha característica es aprovechada por los programadores automáticos para el reconocimiento de la EPROM y, consecuentemente, para la aplicación del algoritmo óptimo que le permita la mayor rapidez y garantía de un buen grabado. Para activar este módulo el equipo debe proporcionar una tensión entre 11.5 a 12.5 voltios en la línea de direccionamiento A_9 . Dos bytes dados en una secuencia lógica, identifican al dispositivo. Dichos bytes se obtienen aplicando un pulso a nivel bajo en la línea A_0 ; con esta señal se obtiene el código de identifica-

ción de manufactura , mientras que con un pulso a nivel alto en la misma patilla de direccionamiento se obtiene el código de identificación del dispositivo. Durante este proceso, toda las demás señales de direccionamiento deben estar a nivel bajo TTL.

A.C. PROGRAMMING CHARACTERISTICS: $T_a = 25 \pm 5^\circ\text{C}$, $V_{cc} = 5V \pm 5\%$, $V_{pp} = 21V \pm 0.5V$ (see Note 1)

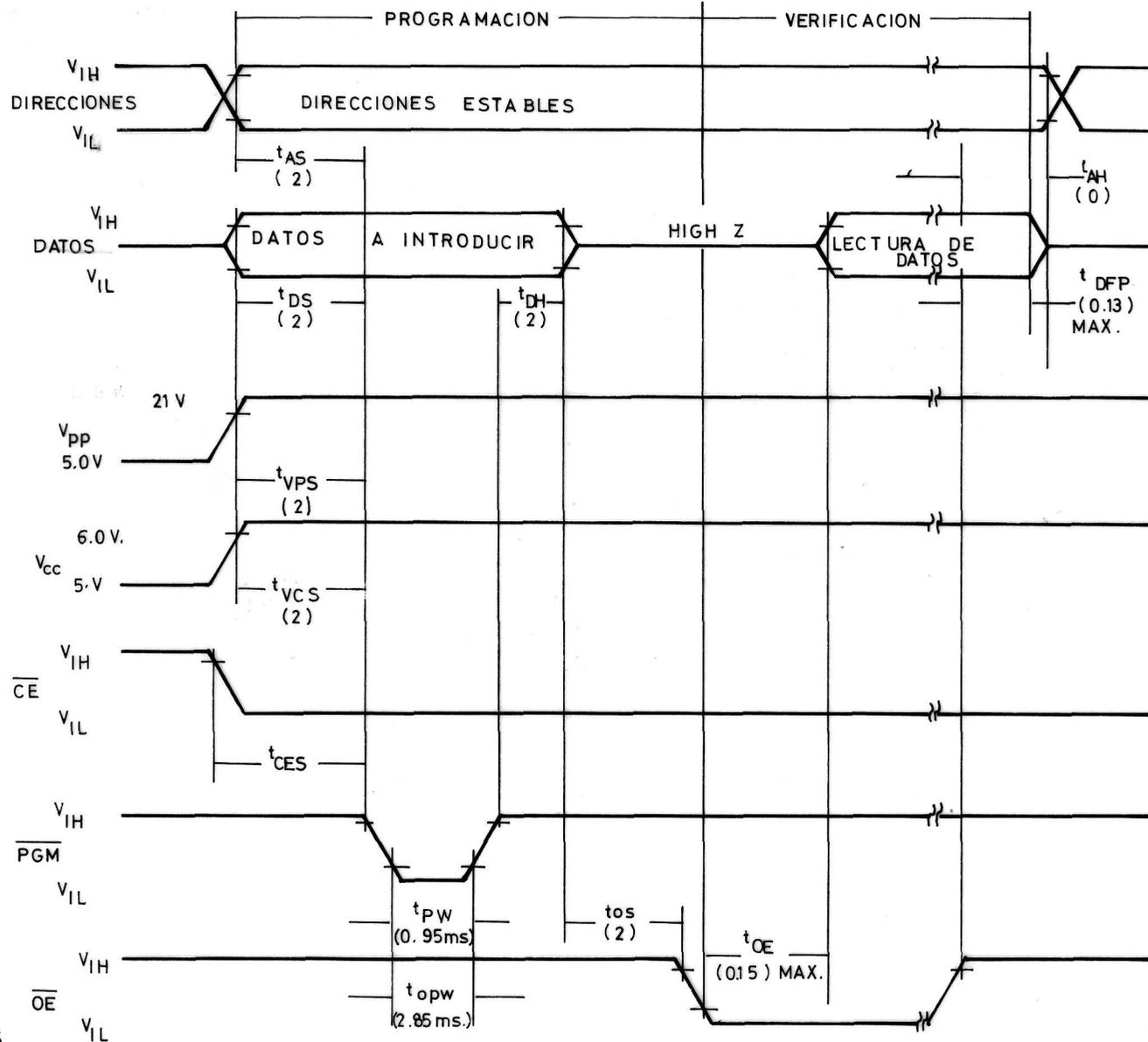
| Symbol | Parameter | Limits | | | | Test Conditions ¹ |
|------------|-------------------------------------|--------|------|------|---------------|------------------------------|
| | | Min. | Typ. | Max. | Unit | |
| t_{AS} | Address Setup Time | 2 | | | μs | |
| t_{OES} | \overline{OE} Setup Time | 2 | | | μs | |
| t_{OS} | Data Setup Time | 2 | | | μs | |
| t_{AH} | Address Hold Time | 0 | | | μs | |
| t_{DH} | Data Hold Time | 2 | | | μs | |
| t_{DF}^2 | Output Enable to Output Float Delay | 0 | | 130 | ns | |
| t_{VS} | V_{pp} Setup Time | 2 | | | μs | |
| t_{PW} | PGM Pulse Width During Programming | 45 | 50 | 55 | ms | |
| t_{CES} | \overline{CE} Setup Time | 2 | | | μs | |
| t_{OE} | Data Valid from \overline{OE} | | | 150 | ns | |

A.C. CHARACTERISTICS

| Symbol | Parameter | 2764-2 Limits | | 2764-25 & 2764 Limits | | 2764-30 & 2764-3 Limits | | 2764-45 & 2764-4 Limits | | Unit | Test Conditions |
|------------|---|---------------|-----|-----------------------|-----|-------------------------|-----|-------------------------|-----|------|--|
| | | Min | Max | Min | Max | Min | Max | Min | Max | | |
| t_{ACC} | Address to Output Delay | | 200 | | 250 | | 300 | | 450 | ns | $\overline{CE} = \overline{OE} = V_{IL}$ |
| t_{CE} | \overline{CE} to Output Delay | | 200 | | 250 | | 300 | | 450 | ns | $\overline{OE} = V_{IL}$ |
| t_{OE} | \overline{OE} to Output Delay | | 75 | | 100 | | 120 | | 150 | ns | $\overline{CE} = V_{IL}$ |
| t_{DF}^4 | \overline{OE} High to Output Float | 0 | 60 | 0 | 60 | 0 | 105 | 0 | 130 | ns | $\overline{CE} = V_{IL}$ |
| t_{OH} | Output Hold from Addresses. \overline{CE} or \overline{OE} Whichever Occurred First | 0 | | 0 | | 0 | | 0 | | ns | $\overline{CE} = \overline{OE} = V_{IL}$ |

| Identifier | Pins | A ₀ (10) | O ₇ (19) | O ₆ (18) | O ₅ (17) | O ₄ (16) | O ₃ (15) | O ₂ (13) | O ₁ (12) | O ₀ (11) | Hex Data |
|-------------------|----------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|----------|
| Manufacturer Code | V_{IL} | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 89 |
| Device Code | V_{IH} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 02 |

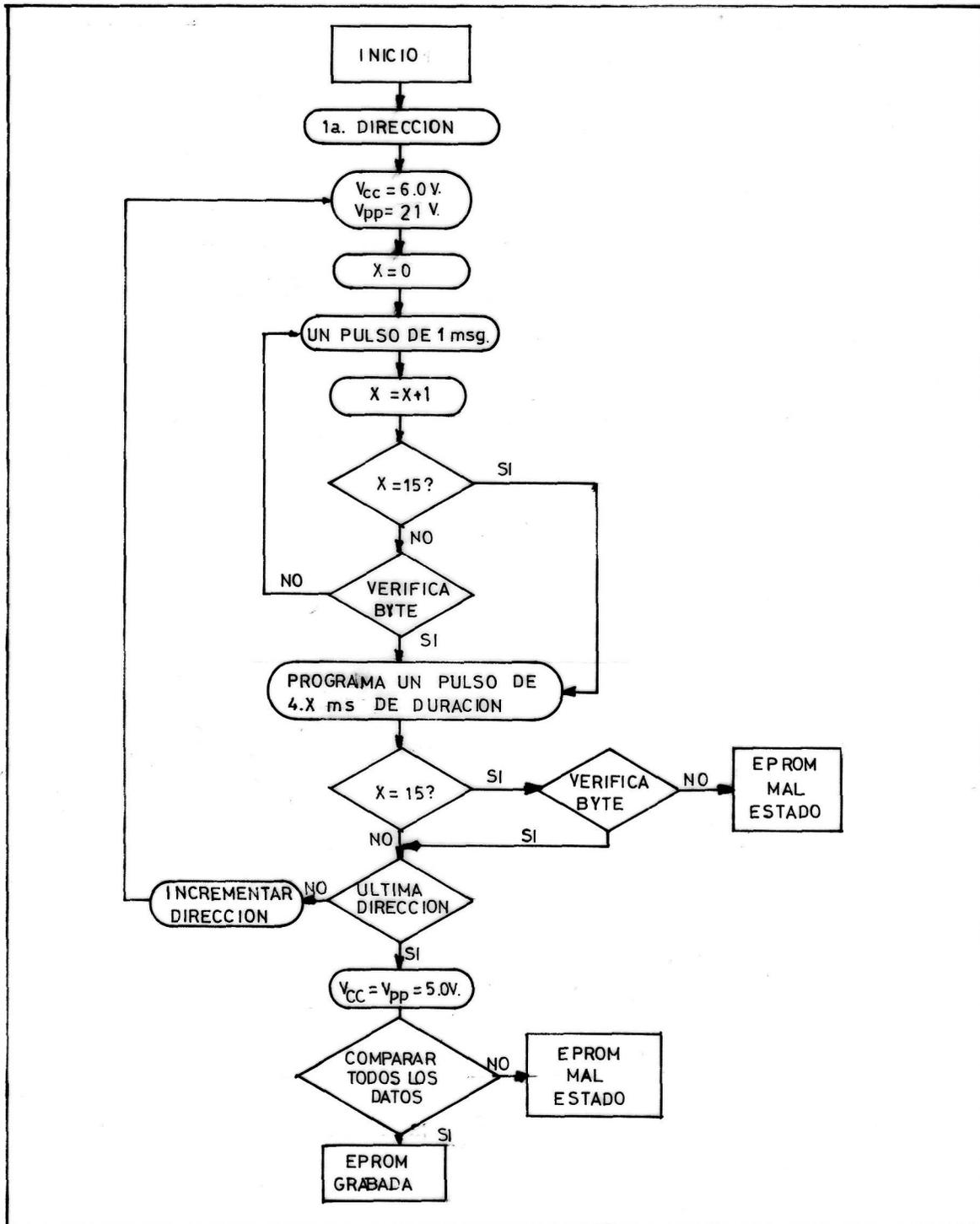
CRONOGRAMA ALGORITMO INTELIGENTE



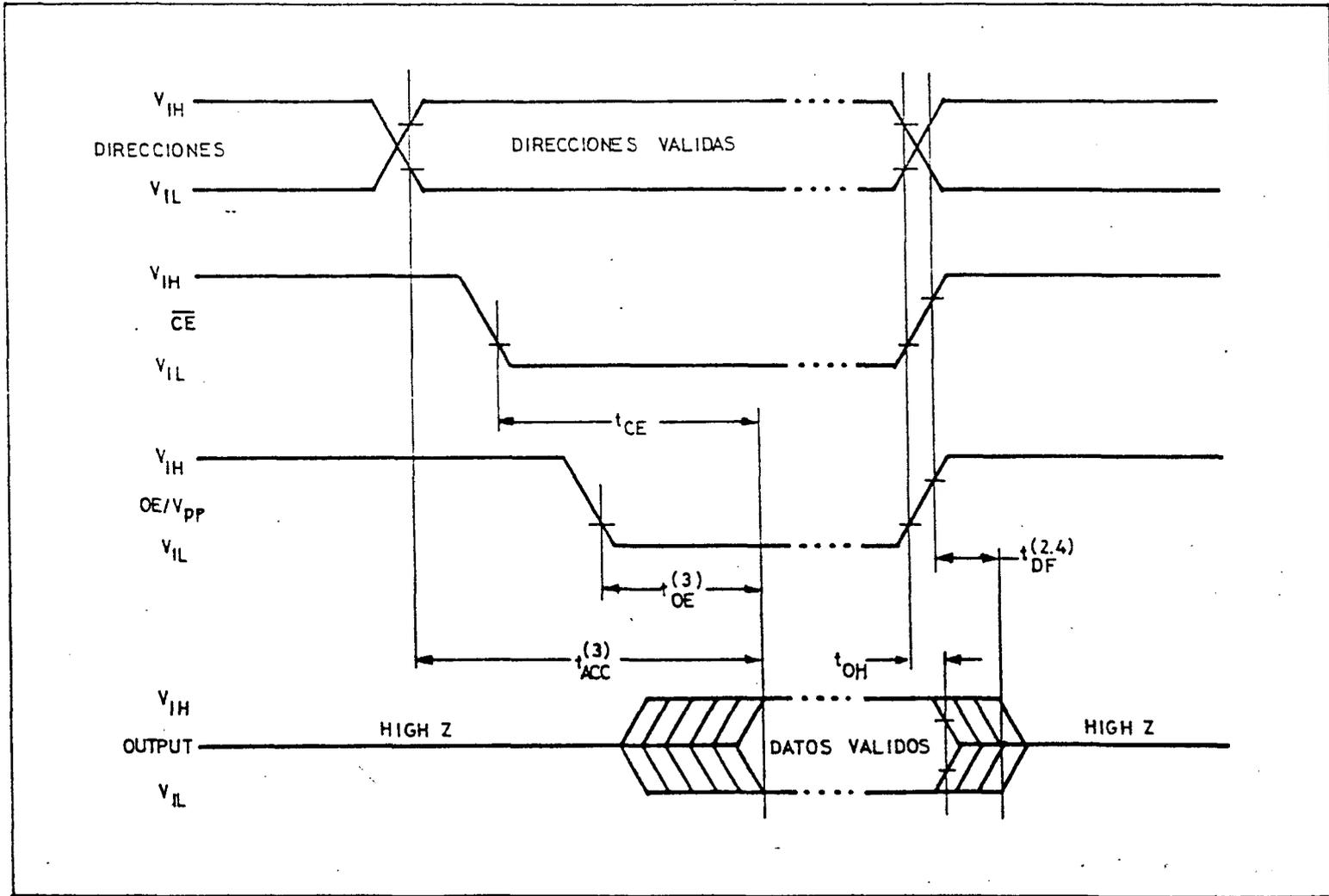
NOTAS

1. TODOS LOS TIEMPOS ENTRE () SON MINIMOS Y EN μ S SI NO SE ESPECIFICA LO CONTRARIO.
2. LOS NIVELES DE LAS ENTRADAS SON 0.8V. PARA V_{IL} Y 2V. PARA V_{IH} .

FLUJOGRAMA ALGORITMO INTELIGENTE.



MAPA LECTURA



103

2 7 6 4 A

PROM BORRABLE POR RAYOS ULTRAVIOLETA Y PROGRAMABLE
ELECTRICAMENTE CON CAPACIDAD DE 64K (8K*8).

La Intel 2764A con una sola tensión de polarización de 5 voltios. Es una memoria de sólo lectura borrable por rayos ultravioleta y reprogramable eléctricamente. Tiene una capacidad de 65536 bits. Es una moderna versión de la 2764 realizada con tecnología HMOSII-E lo cual reduce significativamente el tiempo de acceso, el consumo eléctrico y produce un mayor beneficio en la implementación de sistemas.

La típica 2764A es la misma que la 2746A-2 con un tiempo de acceso de 200 ns. Es una importante reducción si consideramos que la 2764 standar es de 250 ns. Dicha EPROM es ideal para los sistemas que requieren una alta velocidad como por ejemplo pueden ser los conocidos sistemas Intel que trabajan a 8 MHz. Esta particularidad permite trabajar sin que entre en estados WAIT. La 2764 también puede trabajar en sistemas de 12 MHz.

La 2764A también permite la reducción en el consumo de corriente. La máxima corriente consumida es de 60 mA mientras que en modo Standby es sólo de 20 mA.

| MODE | CS (20) | OE (22) | PGM (27) | A ₉ (24) | V _{pp} (1) | V _{CC} (28) | Outputs (11-13, 15-19) |
|-------------------------|-----------------|-----------------|-----------------|---------------------|---------------------|----------------------|------------------------|
| Read | V _{IL} | V _{IL} | V _{IH} | X | V _{CC} | V _{CC} | DOUT |
| Output Disable | V _{IL} | V _{IH} | V _{IH} | X | V _{CC} | V _{CC} | High Z |
| Standby | V _{IH} | X | X | X | V _{CC} | V _{CC} | High Z |
| Verify | V _{IL} | V _{IL} | V _{IH} | X | V _{pp} | V _{CC} | DOUT |
| Program Inhibit | V _{IH} | X | X | X | V _{pp} | V _{CC} | High Z |
| intelligent Identifier | V _{IL} | V _{IL} | V _{IH} | V _H | V _{CC} | V _{CC} | Code |
| intelligent Programming | V _{IL} | V _{IH} | V _{IL} | X | V _{pp} | V _{CC} | D _{IN} |

Este modo permite la reducción del consumo sin incrementar el tipo de acceso.

MODOS DE OPERACION

Los ocho modos de operación de la 2764A están contenidos en el diagrama A.25. Una sola tensión es necesaria para el modo de lectura. Todas las entradas son a niveles TTL, excepto para V_{pp} y para el identificador inteligente, en el cual deben aplicarse 12V. al pin A9.

Modo de lectura

La 2764A tiene dos funciones de control, ambas deben seguir una secuencia lógica para obtener los datos en la salida. El \overline{CE} es el que permite la selección del dispositivo para ser usado. La función de control \overline{OE} puede ser usado para habilitar los datos en sus salidas correspondientes, independientemente de la selección del dispositivo. Asumiendo que las direcciones son estables, el tiempo de direccionamiento (t_{ACC}) es igual al tiempo de espera para seleccionar el para la salida (t_{OE}). Los datos son válidos a la salida después de un tiempo t_{CE} desde la presentación de la señal \overline{OE} , asumiendo que \overline{CE} esté a

nivel bajo y las direcciones sean estables por un tiempo

$$t_{ACC} - t_{OE}$$

Modo Standby

La 2764A tiene un modo de funcionamiento que reduce el consumo de corriente desde 100 mA. a 40 mA. La 2764A se encuentra en este modo cuando hay una señal alta TTL en la entrada \overline{CE} . Estando en este modo, las salidas se encuentran en estado de alta impedancia, independientemente de la entrada \overline{OE} .

Output OR-Tieing

La razón de que estas EPROMS sean usadas en configuraciones que requieren varias unidades 2764A es que poseen dos líneas de control que permiten la conexión de múltiples memorias. Las dos líneas de control permiten:

- a) Disminuir en lo posible la disipación de la memoria.
- b) Asegurar la independencia de las salidas del bus del sistema.

Estas dos funciones de control permiten una

mayor eficacia a la hora de seleccionar los diferentes dispositivos. La señal \overline{OE} puede ser conectada a las señales READ que configuran el bus del sistema.

Modo de programación

La 2764A se encuentra en modo de programación cuando V_{pp} está a 12.5V., \overline{CE} y \overline{PGM} a nivel bajo TTL. Los datos serán programados cuando se apliquen ocho bits en paralelo a los pines de salida de datos. Los niveles requeridos para las señales de direcciones y de datos son TTL.

Programación Stándar

Para la programación, \overline{CE} debe estar a nivel bajo durante todo el proceso, mientras que V_{pp} debe estar a 12.5V. Cuando las direcciones y los datos son estables, un pulso a nivel bajo de 50 ms. aplicado a la entrada \overline{PGM} . Un pulso debe ser aplicado para cada posición de memoria a programar. El periodo de duración del pulso no debe durar más de 55 ms.

La programación de múltiples 2764A en paralelo con un mismo dato puede ser fácilmente realizado. Como las entradas están conectadas en paralelo para la programación con un mismo dato, un

pulso aplicado a $\overline{\text{PGM}}$ realiza la grabación del dato.

Verificación.

Para la verificación de que la información programada es la correcta, el modo de operación es el siguiente: la señal $\overline{\text{CE}}$ y la $\overline{\text{OE}}$ al nivel V_{IH}, mientras que la señal $\overline{\text{PGM}}$ está a V_{IL} y V_{pp} a 12.5 voltios.

Algoritmo inteligente de programación.

El algoritmo de programación que nos permite ahorrar un tiempo importante en la grabación respecto al método standar. Dicha reducción se encuentra en el orden de un minuto y medio. Un flujograma de dicho método está descrito en este mismo apartado siendo compatible con el algoritmo inteligente empleado por la 27128.

Este algoritmo emplea dos pulsos distintos de programación. En primer lugar realizará un pulso de 1 ms. A continuación verificará si la información ha sido grabada. En caso contrario volverá a lanzar un nuevo pulso repitiendo este proceso hasta un total de 15 veces. Si llegado a este tope aún la información introducida no ha sido recogida por el dispositivo, se enviará un pulso de una duración de 4 x "número de veces que se han lanzados los pulsos de 1 ms". Si aún

así persistiese la no verificación de byte, la EPROM sería desechada por considerarla estropeada.

En el algoritmo inteligente la tensión a aplicar en la patilla de polarización Vcc va a cambiar de 5 a 6 voltios y la entrada Vpp se encontrará permanentemente a 12.5 voltios.

Reconocedor inteligente

La 2764A viene provista de un modo de identificación mediante programación. Dicha característica es aprovechada por los programadores automáticos para el reconocimiento de la EPROM y, consecuentemente, para la aplicación del algoritmo óptimo que le permita la mayor rapidez y garantía de un buen grabado. Para activar este módulo el equipo debe proporcionar una tensión entre 11.5 a 12.5 voltios en la línea de direccionamiento A9. Dos bytes dados en una secuencia lógica, identifican al dispositivo. Dichos bytes se obtienen aplicando un pulso a nivel bajo en la línea A0; con esta señal se obtiene el código de identificación de manufactura, mientras que con un pulso a nivel alto en la misma patilla de direccionamiento se obtiene el código de identificación del dispositivo. Durante este proceso, toda las demás señales de direccionamien-

to deben estar a nivel bajo TTL.

2764A Intelligent Identifier™ Bytes

| Identifier | Pins | 2764A Intelligent Identifier™ Bytes | | | | | | | | | Hex Data |
|-------------------|-----------------|-------------------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|----------|
| | | A ₀ (10) | O ₇ (19) | O ₆ (18) | O ₅ (17) | O ₄ (16) | O ₃ (15) | O ₂ (13) | O ₁ (12) | O ₀ (11) | |
| Manufacturer Code | V _{IL} | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 89 | |
| Device Code | V _{IH} | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 08 | |

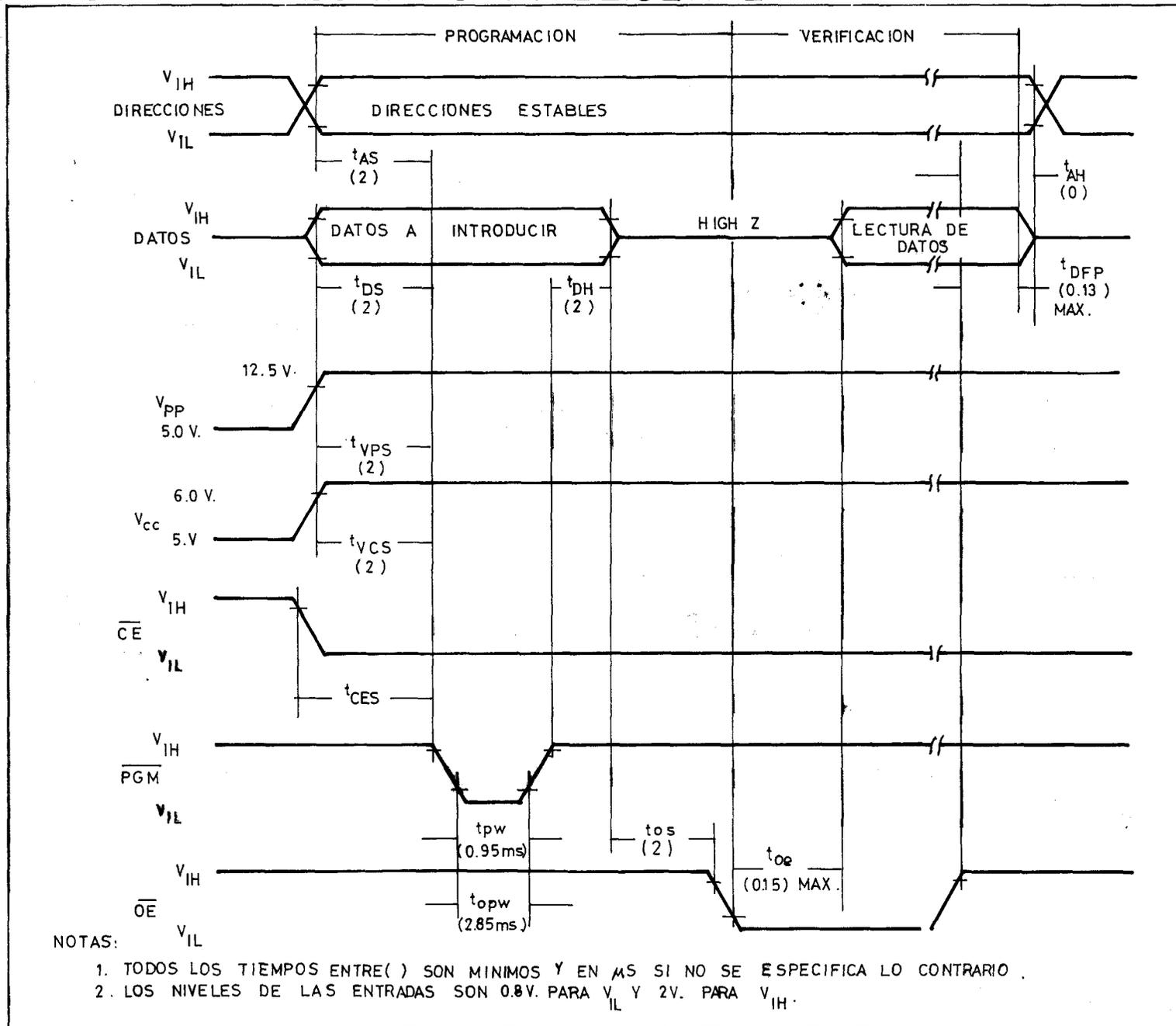
NOTES:

1. A₉ = 12.0V ±0.5V
2. A₁-A₆, A₁₀-A₁₃, \overline{CE} , \overline{OE} = V_{IL}
3. A₁₄ = V_{IH} or V_{IL}

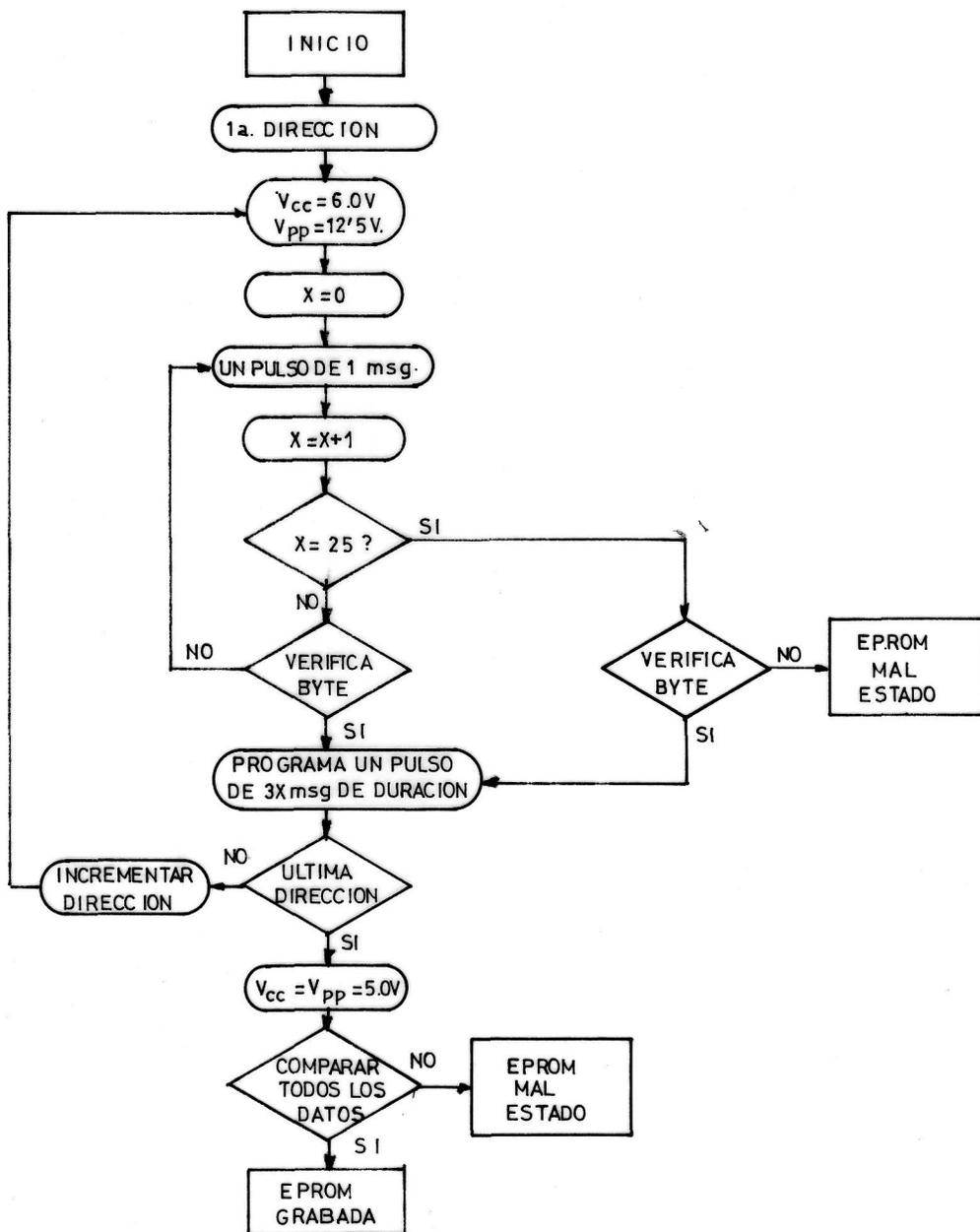
A.C. CHARACTERISTICS

| Symbol | Parameter | 2764A-1 Limits | | 2764A-20 & 2764A-2 Limits | | 2764A-25 & 2764A Limits | | 2764A-30 & 2764A-3 Limits | | 2764A-45 & 2764A-4 Limits | | Unit | Test Conditions |
|------------------------------|--|----------------|-----|---------------------------|-----|-------------------------|-----|---------------------------|-----|---------------------------|-----|------|--|
| | | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | | |
| t _{ACC} | Address to Output Delay | | 180 | | 200 | | 250 | | 300 | | 450 | ns | $\overline{CE} = \overline{OE} = V_{IL}$ |
| t _{CE} | \overline{CE} to Output Delay | | 180 | | 200 | | 250 | | 300 | | 450 | ns | $\overline{OE} = V_{IL}$ |
| t _{OE} | \overline{OE} to Output Delay | | 65 | | 75 | | 100 | | 120 | | 150 | ns | $\overline{CE} = V_{IL}$ |
| t _{DF} ⁴ | \overline{OE} High to Output Float | 0 | 55 | 0 | 55 | 0 | 60 | 0 | 105 | 0 | 130 | ns | $\overline{CE} = V_{IL}$ |
| t _{OH} | Output Hold from Addresses \overline{CE} or \overline{OE} Whichever Occurred First | 0 | | 0 | | 0 | | 0 | | 0 | | ns | $\overline{CE} = \overline{OE} = V_{IL}$ |

CRONOGRAMA ALGORITMO INTELIGENTE



FLUJOGRAMA ALGORITMO INTELIGENTE



27128

PROM BORRABLE POR RAYOS ULTRAVIOLETA Y PROGRAMABLE POR RAYOS ELECTRICAMENTE.

La Intel 27128 requiere solamente 5V. para su normal funcionamiento. Tiene una capacidad de 131072 bits que pueden ser borrados por rayos ultravioleta y programables eléctricamente. La 27128 estándar tiene un tiempo de acceso de 250ns. que la hace compatible con los sistemas de altas velocidades. Puede operar sin la necesidad de entrar en estados WAIT. Es compatible con los sistemas que trabajan hasta 12 Mhz.

Una característica importante de la 27128 es la separación de las señales de control \overline{OE} y \overline{CE} . La señal OE elimina la salida del bus del sistema. Además permite configuraciones complejas gracias a la entrada \overline{CE} .

La 27128 tiene un estado de permanencia llamado Standby, que reduce el consumo sin incrementar por ello el tiempo de acceso. La máxima corriente que consume es de 100 mA. en modo activo, mientras que la máxima corriente en modo Standby es de solo 40 mA. Este modo es seleccionado cuando se aplica una señal alta en la entrada \overline{CE} .

La 27128 admite una tolerancia en su tensión de polarización de ± 10 V. mientras que en otras la tolerancia es de tan solo de ± 5 V.

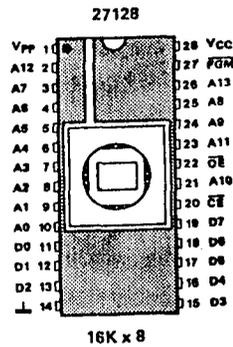
La 27128 esta fabricada con tecnología HMOS-E.

Patillaje 27128

MODE SELECTION

| Mode | Pins | CE (20) | OE (22) | PGM (27) | A ₉ (24) | V _{pp} (1) | V _{CC} (28) | Outputs (11-13, 15-19) |
|-------------------------|------|-----------------|-----------------|-----------------|---------------------|---------------------|----------------------|------------------------|
| Read | | V _{IL} | V _H | V _H | X | V _{CC} | V _{CC} | Output |
| Output Disable | | V _{IL} | V _H | V _H | X | V _{CC} | V _{CC} | High Z |
| Standby | | V _H | X | X | X | V _{CC} | V _{CC} | High Z |
| Program | | V _{IL} | V _H | V _{IL} | X | V _{pp} | V _{CC} | DN |
| Verify | | V _{IL} | V _H | V _H | X | V _{pp} | V _{CC} | Output |
| Program Inhibit | | V _H | X | X | X | V _{pp} | V _{CC} | High Z |
| Intelligent Identifier | | V _{IL} | V _{IL} | V _H | V _{CC} | V _{CC} | V _{CC} | Code |
| Intelligent Programming | | V _{IL} | V _H | V _{IL} | X | V _{pp} | V _{CC} | DN |

NOTES:
 1. X can be V_H or V_{IL}
 2. V_H = 12.0V \pm 0.5V



MODOS DE OPERACION

Los ocho modos de operación de la 27128 están contenidos en el diagrama A.25. Una sola tensión es necesaria para el modo de lectura. Todas las entradas son a niveles TTL, excepto para Vpp y para el identificador inteligente, en el cual deben aplicarse 12V. al pin A9.

Modo de lectura

La 27128 tiene dos funciones de control, ambas deben seguir una secuencia lógica para obtener los datos en la salida. El CE es el que permite la selección del dispositivo para ser usado. La función de

control \overline{OE} puede ser usado para habilitar los datos en sus salidas correspondientes, independientemente de la selección del dispositivo. Asumiendo que las direcciones son estables, el tiempo de direccionamiento (t_{ACC}) es igual al tiempo de espera para seleccionar el \overline{CE} para la salida (t_{CE}). Los datos son válidos a la salida después de un tiempo t_{OE} desde la presentación de la señal \overline{OE} , asumiendo que \overline{CE} este a nivel bajo y las direcciones sean estables por un tiempo $t_{ACC} - t_{OE}$.

Modo Standby

La 27128 tiene un modo de funcionamiento que reduce el consumo de corriente desde 100 mA. a 40 mA. La 27128 se encuentra en este modo cuando hay una señal alta TTL en la entrada \overline{CE} . Estando en este modo, las salidas se encuentran en estado de alta impedancia, independientemente de la entrada \overline{OE} .

Output OR-Tieing

La razón de que estas EPROMS sean usadas en configuraciones que requieren varias unidades 27128 es que poseen dos líneas de control que permiten la

conexión de multiples memorias. Las dos líneas de control permiten:

a) Disminuir en lo posible la disipación de la memoria.

b) Asegurar la independencia de las salidas del bus del sistema.

Estas dos funciones de control permiten una mayor eficacia a la hora de seleccionar los diferentes dispositivos. La señal \overline{OE} puede ser conectada a las señales READ que configuran el bus del sistema.

Modo de programación

La 27128 se encuentra en modo de programación cuando V_{pp} está a 21V., \overline{CE} y \overline{PGM} a nivel bajo TTL. Los datos serán programados cuando se apliquen ocho bits en paralelo a los pines de salida de datos. Los niveles requeridos para las señales de direcciones y de datos son TTL.

Programación Stándar

Para la programación, \overline{CE} debe estar a nivel bajo durante todo el proceso, mientras que V_{pp} debe estar a 21V. Cuando las direcciones y los datos son estables, un pulso a nivel bajo de 50 ms. aplicado a la

entrada $\overline{\text{PGM}}$. Un pulso debe ser aplicado para cada posición de memoria a programar. El periodo de duración del pulso no debe durar más de 55 ms.

La programación de múltiples 27128 en paralelo con un mismo dato puede ser fácilmente realizado. Como las entradas están conectadas en paralelo para la programación con un mismo dato, un pulso aplicado a $\overline{\text{PGM}}$ realiza la grabación del dato.

Verificación.

Para la verificación de que la información programada es la correcta, el modo de operación es el siguiente: la señal $\overline{\text{CE}}$ y la $\overline{\text{OE}}$ al nivel V_{IH} , mientras que la señal $\overline{\text{PGM}}$ está a V_{IL} y V_{pp} a 21 voltios.

Algoritmo inteligente de programación.

El algoritmo de programación que nos permite ahorrar un tiempo importante en la grabación respecto al método standar. Dicha reducción se encuentra en el orden de un minuto y medio. Un flujograma de dicho método está descrito en este mismo apartado siendo compatible con el algoritmo inteligente empleado por la 27128.

Este algoritmo emplea dos pulsos distintos de programación. En primer lugar realizará un pulso de 1 ms. A continuación verificará si la información ha sido grabada. En caso contrario volverá a lanzar un nuevo pulso repitiendo este proceso hasta un total de 15 veces. Si llegado a este tope aún la información introducida no ha sido recogida por el dispositivo, se enviará un pulso de una duración de $4 \times$ "número de veces que se han lanzados los pulsos de 1 ms". Si aún así persistiese la no verificación de byte, la EPROM sería desechada por considerarla estropeada.

En el algoritmo inteligente la tensión a aplicar en la patilla de polarización Vcc va a cambiar de 5 a 6 voltios y la entrada Vpp se encontrará permanentemente a 21 voltios.

Reconocedor inteligente

La 27128 viene provista de un modo de identificación mediante programación. Dicha característica es aprovechada por los programadores automáticos para el reconocimiento de la EPROM y, consecuentemente, para la aplicación del algoritmo óptimo que le permita la mayor rapidez y garantía de un buen grabado. Para activar este módulo el equipo debe proporcionar,

una tensión entre 11.5 a 12.5 voltios en la línea de direccionamiento A9. Dos bytes dados en una secuencia lógica, identifican al dispositivo. Dichos bytes se obtienen aplicando un pulso a nivel bajo en la línea A0, con esta señal se obtiene el código de identificación de manufactura, mientras que con un pulso a nivel alto en la misma patilla de direccionamiento se obtiene el código de identificación del dispositivo. Durante este proceso, todas las demás señales de direccionamiento deben estar a nivel bajo TTL.

D.C. PROGRAMMING CHARACTERISTICS: $T_A = 25 \pm 5^\circ\text{C}$, $V_{CC} = 6.0\text{V} \pm 0.25\text{V}$, $V_{PP} = 21\text{V} \pm 0.5\text{V}$

| Symbol | Parameter | Limits | | | Test Conditions (see Note 1) |
|-----------|---|--------|----------|---------------|---|
| | | Min. | Max. | Unit | |
| I_{LI} | Input Current (All Inputs) | | 10 | μA | $V_{IN} = V_{IL} \text{ or } V_{IH}$ |
| V_{IL} | Input Low Level (All Inputs) | -0.1 | 0.8 | V | |
| V_{IH} | Input High Level | 2.0 | V_{CC} | V | |
| V_{OL} | Output Low Voltage During Verify | | 0.45 | V | $I_{OL} = 2.1 \text{ mA}$ |
| V_{OH} | Output High Voltage During Verify | 2.4 | | V | $I_{OH} = -400 \mu\text{A}$ |
| I_{CC2} | V_{CC} Supply Current (Program & Verify) | | 100 | mA | |
| I_{PP2} | V_{PP} Supply Current (Program) | | 30 | mA | $\overline{CE} = V_{IL} = \overline{PGM}/\overline{WE}$ |
| V_{ID} | A ₉ intelligent Identifier Voltage | 11.5 | 12.5 | V | |

FLUJOGRAMA ALGORITMO INTELIGENTE.

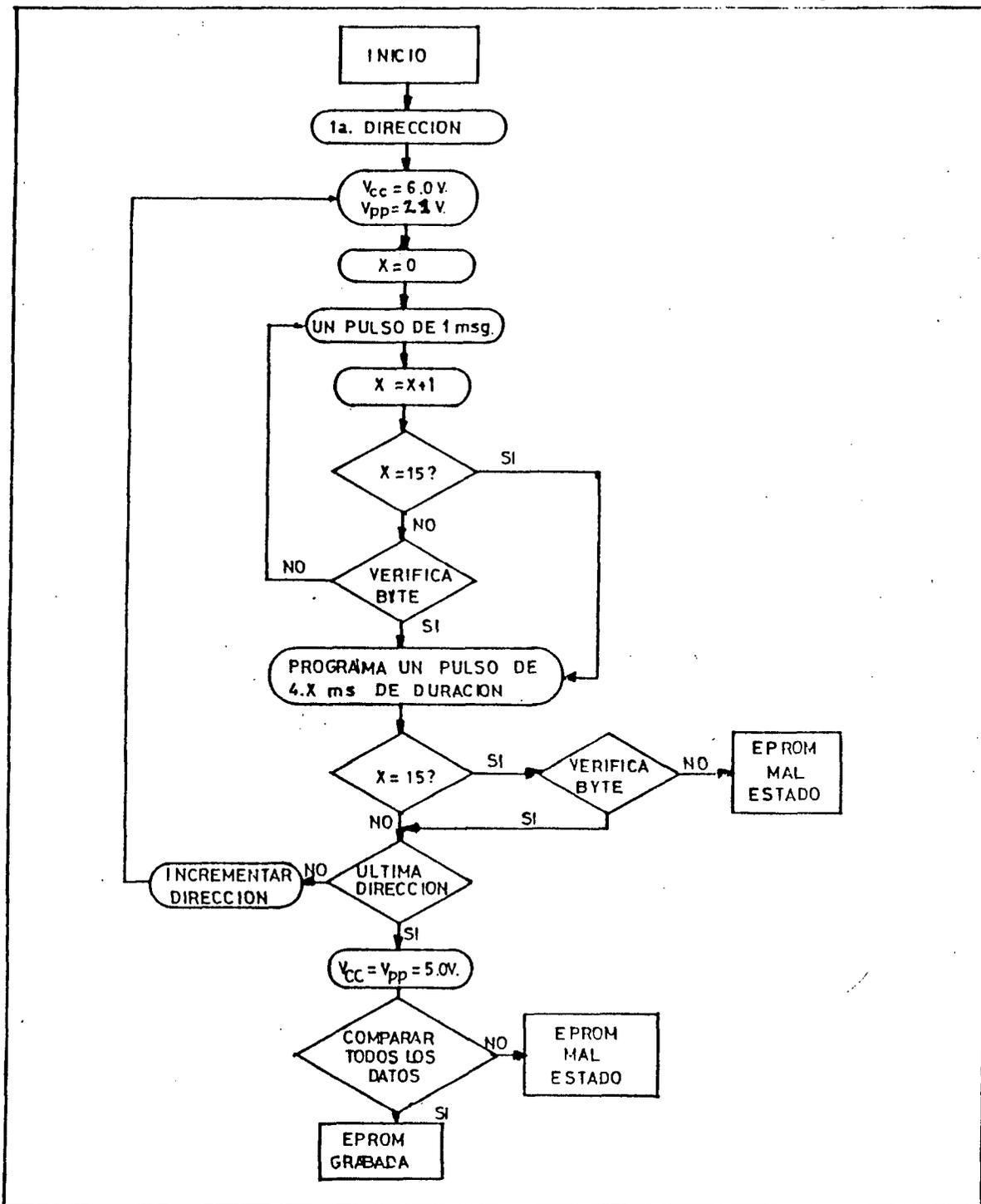
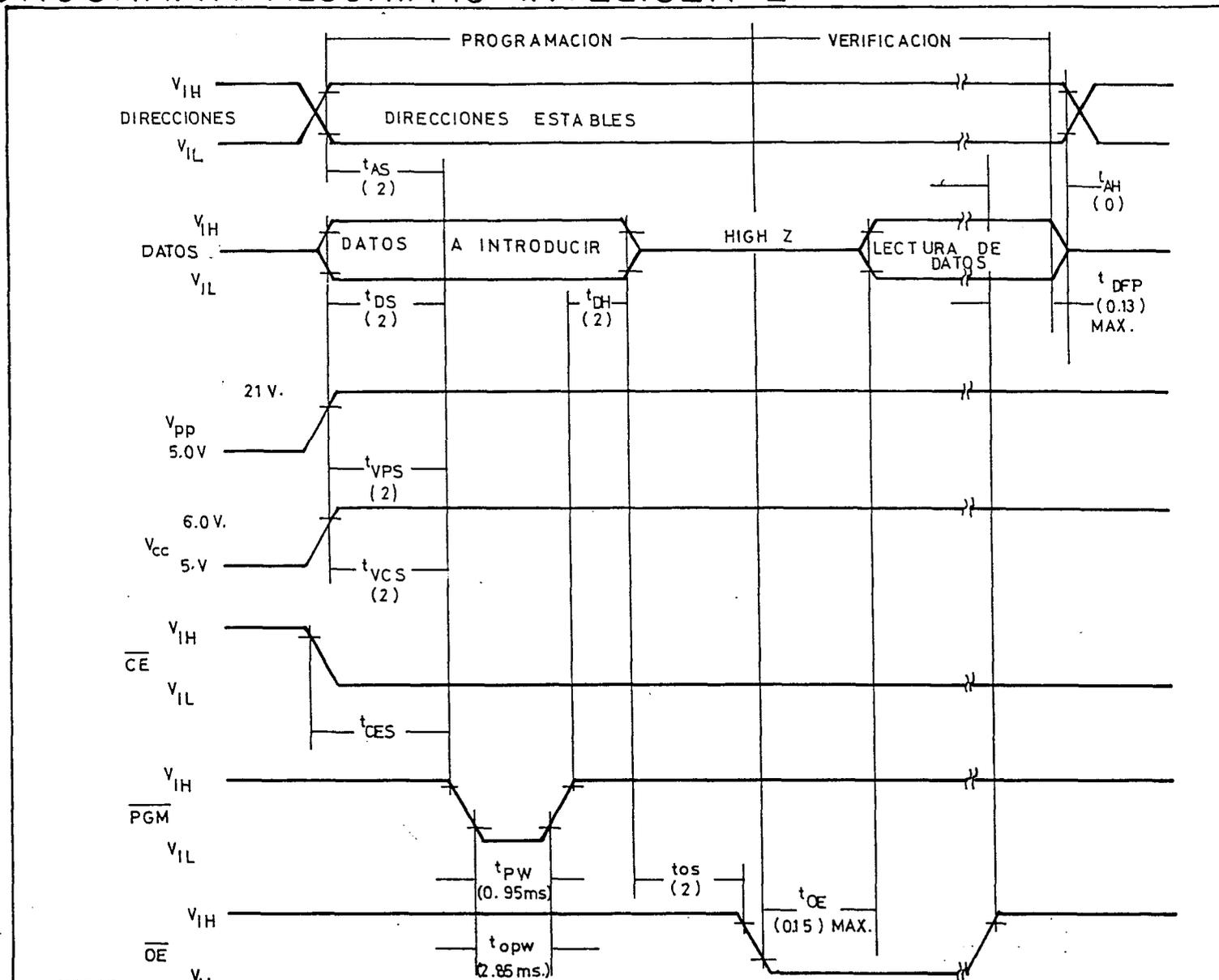


Diagrama n°

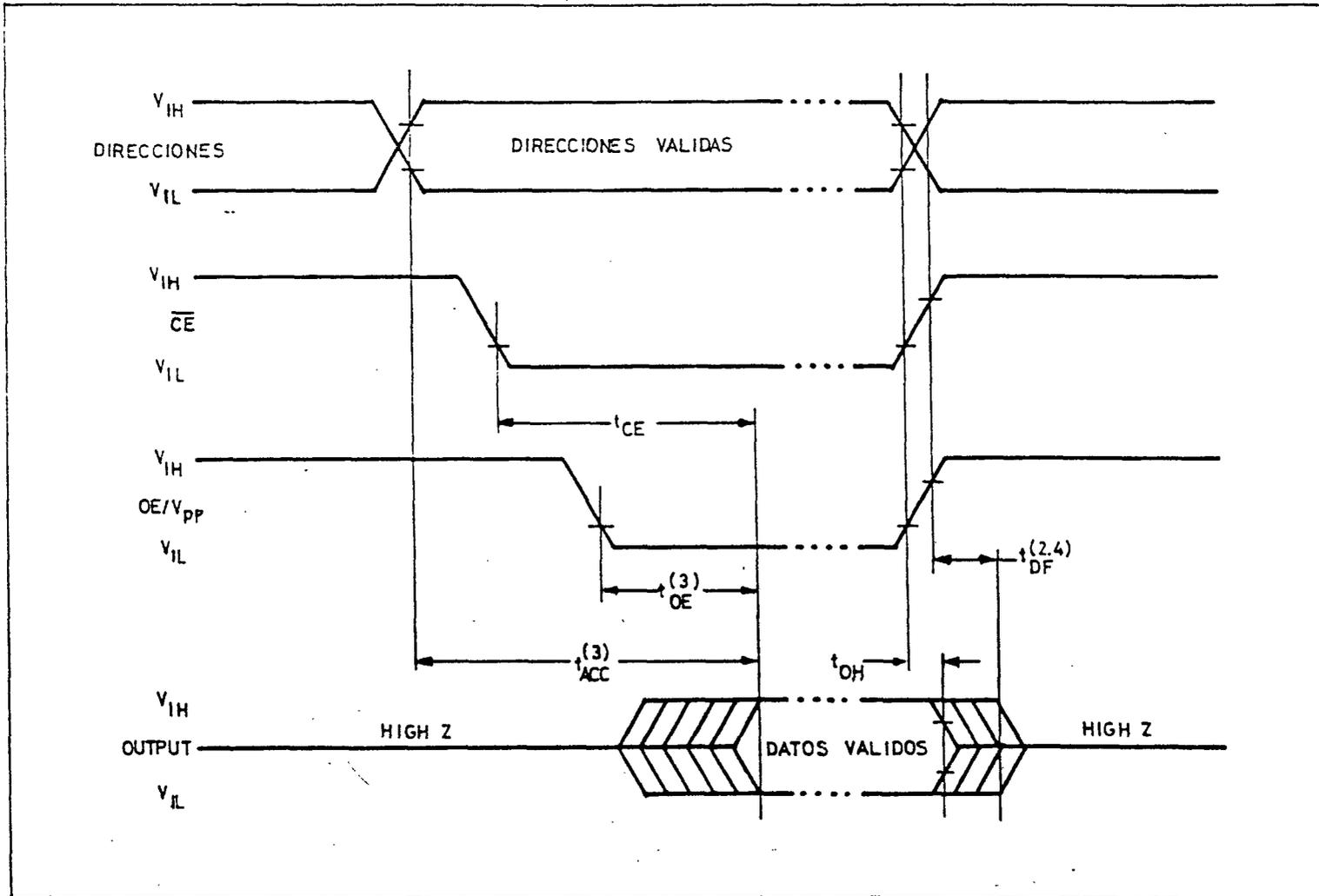
CRONOGRAMA ALGORITMO INTELIGENTE



NOTAS

1. TODOS LOS TIEMPOS ENTRE () SON MINIMOS Y EN μS SI NO SE ESPECIFICA LO CONTRARIO.
2. LOS NIVELES DE LAS ENTRADAS SON 0.8V. PARA V_{IL} Y 2V. PARA V_{IH} .

MAPA LECTURA



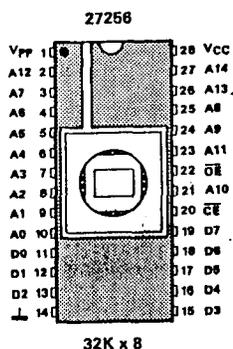
27256

FROM BORRABLE POR RAYOS ULTRAVIOLETA Y PROGRAMABLE ELECTRICAMENTE CON CAPACIDAD DE 256 Kb. (32K*8).

La Intel 27256 es una memoria que utiliza una zona tensión de 5V. para la polarización. Es una memoria de solo lectura borrable por rayos ultravioleta y programable electricamente. La organización de las palabras es de 32K.*8 bits. Cada byte tiene un tiempo de acceso por debajo de los 250 ns. Es compatible en sistemas que requieren grandes velocidades sin la necesidad de entrar en estados WAIT. Es compatible directamente con la familia de microcontroladores 8051.

La 27256 tiene una arquitectura totalmente diferente a las anteriores, llegandose a un alto grado de implementación. La utilización de la 27256 en sistemas que requieran alta capacidad disminuyen el coste de implementación de los sistemas.

Patillaje de la 27256



MODOS DE OPERACION

Los ocho modos de operación de la 27256 están contenidos en el diagrama final. Una sola tensión es necesaria para el modo de lectura. Todas las entradas son a niveles TTL, excepto para Vpp y para el identificador inteligente, en el cual deben aplicarse 12V. al pin A9.

Modo de lectura

La 27256 tiene dos funciones de control, ambas deben seguir una secuencia lógica para obtener los datos en la salida. El \overline{CE} es el que permite la selección del dispositivo para ser usado. La función de control \overline{OE} puede ser usado para habilitar los datos en sus salidas correspondientes, independientemente de la selección del dispositivo. Asumiendo que las direcciones son estables, el tiempo de direccionamiento (t_{ACC}) es igual al tiempo de espera para seleccionar el CE para la salida (t_{CE}). Los datos son válidos a la salida después de un tiempo t_{OE} desde la presentación de la señal \overline{OE} , asumiendo que CE este a nivel bajo y las direcciones sean estables por un tiempo $t_{ACC} - t_{OE}$.

Modo Standby

La 27256 tiene un modo de funcionamiento que reduce el consumo de corriente desde 100 mA. a 40 mA. La 27256 se encuentra en este modo cuando hay una señal alta TTL en la entrada \overline{CE} . Estando en este modo, las salidas se encuentran en estado de alta impedancia, independientemente de la entrada \overline{OE} .

Output OR-Tieing

La razón de que estas EPROMS sean usadas en configuraciones que requieren varias unidades 27256 es que poseen dos líneas de control que permiten la conexión de multiples memorias. Las dos líneas de control permiten:

- a) Disminuir en lo posible la disipación de la memoria.
- b) Asegurar la independendencia de las salidas del bus del sistema.

Estas dos funciones de control permiten una mayor eficacia a la hora de seleccionar los diferentes dispositivos. La señal \overline{OE} puede ser conectada a las señales READ que configuran el bus del sistema.

Modo de programación

La 27256 se encuentra en modo de programación cuando V_{pp} está a 12.5V., y la entrada \overline{CE} envía un pulso a nivel bajo TTL teniendo estos pulsos una duración típica de 1 ms. Los datos serán programados cuando se apliquen ocho bits en paralelo a los pines de salida de datos. Los niveles requeridos para las señales de direcciones y de datos son TTL.

La programación de múltiples 27256 en paralelo con un mismo dato puede ser fácilmente realizado. Como las entradas están conectadas en paralelo para la programación con un mismo dato, un pulso aplicado a \overline{CE} realiza la grabación del dato.

Verificación.

Para la verificación de que la información programada es la correcta, el modo de operación es el siguiente: la señal \overline{CE} y la \overline{OE} al nivel V_{IH} , mientras que las demás señales están a V_{IL} y V_{pp} a 12.5 voltios.

Algoritmo inteligente de programación.

El algoritmo de programación que nos permite ahorrar un tiempo importante en la grabación respecto al método standar. Dicha reducción se encuentra en el

orden de un minuto y medio. Un flujograma de dicho método está descrito en este mismo apartado siendo compatible con el algoritmo inteligente empleado por la 27128.

Este algoritmo emplea dos pulsos distintos de programación. En primer lugar realizará un pulso de 1 ms. A continuación verificará si la información ha sido grabada. En caso contrario volverá a lanzar un nuevo pulso repitiendo este proceso hasta un total de 15 veces. Si llegado a este tope aún la información introducida no ha sido recogida por el dispositivo, se enviará un pulso de una duración de $4 \times$ "número de veces que se han lanzados los pulsos de 1 ms". Si aún así persistiese la no verificación de byte, la EPROM sería desechada por considerarla estropeada.

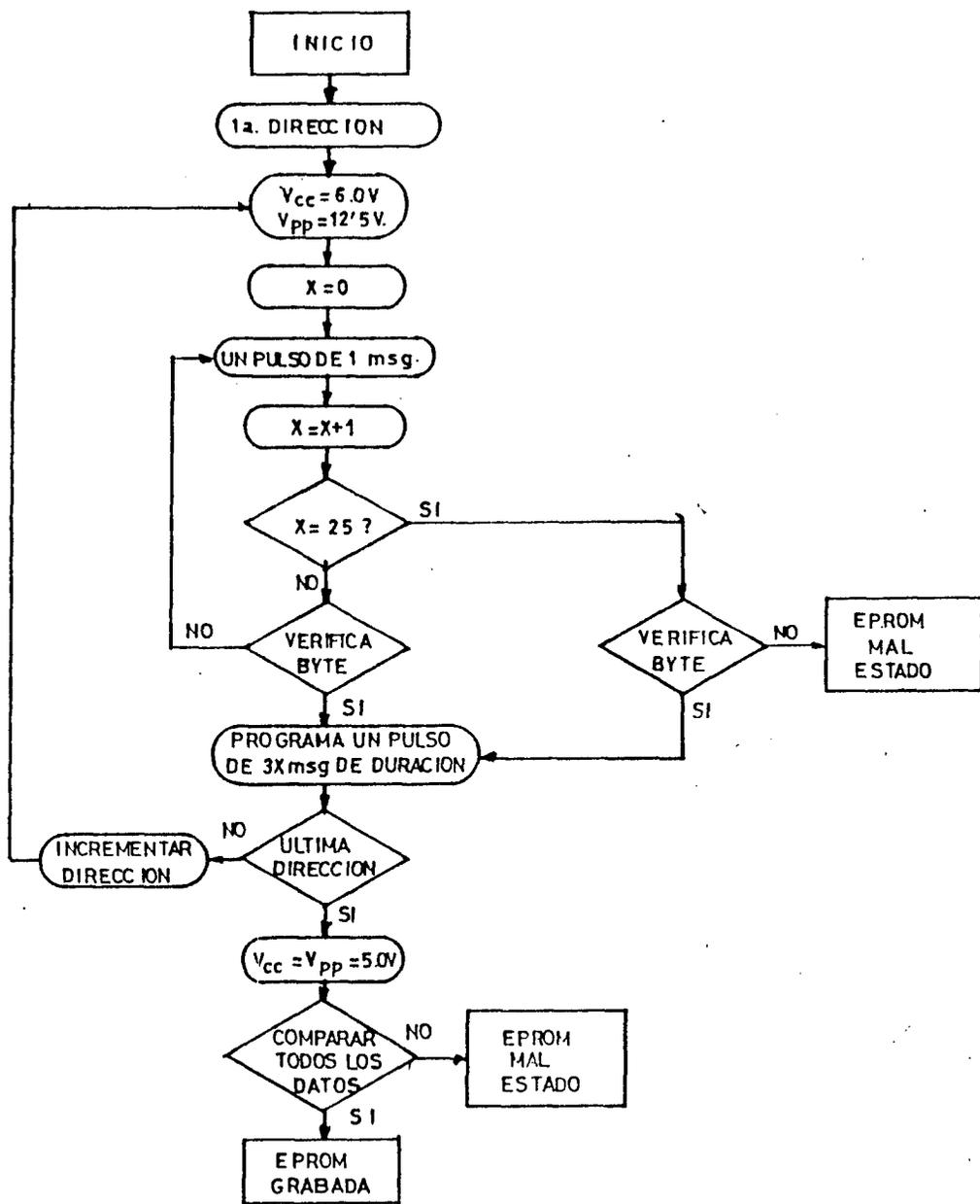
En el algoritmo inteligente la tensión a aplicar en la patilla de polarización Vcc va a cambiar de 5 a 6 voltios y la entrada Vpp se encontrará permanentemente a 12.5 voltios.

Reconocedor inteligente

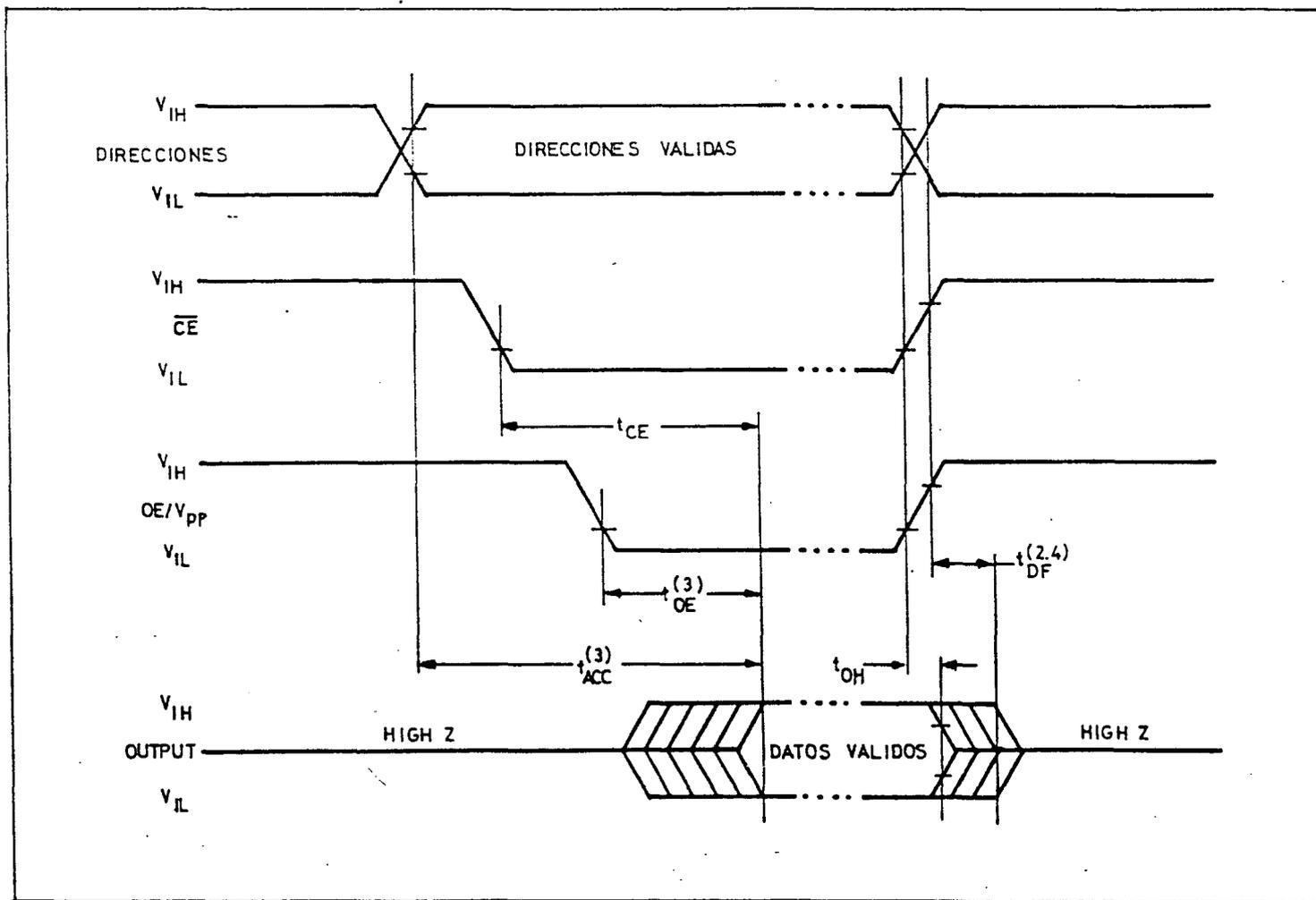
La 27256 viene provista de un modo de identificación mediante programación. Dicha característica es aprovechada por los programadores automáticos

para el reconocimiento de la EPROM y , consecuentemente, para la aplicación del algoritmo óptimo que le permita la mayor rapidez y garantía de un buen grabado. Para activar este módulo el equipo debe proporcionar una tensión entre 11.5 a 12.5 voltios en la línea de direccionamiento A9. Dos bytes dados en una secuencia lógica, identifican al dispositivo. Dichos bytes se obtienen aplicando un pulso a nivel bajo en la línea A0, con esta señal se obtiene el código de identificación de manufactura , mientras que con un pulso a nivel alto en la misma patilla de direccionamiento se obtiene el código de identificación del dispositivo. Durante este proceso, toda las demás señales de direccionamiento deben estar a nivel bajo TTL.

FLUJOGRAMA ALGORITMO INTELIGENTE



MAPA LECTURA

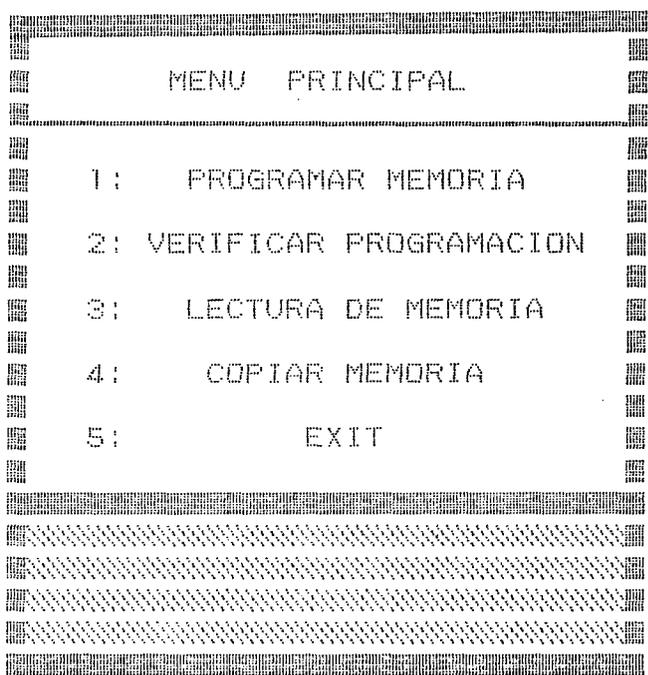


LISTADO PARA MSX

```

10 CLEAR 3000:DEFINT V
20 REM DRIGEN
30 REM PROGRAMADOR DE EPROMS
40 DIM V1(7,255)
50 KEY OFF
60 REM PRESENTACION ``
70 PLAY "T240EF+G+AB05C+D+E"
80 PLAY "N41N43N45N46N48N50N52N53"
90 REM PRESENTACION
100 CLS
110 LOCATE 3,6:PRINT "PROGRAMADOR UNIVERSAL DE EPROMS"
120 LOCATE 10,12:PRINT "Copyright 1987,(@)"
130 LOCATE 17,15:PRINT"by"
140 LOCATE 7,18:PRINT"JOSE MANUEL GARCIA MARTEL"
150 GOSUB 1170
160 CLS
170 PRINT "
180 PRINT "
190 PRINT "
200 PRINT "
210 PRINT "
220 PRINT "
230 PRINT "
240 PRINT "
250 PRINT "
260 PRINT "
270 PRINT "
280 PRINT "
290 PRINT "
300 PRINT "
310 PRINT "
320 PRINT "
330 PRINT "
340 PRINT "
350 PRINT "
360 PRINT "
370 PRINT "
380 A$=INKEY$:IF A$=""THEN 380
390 IF A$="1" THEN 450
400 IF A$="2" THEN 930
410 IF A$="3" THEN 940
420 IF A$="4" THEN 5970
430 IF A$="5" THEN 5790
440 GOTO 380
450 REM INFORMACION. OPCION 1
460 CLS:PRINT:PRINT:PRINT:PRINT
470 PRINT " CON ESTE PROGRAMA UD. PUEDE GRABAR"
480 PRINT " EPROM 27XXX. SI BIEN LO REALIZARA EN"
490 PRINT " SUCESIVOS PASOS SEGUN SE LE INDIQUE"
500 GOSUB 1170
510 CLS
520 LOCATE 1,3
530 PRINT "INSERTE LA EPROM A GRABAR EN EL ZOCALO"
540 GOSUB 1170
550 GOSUB 1220
560 LOCATE 0,1
570 PRINT "QUE EPROM VA A GRABAR? (INDIQUE INDICE)"
580 GOSUB 1370
590 REM DIRECCION INICIAL

```



```

600 GOSUB 1460
610 ZP=KI
620 REM DIRECCION FINAL
630 GOSUB 1680
640 ZH=RF
650 ZF=RF:ZI=ZP
660 GOSUB 2380
670 ZJ=DL
680 IF ZJ<=0 THEN 2480
690 ZM=ZJ
700 GOSUB 2410
710 GOSUB 2300
720 CB=NB:UD=KI:ZF=RF
730 ZI=UD:DT=UD:VM=UD
740 GOSUB 2570
750 UD=ZI
760 REM SUBPROGRAMA A GRABAR
770 REM DECIR ESTOY GRABANDO
780 GOSUB 2950
790 REM ELIGE ALGORITMO Y GRABA
800 GOSUB 2870
810 REM VERIFICA EN MODO DE LECTURA
820 GOSUB 3140
830 REM SI NO VERIFICA DECIRLO
840 IF LT=1 THEN 2000
850 REM DESCUENTA EL BLOQUE
860 CB=CB-1
870 IF CB<>0 THEN 730
880 CLS
890 LOCATE 5,5
900 PRINT "EPROM GRABADA"
910 GOSUB 1170
920 GOTO 160
930 REM *** OPCION 2 ***
940 REM OPCION 3
950 REM COPIAR EN PANTALLA LECTURA
960 CLS
970 LOCATE 0,3
980 PRINT"ESTA OPCION LE PERMITE LA LECTURA"
990 GOSUB 4830
1000 PRINT"EPROM"
1010 GOSUB 1170
1020 GOSUB5590
1030 REM TRASBACE DE VARIABLES
1040 UD=KI:CB=INT (DL/2048)+1
1050 DM=UD
1060 REM LECTURA DE MEMORIA
1070 GOSUB 4830
1080 REM TRASBACE DE VARIABLES
1090 RF=UD
1100 GOSUB4690
1110 UD=DM
1120 CB=CB-1
1130 IF CB>0 THEN GOTO 1050
1140 GOTO 160

```

```

1150 REM *** OPCION 4 ***
1160 PRINT
1170 REM APRETAR TECLA PARA CONTINUAR
1180 LOCATE 1,24
1190 PRINT " (Apretar una tecla para continuar)"
1200 A$=INKEY$:IF A$=""THEN1200
1210 RETURN
1220 REM TIPOS DE EPROMS A PANTALLA
1230 CLS
1240 LOCATE 0,5
1250 PRINT "1: 2716"
1250 PRINT
1270 PRINT "2: 2732A, P2732A"
1280 PRINT
1290 PRINT"3: 2764, P2764A, P2764
1300 PRINT
1310 PRINT"4: 2764A
1320 PRINT
1330 PRINT"5: 27128, 27128A"
1340 PRINT
1350 PRINT"6: 27256"
1360 RETURN
1370 REM PETICION TIPO DE EPROM
1380 A$=INKEY$ :IF A$=""THEN 1380
1390 IF A$="1"THEN MEMO=1:RETURN
1400 IF A$="2"THEN MEMO=2:RETURN
1410 IF A$="3"THEN MEMO=3:RETURN
1420 IF A$="4"THEN MEMO=4:RETURN
1430 IF A$="5"THEN MEMO=5:RETURN
1440 IF A$="6"THEN MEMO=6:RETURN
1450 GOTO 1380
1460 REM PETICION DIRECCION INICIAL
1470 DI$="&H"
1480 CLS
1490 LOCATE 4,5
1500 PRINT"INTRODUCIR DIRECCION INICIAL"
1510 LOCATE 8,7:PRINT" EN HEXADECIMAL"
1520 FOR I=0 TO 4
1530 DP$=INKEY$:IF DP$=""THEN 1530
1540 IF DP$=CHR$(32) THEN 1530
1550 IF I=4 AND DP$=CHR$(13) THEN 1640
1560 IF I=4 THEN 1530
1570 REM COMPROBAR SI ES HEX.
1580 GOSUB 3000
1590 IF L=0 THEN1530
1600 DI$=DI$+DP$
1610 LOCATE 12,10
1620 PRINTDI$
1630 NEXTI
1640 PK=VAL(MID$(DI$,1,4))
1650 B$="&H"+MID$(DI$,5,2)
1660 KI=PK*256+VAL(B$)
1670 RETURN
1680 REM PETICION DIRECCION FINAL
1690 DI$="&H"
1700 CLS

```

```

1710 LOCATE 4,5
1720 PRINT"INTRODUCIR DIRECCION FINAL  "
1730 LOCATE 8,7:PRINT" EN HEXADECIMAL"
1740 FOR I=0 TO 4
1750 DP$=INKEY$:IF DP$=""THEN 1750
1760 IF DP$=CHR$(32) THEN 1750
1770 IF I=4 AND DP$=CHR$(13) THEN 1860
1780 IF I=4 THEN 1750
1790 REM COMPROBAR SI ES HEX.
1800 GOSUB 3000
1810 IF L=0 THEN1750
1820 DI$=DI$+DP$
1830 LOCATE 12,10
1840 PRINTDI$
1850 NEXTI
1860 PK=VAL(MID$(DI$,1,4))
1870 B$="&H"+MID$(DI$,5,2)
1880 RF=PK*256+VAL(B$)
1890 RETURN
1900 REM MENU PARA DIRECCIONES NO CORRECTAS
1910 CLS
1920 LOCATE 3,5
1930 PRINT"1: VOLVER A INTENTAR"
1940 LOCATE 3,7
1950 PRINT"2: AL MENU PRINCIPAL"
1960 A$=INKEY$:IF A$="" THEN 1960
1970 IF A$="1" THEN 550
1980 IF A$="2" THEN 160
1990 GOTO 1960
2000 REM PARA EPROM ESTROPEADA
2010 T=0:CLS
2020     T=T+1
2030 LOCATE 10,3
2040 IF T=30 THEN 2180
2050 PRINT "#####"
2060 PRINT "#####"
2070 PRINT "### EPROM ESTROPEADA ###"
2080 PRINT "#####"
2090 PRINT "#####"
2100 PRINT:PRINT:PRINT:LOCATE 10,3
2110 PRINT "#####"
2120 PRINT "#####"
2130 PRINT "### EPROM ESTROPEADA ###"
2140 PRINT "#####"
2150 PRINT "#####"
2160 SOUND 7,162:SOUND11,179
2170 GOTO 2020
2180 CLS:LOCATE 7,15
2190 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXX"
2200 PRINT"X X"
2210 PRINT"X 1: IR MENU PRINCIPAL X"
2220 PRINT"X X"
2230 PRINT"X 2:      EXIT      X"
2240 PRINT"X X"
2250 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXX"
2260 A$=INKEY$:IF A$="" THEN 2260
2270 IF A$="1" THEN 160
2280 IF A$="2" THEN 5790
2290 GOTO 2260
2300 REM INDICAR NUMERO DE BLOQUES

```

```

2310 IF MEMO=1 THEN NB=1
2320 IF MEMO=2 THEN NB=2
2330 IF MEMO=3 THEN NB=4
2340 IF MEMO=4 THEN NB=4
2350 IF MEMO=5 THEN NB=8
2360 IF MEMO=6 THEN NB=16
2370 RETURN
2380 REM MEDIR LONGITUD ENTRE DIRECCIONES
2390 DL=ZF-ZI
2400 RETURN
2410 REM COMPARAR LONGITUD CON MEMORIA
2420 IF ZM <=2048 AND MEMO=1 THEN RETURN
2430 IF ZM <=4096 AND MEMO=2 THEN RETURN
2440 IF ZM <=8192 AND MEMO=3 THEN RETURN
2450 IF ZM <=8192 AND MEMO=4 THEN RETURN
2460 IF ZM <=16384 AND MEMO=5 THEN RETURN
2470 IF ZM <=32768 AND MEMO=6 THEN RETURN
2480 REM DECIR DIRECCIONES MAL
2490 CLS
2500 LOCATE 0,7
2510 PRINT"LAS DIRECCIONES NO SE CORRESPONDEN"
2520 PRINT
2530 PRINT"CON EL TIPO DE MEMORIA"
2540 GOSUB 1170
2550 GOSUB 1900
2560 RETURN
2570 REM INTRODUCIR DATOS
2580 DI$="&H"
2590 CLS
2600 LOCATE 2,4
2610 PRINT"INTRODUCIR DATOS SEGUN DIRECCION"
2620 LOCATE 2,6
2630 PRINT" EN HEXADECIMAL"
2640 FOR I=0 TO 7
2650 FOR J=0 TO 255
2660 LOCATE 3,18
2670 PRINT"DIRECCION  &H";HEX$(ZI)
2680 LOCATE 25,18:PRINT"      "
2690 FOR K=0 TO 2
2700 DP$=INKEY$:IF DP$="" THEN 2700
2710 IF DP$=CHR$(32) THEN 2700
2720 IF K=2 AND DP$=CHR$(13) THEN 2800
2730 IF K=2 THEN 2700
2740 GOSUB 3000
2750 IF L=0 THEN 2700
2760 DI$=DI$+DP$
2770 LOCATE 25,18
2780 PRINT DI$
2790 NEXT K
2800 V1(I,J)=VAL(DI$)
2810 DI$="&H"
2820 IF ZI=ZF THEN RETURN
2830 ZI=ZI+1
2840 NEXTJ
2850 NEXTI
2860 RETURN

```

```

2870 REM SELECCION ALGORITMO DE GRABAR
2880 IF MEMD=1 THEN GOSUB 3150
2890 IF MEMD=2 THEN GOSUB3490
2900 IF MEMD=3 THEN GOSUB3830
2910 IF MEMD=4 THEN GOSUB3830
2920 IF MEMD=5 THEN GOSUB3830
2930 IF MEMD=6 THEN GOSUB4270
2940 RETURN
2950 REM Rutina DECIR ESTOY GRABANDO
2960 CLS
2970 LOCATE 5,10
2980 PRINT"ESTOY GRABANDO, ESPERA"
2990 RETURN
3000 REM COMPARAR DATOS HEXADECIMAL
3010 IF DP$="0"OR DP$="1"THEN L=1:RETURN
3020 IF DP$="2"OR DP$="3"THEN L=1:RETURN
3030 IF DP$="4"OR DP$="5"THEN L=1:RETURN
3040 IF DP$="6"OR DP$="7"THEN L=1:RETURN
3050 IF DP$="8"OR DP$="9"THEN L=1:RETURN
3060 IF DP$="A"OR DP$="a"THEN L=1:RETURN
3070 IF DP$="B"OR DP$="b"THEN L=1:RETURN
3080 IF DP$="C"OR DP$="c"THEN L=1:RETURN
3090 IF DP$="D"OR DP$="d"THEN L=1:RETURN
3100 IF DP$="E"OR DP$="e"THEN L=1:RETURN
3110 IF DP$="F"OR DP$="f"THEN L=1:RETURN
3120 L=0
3130 RETURN
3140 REM ***SUB. VERIFICACION EN MODD LECTURA ***
3150 REM GRABAR CDN 2716
3160 REM INICIALIZA CONTADOR Y 8255/1
3170 L1=0
3180 OUT 227,128
3190 OUT 226,6
3200 FOR I=0 TO 7
3210 FOR J=0 TO 255
3220 L1=L1+1
3230 OUT 226,38
3240 REM PRESENTA DIRECCION
3250 AH=INT(DT/256)
3260 AL=DT-AH*256
3270 OUT 224,AH
3280 OUT 225,AL
3290 REM PRESENTAR DATO
3300 OUT 211,128
3310 OUT 210,V1(I,J)
3320 REM ENVIA PULSO
3330 OUT 226,39
3340 FOR KT=0 TO 25:NEXT KT
3350 OUT 226,38
3360 OUT 226,6
3370 REM LECTURA DATO
3380 OUT 211,137
3390 RD=INP(210)
3400 REM VERIFICA DATO METIDO
3410 IF RD<>V1(I,J) AND L1<5 THEN 3220
3420 IF L1=5 THEN 2000
3430 L1=0

```

```

3440 DT=DT+1
3450 IF DT=RF THEN RETURN
3460 NEXT J
3470 NEXT I
3480 RETURN
3490 REM GRABAR CON 2732
3500 REM INICIALIZAR CONTADOR Y8255/1
3510 L1=0
3520 OUT 227,128
3530 OUT 226,1
3540 FOR I= 0 TO 7
3550 FOR J= 0 TO 255
3560 L1=L1+1
3570 OUT 226,7
3580 REM PRESENTAR DIRECCION
3590 AH=INT(DT/256)
3600 AL= DT-AH *256
3610 OUT 224,AH
3620 OUT 225,AL
3630 REM ENVIAR DATO
3640 OUT 211,128
3650 OUT 210,V1(I,J)
3660 REM LANZAR PULSO
3670 OUT 226,6
3680 FOR K=0 TO 25:NEXT K
3690 OUT 226,7
3700 OUT 266,1
3710 REM LECTURA DE DATOS
3720 OUT 211,137
3730 RD=INP(210)
3740 REM COMPRUEBA DATO LEIDO
3750 IF RD<>V1(I,J) AND L1<5 THEN 3560
3760 IF L1=5 THEN 2000
3770 L1=0
3780 DT=DT+1
3790 IF DT=RF THEN RETURN
3800 NEXT J
3810 NEXT I
3820 RETURN
3830 REM GRABAR CON 2764/27128
3840 REM INICIALIZACIONES
3850 OUT 227,128
3860 OUT 226,99
3870 OUT 226,103
3880 FOR I= 0 TO 7
3890 FOR J= 0 TO 255
3900 OUT 226,102
3910 OUT 226,116
3920 REM PRESENTAR DIRECCION
3930 AH=INT(DT/256)
3940 AL= DT-AH *256
3950 OUT 224,AH
3960 OUT 225,AL
3970 REM PRESENTAR DATO
3980 OUT 211,128
3990 OUT 210,V1(I,J)
4000 L1=0
4010 OUT 226,54:OUT 226,118
4020 L1=L1+1

```

```

4030 IF L1=15 THEN4100
4040 OUT226,86
4050 REM LECTURA DE DATOS
4060 OUT 211,137
4070 RD=INP(210)
4080 OUT 226,118
4090 IF RD<>V1(I,J) THEN 4010
4100 OUT226,54
4110 FORK=0 TO (L1*3):NEXTK
4120 OUT 226,118
4130 IF L1=15 THEN GOSUB 4190
4140 DT=DT+1
4150 NEXT J
4160 NEXT I
4170 OUT 226,99
4180 RETURN
4190 REM RUTINA VERIFICA BYTE
4200 OUT 226,86
4210 REM LECTURA DE DATOS
4220 OUT 211,137
4230 RD=INP(210)
4240 OUT 226,118
4250 IF RD<>V1(I,J) THEN 2000
4260 RETURN
4270 REM GRABAR CON 27256
4280 REM INICIALIZACIONES
4290 OUT 227,128
4300 OUT 226,35
4310 OUT 226,55
4320 FOR I= 0 TO 7
4330 FOR J= 0 TO 255
4340 REM PRESENTAR DIRECCION
4350 AH=INT(DT/256)
4360 AL= DT-AH *256
4370 OUT 224,AH
4380 OUT 225,AL
4390 REM PRESENTAR DATO
4400 OUT 211,128
4410 OUT 210,V1(I,J)
4420 L1=0
4430 OUT226,54:OUT 226,55
4440 L1=L1+1
4450 IF L1=15 THEN4520
4460 OUT226,23
4470 REM LECTURA DE DATOS
4480 OUT 211,137
4490 RD=INP(210)
4500 OUT 226,55
4510 IF RD<>V1(I,J) THEN 4440
4520 OUT226,54
4530 FORK=0 TO (L1*3):NEXTK
4540 OUT 226,55
4550 IF L1=15 THEN GOSUB 4510
4560 DT=DT+1
4570 NEXT J
4580 NEXT I
4590 OUT 226,36
4600 RETURN

```

```

4610 REM RUTINA VERIFICA BYTE/27256
4620 OUT226,23
4630 REM LECTURA DE DATOS
4640 OUT211,137
4650 RD=INP(210)
4660 OUT 226,55
4670 IF RD<>V1(I,J) THEN 2000
4680 RETURN
4690 REM PRESENTAR DIRECCIONES Y DATOS
4700 FOR I=0 TO 7
4710 FOR J=0 TO 255
4720 CLS
4730 FOR K=3 TO 18
4740 LOCATE 0,K
4750 PRINT"DIRECCION ";HEX$(RP);" DATO=&H";HEX$(V1(I,J)
)
4760 IF RP=RF THEN GOSUB 1170:GOTO4820
4770 RP=RP+1
4780 NEXTK
4790 GOSUB 1170
4800 NEXTJ
4810 NEXTI
4820 RETURN
4830 REM ASIGNACION ALGO. DE LECTURA
4840 IF MEMO=1 THEN GOSUB 4890
4850 IF MEMO=2 THEN GOSUB 5140
4860 IF MEMO=3OR MEMO=4 THEN GOSUB5380
4870 IF MEMO=5ORMEMO=6THEN GOSUB5380
4880 RETURN
4890 REM LEER MEMORIA 2716
4900 OUT 211,137
4910 OUT 227,128
4920 OUT226,35
4930 FORI=0TO7
4940 FORJ=0 TO255
4950 OUT 226,34
4960 AH=INT(DM/256)
4970 AL=DM-AH*256
4980 OUT 224,AH
4990 OUT225,AL
5000 REM LECTURA DE DATOS
5010 OUT 226,2
5020 RD=INP(210)
5030 V1(I,J)=RD
5040 OUT226,34
5050 OUT 226,35
5060 REM ULTIMA DIRECCION?
5070 IF DM=RF THEN RETURN
5080 DM=DM+1
5090 NEXTJ
5100 NEXTI
5110 RETURN
5120 REM LEER MEMORIA 2732
5130 REM LEER MEMORIA 2732
5140 OUT 227,128
5150 OUT 211,137
5160 OUT226,3
5170 FORI=0TO7

```

```

5180 FORJ=0 TO 255
5190 OUT226,2
5200 REM PRESENTAR DIRECCION
5210 AH=INT(DM/256)
5220 AL=DM-AH*256
5230 OUT 224,AH
5240 OUT225,AL
5250 REM HABILITA SALIDAS
5260 OUT226,0
5270 REM LECTURA DE DATOS
5280 RD=INP(210)
5290 V1(I,J)=RD
5300 OUT226,2
5310 OUT226,3
5320 REM ULTIMA DIRECCION
5330 IF DM=RF THEN RETURN
5340 DM=DM+1
5350 NEXT J
5360 NEXT I
5370 RETURN
5380 REM LEER EPROM 2764/27128
5390 OUT211,137
5400 OUT227,128
5410 OUT226,99
5420 FORI=0 TO7
5430 FORJ=0 TO 255
5440 OUT226,98
5450 REM PRESENTAR DIRECCION
5460 AH=INT(DM/256)
5470 AL=DM-AH*256
5480 OUT224,AH
5490 OUT225,AL
5500 REM HABILITA SALIDAS
5510 OUT226,66
5520 OUT 226,99
5530 REM ULTIMA DIRECCION
5540 IF DM=RF THEN RETURN
5550 DM=DM+1
5560 NEXTJ
5570 NEXTI
5580 RETURN
5590 REM SUBPROGRAMA PEDIR DIRECCIONES
5600 REM Y CALCULOS
5610 GOSUB1220
5620 LOCATE 0,1
5630 PRINT"MODELO DE EPROM? (Indique INDICE)"
5640 GOSUB1370
5650 REM DIRECCION INICIAL
5660 GOSUB1460
5670 REM DIRECCION FINAL
5680 GOSUB 1680
5690 REM MEDIR LONGITUD
5700 ZI=KI;ZF=RF
5710 GOSUB2380
5720 IF DL<=0 THEN 2480
5730 REM CORRESPONDENCIA MEM. Y LONG.
5740 ZM=DL

```

```

5750 GOSUB2410
5760 REM MIDE NUMERO DE BLOQUES
5770 GOSUB 2300
5780 RETURN
5790 REM ABANDONAR PROGRAMA
5800 CLS
5810 LOCATE3,7
5820 PRINT"VUELVO A BASIC"
5830 FOR I=0 TO 1000:NEXTI
5840 CLS
5850 KEY ON
5860 REM * * * OPCION COPIAR EPROM
5870 CLS
5880 LOCATE 0,3
5890 PRINT "ESTA OPCION LE PERMITE COPIAR"
5900 PRINT" PARA ELLO INSERTE LA EPROM"
5910 PRINT" A COPIAR EN EL ZOCALO"
5920 GOSUB 1170
5930 REM PEDIR DATOS
5940 GOSUB 5590
5950 UD=KI:CB=(DL/2048)+1
5960 DM=UD
5970 CLS
5980 LOCATE 4,4
5990 PRINT " ESTOY LEYENDO"
6000 REM LEER MEMORIA
6010 GOSUB 4830
6020 REM TRASBACE DE VARIABLES
6030 DT=UD
6040 GOSUB 4830
6050 UD=DM
6060 CLS
6070 LOCATE4,5
6080 PRINT"INTERCAMBIE EPROM"
6090 REM GRABAR
6100 REM DECIR ESTOY GRABANDO
6110 GOSUB 770
6120 GOSUB 4760
6130 CB=CB-1
6140 IF CB<0 THEN PRINT"COPIADA":RETURN
6150 CLS
6160 LOCATE 3,3
6170 PRINT"VUELVA A COLOCAR LA EPROM ORIGINAL"
6180 GOSUB 1170
6190 GOTO 6000
6200 END

```

LISTADO PROGRAMA MONITOR

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|-----|------|---|
| | | 1 | ;***** |
| | | 2 | ;PROGRAMA: PROGRAMADOR DE EPROMS. |
| | | 3 | ;by Jose Manuel Garcia martel |
| | | 4 | ;***** |
| | | 5 | |
| | | 6 | ;+++++ |
| | | 7 | ;PROGRAMA DIRECCIONES DE REGISTROS Y CONTADORES |
| | | 8 | ;EN RAM |
| | | 9 | ;+++++ |
| | | 10 | |
| FFB4 | | 11 | ORG 0FFB4H |
| | | 12 | MONRAM: |
| FFB4 | | 13 | FRET : DS 1 ; PARA FLAGS |
| FFB5 | | 14 | ARET : DS 1 ; PARA ACUMULADOR |
| FFB6 | | 15 | CRET : DS 1 ; PARA REGISTRO C |
| FFB7 | | 16 | BRET : DS 1 ; PARA REGISTRO B |
| FFB8 | | 17 | ERET : DS 1 ; PARA REGISTRO E |
| FFB9 | | 18 | DRET : DS 1 ; PARA REGISTRO D |
| FFBA | | 19 | LRET : DS 1 ; PARA REGISTRO L |
| FFBB | | 20 | HRET : DS 1 ; PARA REGISTRO H |
| FFBC | | 21 | PRET : DS 2 ; PARA PC |
| FFBE | | 22 | SRET : DS 2 ; PARA SP |
| FFC0 | | 23 | IRET : DS 1 ; ESTADO RIM |
| FFC1 | | 24 | DS 1 ; ESTADO SIM |
| | | 25 | |
| | | 26 | |
| | | 27 | ;+++++ |
| | | 28 | ;DIRECCIONES DEL PROGRAMA MONITOR |
| | | 29 | ;+++++ |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|-----|------|---------------------------------------|
| | | 30 | |
| FFC2 | | 31 | BEFREG: DS 8 |
| FFCA | | 32 | REGAD : DS 1 |
| FFCB | | 33 | MEMAD : DS 3 |
| FFCE | | 34 | AA : DS 2 |
| FFD0 | | 35 | EA : DS 2 |
| FFD2 | | 36 | BA : DS 2 |
| FFD4 | | 37 | H1 : DS 2 |
| FFD6 | | 38 | H2 : DS 2 |
| FFD8 | | 39 | AU : DS 2 |
| FFDA | | 40 | EU : DS 2 |
| FFDC | | 41 | DAT : DS 1 |
| FFDD | | 42 | ADR : DS 1 |
| FFDE | | 43 | MODUS : DS 1 |
| FFDF | | 44 | KOMZE : DS 1 |
| FFE0 | | 45 | RAMHZ : DS 1 |
| FFE1 | | 46 | DEFAULT: DS 1 |
| FFE2 | | 47 | DS 30 |
| | | 48 | STKINI: |
| | | 49 | |
| | | 50 | |
| 00EE | | 51 | DATEA EQU 0EEH; DATOS PARA 8279 |
| 00EF | | 52 | KOSEA EQU 0EFH; COMANDO/ESTATUS 8279 |
| 00FD | | 53 | TIMEH EQU 0FDH; TEMPORIZADOR 8155 MSB |
| 00FC | | 54 | TIMEL EQU 0FCH; " " LSB |
| 00F8 | | 55 | KOADR EQU 0F8H; COMANDOS/STATUS 8155 |
| | | 56 | |
| | | 57 | ;+++++ |
| | | 58 | ;DIRECCIONES DE LAS INTERRUPCIONES |
| | | 59 | ;+++++ |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|----------------------------------|
| | | 60 | ; RST1---->808 |
| | | 61 | ; RST2----> 810H |
| | | 62 | ; RST3----> 818H |
| | | 63 | ; RST4----> 820H TRAP---->824H |
| | | 64 | ; RST5----> 828H RST5.5---->832H |
| | | 65 | ; RST6----> 830H RST6.5---->834H |
| | | 66 | |
| | | 67 | |
| 0008 | | 68 | ORG 08H;DIRECCION PARA RST1 |
| 0008 | C30808 | 69 | JMP 808H; |
| 0010 | | 70 | ORG 10H; |
| 0010 | C31008 | 71 | JMP 810H; |
| 0018 | | 72 | ORG 18H; DIRECCION PARA RST3 |
| 0018 | C31808 | 73 | JMP 818H; |
| 0020 | | 74 | ORG 20H;DIRECCION PARA RST4 |
| 0020 | C32008 | 75 | JMP 820H; |
| 0024 | | 76 | ORG 24H;DIRECCION PARA TRAP |
| 0024 | C32408 | 77 | JMP 824H; |
| 0028 | | 78 | ORG 28H ;DIRECCION PARA RST5 |
| 0028 | C32808 | 79 | JMP 828H; |
| 002C | | 80 | ORG 2CH ;DIRECCIONES PARA RST5.5 |
| 002C | C32C08 | 81 | JMP 82CH ; |
| 0030 | | 82 | ORG 30H ;DIRECCIONES RST6 |
| 0030 | C33008 | 83 | JMP 830H; |
| 0034 | | 84 | ORG 34H;DIRECCION RST6.5 |
| 0034 | C33408 | 85 | JMP 834H; |
| | | 86 | |
| | | 87 | ;***** |
| | | 88 | ;ORIGEN. INICIALIZACION SISTEMA |
| | | 89 | ;***** |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|---|
| | | 90 | |
| | | 91 | NEXADR: |
| 0000 | | 92 | ORG 00H |
| 0000 | 3E40 | 93 | START: MVI A,01000000B |
| 0002 | F3 | 94 | DI |
| 0003 | 30 | 95 | SIM |
| 0004 | C33700 | 96 | JMP START1 |
| 0037 | | 97 | ORG NEXADR |
| 0037 | 3E0F | 98 | START1: MVI A,0FH |
| 0039 | 32DFFF | 99 | STA KOMZE |
| 003C | 20 | 100 | RIM |
| 003D | 32C0FF | 101 | STA IRET |
| 0040 | 97 | 102 | SUB A |
| 0041 | 32C1FF | 103 | STA IRET+1 |
| 0044 | 21CEFF | 104 | LXI H,AA |
| 0047 | 0616 | 105 | MVI B,16H |
| 0049 | 77 | 106 | LOES: MOV M,A |
| 004A | 23 | 107 | INX H |
| 004B | 05 | 108 | DCR B |
| 004C | C24900 | 109 | JNZ LOES |
| | | 110 | |
| | | 111 | ;***** |
| | | 112 | ;INICIALIZACION 8279 (CONTROLADOR DE TECLADO) |
| | | 113 | ;***** |
| 004F | 310000 | 114 | CLRALL: LXI SP,STKINI |
| 0052 | CDBC07 | 115 | CALL ANWAZL |
| 0055 | 97 | 116 | ANFANG: SUB A |
| 0056 | D3EF | 117 | OUT KOSEA |
| 0058 | 3EC2 | 118 | MVI A,0C2H |
| 005A | D3EF | 119 | OUT KOSEA |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|--------------------------------|
| | | 120 | |
| | | 121 | |
| 005C | 310000 | 122 | LXI SP,STKINI; |
| 005F | 1680 | 123 | MVI D,80H |
| 0061 | 1E7F | 124 | MVI E,7FH |
| 0063 | 3ADFFF | 125 | LDA KOMZE |
| 0066 | FE0C | 126 | CPI 0CH |
| 0068 | D27800 | 127 | JNC ANFA1 |
| 006B | 4F | 128 | MOV C,A |
| 006C | CDDE02 | 129 | CALL UMW |
| 006F | F690 | 130 | ORI 80H |
| 0071 | 42 | 131 | MOV B,D |
| 0072 | CDBA02 | 132 | CALL AUS |
| 0075 | CD8001 | 133 | CALL BEDIEN |
| 0078 | 3E88 | 134 | ANFA1: MVI A,88H |
| 007A | 42 | 135 | MOV B,D |
| 007B | CDBA02 | 136 | CALL AUS |
| 007E | CD8001 | 137 | CALL BEDIEN |
| | | 138 | |
| | | 139 | ;***** |
| | | 140 | ;SUBPROGRAMA PARA EL COMANDO 0 |
| | | 141 | ;***** |
| | | 142 | |
| 0081 | 1640 | 143 | CMD0: MVI D,40H |
| 0083 | 1E04 | 144 | CMD0A: MVI E,04H |
| 0085 | 2AC0FF | 145 | LHLD MEMAD+1 |
| 0088 | 7E | 146 | MOV A,M |
| 0089 | 32CBFF | 147 | STA MEMAD |
| 008C | CD1803 | 148 | CALL DISPLAY |
| 008F | 21CCFF | 149 | LXI H, MEMAD+1 |

147

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|--------------------------------|
| 0092 | 3E80 | 150 | MVI A,80H |
| 0094 | 32DEFF | 151 | STA MODUS |
| 0097 | CD8001 | 152 | CALL BEDIEN |
| 009A | 1602 | 153 | MVI D,02H |
| 009C | 2ACCFE | 154 | LHLD MEMAD+1 |
| 009F | 23 | 155 | INX H |
| 00A0 | FE12 | 156 | CPI 12H |
| 00A2 | C2A700 | 157 | JNZ M1 |
| 00A5 | 2B | 158 | DCX H |
| 00A6 | 2B | 159 | DCX H |
| 00A7 | 22CCFF | 160 | M1: SHLD MEMAD+1 |
| 00AA | C38300 | 161 | JMP CMD0A |
| | | 162 | |
| | | 163 | |
| | | 164 | ;***** |
| | | 165 | ;SUBPROGRAMA PARA EL COMANDO 1 |
| | | 166 | ;***** |
| | | 167 | |
| 00AD | CD5803 | 168 | CMD1: CALL AANF |
| 00B0 | CD6103 | 169 | CALL EANF |
| 00B3 | CD9103 | 170 | CALL DATANF |
| 00B6 | 3ADCFF | 171 | LDA DAT |
| 00B9 | 47 | 172 | MOV B,A |
| 00BA | 2AD0FF | 173 | LHLD EA |
| 00BD | EB | 174 | XCHG |
| 00BE | 2ACEFF | 175 | LHLD AA |
| 00C1 | 78 | 176 | FILL1: MOV A,B |
| 00C2 | CDFA02 | 177 | CALL SPEICH |
| 00C5 | C2C100 | 178 | JNZ FILL1 |
| 00C8 | CD5203 | 179 | CALL GLEICH |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|--------------------------------|
| 00CB | CA4F00 | 180 | JZ CLRALL |
| 00CE | 23 | 181 | INX H |
| 00CF | C3C100 | 182 | JMP FILL1 |
| | | 183 | |
| | | 184 | |
| | | 185 | |
| | | 186 | ;***** |
| | | 187 | ;SUBPROGRAMA PARA EL COMANDO 2 |
| | | 188 | ;***** |
| | | 189 | |
| 00D2 | CD5803 | 190 | CMD2: CALL AANF |
| 00D5 | CD6103 | 191 | CALL EANF |
| 00D8 | CD6A03 | 192 | CALL BANF |
| 00DB | 2AD2FF | 193 | LHLD BA |
| 00DE | E5 | 194 | PUSH H |
| 00DF | C1 | 195 | POP B |
| 00E0 | C5 | 196 | PUSH B |
| 00E1 | 2ACEFF | 197 | LHLD AA |
| 00E4 | E5 | 198 | PUSH H |
| 00E5 | CD4503 | 199 | CALL DSUB |
| 00E8 | C1 | 200 | POP B |
| 00E9 | D1 | 201 | POP D |
| 00EA | 2AD0FF | 202 | LHLD EA |
| 00ED | D2FA00 | 203 | JNC UMS1 |
| 00F0 | E5 | 204 | PUSH H |
| 00F1 | CD4503 | 205 | CALL DSUB |
| 00F4 | EB | 206 | XCHG |
| 00F5 | 19 | 207 | DAD D |
| 00F6 | C1 | 208 | POP B |
| 00F7 | C3FE00 | 209 | JMP UMS3 |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|--------------------------------|
| 00FA | CD4503 | 210 | UMS1: CALL DSUB |
| 00FD | EB | 211 | XCHG |
| 00FE | 0A | 212 | UMS3: LDAX B |
| 00FF | CDF002 | 213 | CALL SPEICH |
| 0102 | C2FE00 | 214 | JNZ UMS3 |
| 0105 | 7A | 215 | MOV A,D |
| 0106 | B3 | 216 | ORA E |
| 0107 | CA4F00 | 217 | JZ CLRALL |
| 010A | 1B | 218 | DCX D |
| 010B | E5 | 219 | PUSH H |
| 010C | CD4503 | 220 | CALL DSUB |
| 010F | E1 | 221 | POP H |
| 0110 | D21701 | 222 | JNC UMS4 |
| 0113 | 23 | 223 | INX H |
| 0114 | 23 | 224 | INX H |
| 0115 | 03 | 225 | INX B |
| 0116 | 03 | 226 | INX B |
| 0117 | 2B | 227 | UMS4: DCX H |
| 0118 | 0B | 228 | DCX B |
| 0119 | C3FE00 | 229 | JMP UMS3 |
| | | 230 | |
| | | 231 | |
| | | 232 | ***** |
| | | 233 | ;SUBPROGRAMA PARA EL COMANDO 3 |
| | | 234 | ***** |
| | | 235 | |
| 011C | CDFB03 | 236 | CMD3: CALL SDEF |
| 011F | CD5803 | 237 | CALL AANF |
| 0122 | CD6103 | 238 | CALL EANF |
| 0125 | CD6A03 | 239 | CALL BANF |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|-------------------|
| 0128 | CD0104 | 240 | CALL RDEF |
| 0128 | CD7303 | 241 | CALL AUANF |
| 012E | CD7C03 | 242 | CALL EUANF |
| 0131 | 2ADAFF | 243 | LHLD EU |
| 0134 | EB | 244 | XCHG |
| 0135 | 2AD8FF | 245 | LHLD AU |
| 0138 | 7E | 246 | UMR1: MOV A,M |
| 0139 | CD0604 | 247 | CALL K |
| 013C | CA4D01 | 248 | JZ UMR3 |
| 013F | CD5203 | 249 | UMR2: CALL GLEICH |
| 0142 | CA4F00 | 250 | JZ CLRALL |
| 0145 | 23 | 251 | INX H |
| 0146 | 05 | 252 | DCR B |
| 0147 | C23F01 | 253 | JNZ UMR2 |
| 014A | C33801 | 254 | JMP UMR1 |
| 014D | D5 | 255 | UMR3: PUSH D |
| 014E | E5 | 256 | PUSH H |
| 014F | 23 | 257 | INX H |
| 0150 | 4E | 258 | MOV C,M |
| 0151 | 23 | 259 | INX H |
| 0152 | 46 | 260 | MOV B,M |
| 0153 | 2AD0FF | 261 | LHLD EA |
| 0156 | EB | 262 | XCHG |
| 0157 | 2ACEFF | 263 | LHLD AA |
| 015A | 79 | 264 | MOV A,C |
| 015B | 95 | 265 | SUB L |
| 015C | 78 | 266 | MOV A,B |
| 015D | 9C | 267 | SBB H |
| 015E | DA7901 | 268 | JC UMREND |
| 0161 | 7B | 269 | MOV A,E |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|---------------------------------|
| 0162 | 91 | 270 | SUB C |
| 0163 | 7A | 271 | MOV A,D |
| 0164 | 98 | 272 | SBB B |
| 0165 | DA7901 | 273 | JC UMREND |
| 0168 | 79 | 274 | MOV A,C |
| 0169 | 95 | 275 | SUB L |
| 016A | 5F | 276 | MOV E,A |
| 016B | 78 | 277 | MOV A,B |
| 016C | 9C | 278 | SBB H |
| 016D | 57 | 279 | MOV D,A |
| 016E | 2AD2FF | 280 | LHLD BA |
| 0171 | 19 | 281 | DAD D |
| 0172 | EB | 282 | XCHG |
| 0173 | E1 | 283 | POP H |
| 0174 | E5 | 284 | PUSH H |
| 0175 | 23 | 285 | INX H |
| 0176 | 73 | 286 | MOV M,E |
| 0177 | 23 | 287 | INX H |
| 0178 | 72 | 288 | MOV M,D |
| 0179 | E1 | 289 | UMREND: POP H |
| 017A | D1 | 290 | POP D |
| 017B | 0603 | 291 | MVI B,3 |
| 017D | C33F01 | 292 | JMP UMR2 |
| | | 293 | |
| | | 294 | ; ++++++ |
| | | 295 | ; PROGRAMA: BEDIEN. |
| | | 296 | ; FUNCION: ENTRADA DE CURSORES. |
| | | 297 | ; ++++++ |
| | | 298 | |
| 0180 | 010040 | 299 | BEDIEN: LXI B,4000H |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|------------------|
| 0183 | DBEF | 300 | TAST: IN KOSEA |
| 0185 | E607 | 301 | ANI 07H |
| 0187 | C29501 | 302 | JNZ CLOE |
| 018A | 0D | 303 | DCR C |
| 018B | C28301 | 304 | JNZ TAST |
| 018E | 05 | 305 | DCR B |
| 018F | C28301 | 306 | JNZ TAST |
| 0192 | C39B02 | 307 | JMP BLIN |
| 0195 | 42 | 308 | CLOE: MOV B,D |
| 0196 | CDCF02 | 309 | CALL KOGEN |
| 0199 | DBEE | 310 | IN DATEA |
| 019B | E67F | 311 | ANI 7FH |
| 019D | C2A701 | 312 | JNZ LES |
| 01A0 | 3AE0FF | 313 | LDA RAMHZ |
| 01A3 | 42 | 314 | MOV B,D |
| 01A4 | CDBA02 | 315 | CALL AUS |
| 01A7 | 3E40 | 316 | LES: MVI A,40H |
| 01A9 | D3EF | 317 | OUT KOSEA |
| 01AB | DBEE | 318 | IN DATEA |
| 01AD | E63F | 319 | ANI 3FH |
| 01AF | FE10 | 320 | CPI 10H |
| 01B1 | DAF101 | 321 | JC HEXZEI |
| 01B4 | CA8A02 | 322 | JZ LEFT |
| 01B7 | FE11 | 323 | CPI 11H |
| 01B9 | CA9402 | 324 | JZ RIGHT |
| 01BC | 4F | 325 | MOV C,A |
| 01BD | 7A | 326 | MOV A,D |
| 01BE | 07 | 327 | RLC |
| 01BF | 79 | 328 | MOV A,C |
| 01C0 | D0 | 329 | RNC |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|------------------|
| 01C1 | CD0104 | 330 | CALL RDEF |
| 01C4 | 3ADFFF | 331 | LDA KOMZE |
| 01C7 | FE0C | 332 | CPI 0CH |
| 01C9 | D28001 | 333 | JNC BEDIEN |
| 01CC | FE09 | 334 | CPI 09H |
| 01CE | C2D901 | 335 | JNZ TABZ |
| 01D1 | 7B | 336 | MOV A,E |
| 01D2 | FE7F | 337 | CPI 7FH |
| 01D4 | CA4F00 | 338 | JZ CLRALL |
| 01D7 | 3E09 | 339 | MVI A,09H |
| 01D9 | D5 | 340 | TABZ: PUSH D |
| 01DA | 11E901 | 341 | LXI D,BTBL |
| 01DD | 6F | 342 | MOV L,A |
| 01DE | 2600 | 343 | MVI H,0 |
| 01E0 | 29 | 344 | DAD H |
| 01E1 | 19 | 345 | DAD D |
| 01E2 | 5E | 346 | MOV E,M |
| 01E3 | 23 | 347 | INX H |
| 01E4 | 56 | 348 | MOV D,M |
| 01E5 | EB | 349 | XCHG |
| 01E6 | D1 | 350 | POP D |
| 01E7 | E3 | 351 | XTHL |
| 01E8 | C9 | 352 | RET |
| 01E9 | 8100 | 353 | BTBL: DW CMD0 |
| 01EB | AD00 | 354 | DW CMD1 |
| 01ED | D200 | 355 | DW CMD2 |
| 01EF | 1C01 | 356 | DW CMD3 |
| 01F1 | 4F | 357 | HEXZEI: MOV C,A |
| 01F2 | 7A | 358 | MOV A,D |
| 01F3 | 07 | 359 | RLC |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|------------------|
| 01F4 | 79 | 360 | MOV A,C |
| 01F5 | D21902 | 361 | JNC ECHO |
| 01F8 | FE0C | 362 | CPI 0CH |
| 01FA | D28001 | 363 | JNC BEDIEN |
| 01FD | 32DFFF | 364 | STA KOMZE |
| 0200 | 1E7F | 365 | MVI E,7FH |
| 0202 | 97 | 366 | SUB A |
| 0203 | 0640 | 367 | MVI B,40H |
| 0205 | CDBA02 | 368 | CALL AUS |
| 0208 | 3ED2 | 369 | MVI A,0D2H |
| 020A | D3EF | 370 | OUT KOSEA |
| 020C | 79 | 371 | MOV A,C |
| 020D | CDDB02 | 372 | CALL UMW |
| 0210 | F680 | 373 | ORI 80H |
| 0212 | 42 | 374 | MOV B,D |
| 0213 | CDBA02 | 375 | CALL AUS |
| 0216 | C38001 | 376 | JMP BEDIEN |
| | | 377 | |
| 0219 | CDDB02 | 378 | ECHO: CALL UMW |
| 021C | 42 | 379 | MOV B,D |
| 021D | CDBA02 | 380 | CALL AUS |
| 0220 | E5 | 381 | ABSP: PUSH H |
| 0221 | 23 | 382 | INX H |
| 0222 | 23 | 383 | INX H |
| 0223 | 06C0 | 384 | MVI B,0C0H |
| 0225 | 78 | 385 | PACK: MOV A,B |
| 0226 | A3 | 386 | ANA E |
| 0227 | C23702 | 387 | JNZ PACK2 |
| 022A | 78 | 388 | MOV A,B |
| 022B | 17 | 389 | RAL |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|------------------|
| 022C | 3F | 390 | CMC |
| 022D | DA3102 | 391 | JC PACK1 |
| 0230 | 2B | 392 | DCX H |
| 0231 | 1F | 393 | PACK1: RAR |
| 0232 | 47 | 394 | MOV B, A |
| 0233 | A2 | 395 | ANA D |
| 0234 | C24502 | 396 | JNZ PACK4 |
| 0237 | 78 | 397 | PACK2: MOV A, B |
| 0238 | 17 | 398 | RAL |
| 0239 | 78 | 399 | MOV A, B |
| 023A | D24002 | 400 | JNC PACK3 |
| 023D | E67F | 401 | ANI 7FH |
| 023F | 37 | 402 | STC |
| 0240 | 1F | 403 | PACK3: RAR |
| 0241 | 47 | 404 | MOV B, A |
| 0242 | C32502 | 405 | JMP PACK |
| 0245 | 78 | 406 | PACK4: MOV A, B |
| 0246 | 17 | 407 | RAL |
| 0247 | 3EF0 | 408 | MVI A, 0F0H |
| 0249 | DA5402 | 409 | JC PACK5 |
| 024C | 79 | 410 | MOV A, C |
| 024D | 87 | 411 | ADD A |
| 024E | 87 | 412 | ADD A |
| 024F | 87 | 413 | ADD A |
| 0250 | 87 | 414 | ADD A |
| 0251 | 4F | 415 | MOV C, A |
| 0252 | 3E0F | 416 | MVI A, 0FH |
| 0254 | A6 | 417 | PACK5: ANA M |
| 0255 | B1 | 418 | ORA C |
| 0256 | 77 | 419 | MOV M, A |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|---------------------|
| 0257 | E1 | 420 | POP H |
| 0258 | 3ADEFB | 421 | LDA MODUS |
| 025B | FE80 | 422 | CPI 80H |
| 025D | C27C02 | 423 | JNZ CURE |
| 0260 | 7A | 424 | MOV A,D |
| 0261 | E603 | 425 | ANI 03H |
| 0263 | CA6F02 | 426 | JZ LESBE |
| 0266 | 2ACFF | 427 | LHLD MEMAD+1 |
| 0269 | 3ACBFF | 428 | LDA MEMAD |
| 026C | CDFA02 | 429 | CALL SPEICH |
| 026F | 2ACFF | 430 | LESBE: LHLD MEMAD+1 |
| 0272 | 7E | 431 | MOV A,M |
| 0273 | 32CBFF | 432 | STA MEMAD |
| 0276 | CD1803 | 433 | CALL DISPLAY |
| 0279 | 21CCFF | 434 | LXI H, MEMAD+1 |
| 027C | 7A | 435 | CURE: MOV A,D |
| 027D | 0F | 436 | RECHTS: RRC |
| 027E | 47 | 437 | MOV B,A |
| 027F | DAB001 | 438 | JC BEDIEN |
| 0282 | A3 | 439 | ANA E |
| 0283 | C27D02 | 440 | JNZ RECHTS |
| 0286 | 50 | 441 | MOV D,B |
| 0287 | C39B02 | 442 | JMP BLIN |
| 028A | 7A | 443 | LEFT: MOV A,D |
| 028B | 07 | 444 | LEFT1: RLC |
| 028C | 57 | 445 | MOV D,A |
| 028D | A3 | 446 | ANA E |
| 028E | C28B02 | 447 | JNZ LEFT1 |
| 0291 | C39B02 | 448 | JMP BLIN |
| 0294 | 7A | 449 | RIGHT: MOV A,D |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|------------------|
| 0295 | 0F | 450 | RIGH1: RRC |
| 0296 | 57 | 451 | MOV D,A |
| 0297 | A3 | 452 | ANA E |
| 0298 | C29502 | 453 | JNZ RIGH1 |
| 029B | 42 | 454 | BLIN: MOV B,D |
| 029C | CDCF02 | 455 | CALL KOGEN |
| 029F | DBEE | 456 | IN DATEA |
| 02A1 | 47 | 457 | MOV B,A |
| 02A2 | E67F | 458 | ANI 7FH |
| 02A4 | CAB002 | 459 | JZ BL1 |
| 02A7 | 78 | 460 | MOV A,B |
| 02A8 | 32E0FF | 461 | STA RAMHZ |
| 02AB | E680 | 462 | ANI 80H |
| 02AD | C3B302 | 463 | JMP BL2 |
| 02B0 | 3AE0FF | 464 | BL1: LDA RAMHZ |
| 02B3 | 42 | 465 | BL2: MOV B,D |
| 02B4 | CDBA02 | 466 | CALL AUS |
| 02B7 | C38001 | 467 | JMP BEDIEN |
| | | 468 | ;+++++ |
| | | 469 | ;PROGRAMA AUS |
| | | 470 | ;+++++ |
| 02BA | C5 | 471 | AUS: PUSH B |
| 02BB | 4F | 472 | MOV C,A |
| 02BC | DBEF | 473 | AUS1: IN KOSEA |
| 02BE | C680 | 474 | ADI 80H |
| 02C0 | DABC02 | 475 | JC AUS1 |
| 02C3 | CDCF02 | 476 | CALL KOGEN |
| 02C6 | 79 | 477 | MOV A,C |
| 02C7 | D3EE | 478 | OUT DATEA |
| 02C9 | C1 | 479 | POP B |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|--|
| 02CA | 78 | 480 | MOV A, B |
| 02CB | 0F | 481 | RRC |
| 02CC | 47 | 482 | MOV B, A |
| 02CD | 97 | 483 | SUB A |
| 02CE | C9 | 484 | RET |
| | | 485 | ;***** |
| | | 486 | ;PROGRAMA KOGEN |
| | | 487 | ;***** |
| 02CF | 78 | 488 | KOGEN: MOV A, B |
| 02D0 | 065F | 489 | MVI B, 5FH |
| 02D2 | 07 | 490 | KOG2: RLC |
| 02D3 | 04 | 491 | INR B |
| 02D4 | D2D202 | 492 | JNC KOG2 |
| 02D7 | 78 | 493 | MOV A, B |
| 02D8 | D3EF | 494 | OUT KOSEA |
| 02DA | C9 | 495 | RET |
| | | 496 | ;***** |
| | | 497 | ;PROGRAMA UMW |
| | | 498 | ;FUNCION: CONVERSION DEL CODIGO DE TECLA |
| | | 499 | ; A SIMBOLO PARA DISPLAY |
| | | 500 | ;***** |
| 02DB | E5 | 501 | UMW: PUSH H |
| 02DC | E60F | 502 | ANI 0FH |
| 02DE | 21EA02 | 503 | LXI H, HEX0 |
| 02E1 | 85 | 504 | ADD L |
| 02E2 | 6F | 505 | MOV L, A |
| 02E3 | 3E00 | 506 | MVI A, 00H |
| 02E5 | 8C | 507 | ADC H |
| 02E6 | 67 | 508 | MOV H, A |
| 02E7 | 7E | 509 | MOV A, M |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|----------------------------------|
| 02E8 | E1 | 510 | POP H |
| 02E9 | C9 | 511 | RET |
| | | 512 | ;***** |
| | | 513 | ;TABLA PARA CODIGO HEXADECIMAL. |
| | | 514 | ;***** |
| 02EA | 37 | 515 | HEX0: DB 37H ;7 SEGMENTOS PARA 0 |
| 02EB | 06 | 516 | DB 06H ;7 SEGMENTOS PARA 1 |
| 02EC | 5B | 517 | DB 5BH ;7 SEGMENTOS PARA 2 |
| 02ED | 4F | 518 | DB 4FH ;7 SEGMENTOS PARA 3 |
| 02EE | 66 | 519 | DB 66H ;7 SEGMENTOS PARA 4 |
| 02EF | 6D | 520 | DB 6DH ;7 SEGMENTOS PARA 5 |
| 02F0 | 7D | 521 | DB 7DH ;7 SEGMENTOS PARA 6 |
| 02F1 | 07 | 522 | DB 07H ;7 SEGMENTOS PARA 7 |
| 02F2 | 7F | 523 | DB 7FH ;7 SEGMENTOS PARA 8 |
| 02F3 | 6F | 524 | DB 6FH ;7 SEGMENTOS PARA 9 |
| 02F4 | 77 | 525 | DB 77H ;7 SEGMENTOS PARA A |
| 02F5 | 7C | 526 | DB 7CH ;7 SEGMENTOS PARA B |
| 02F6 | 39 | 527 | DB 39H ;7 SEGMENTOS PARA C |
| 02F7 | 5E | 528 | DB 5EH ;7 SEGMENTOS PARA D |
| 02F8 | 79 | 529 | DB 79H ;7 SEGMENTOS PARA E |
| 02F9 | 71 | 530 | DB 71H ;7 SEGMENTOS PARA F |
| | | 531 | ;***** |
| | | 532 | ;PROGRAMA SPEICH |
| | | 533 | ;***** |
| | | 534 | |
| 02FA | F5 | 535 | SPEICH: PUSH PSW |
| 02FB | 7C | 536 | MOV A,H |
| 02FC | FE10 | 537 | CPI 10H |
| 02FE | D20D03 | 538 | JNC SPEI1 |
| 0301 | 3E65 | 539 | MVI A,65H |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|-----------------------|
| 0303 | D3FD | 540 | OUT TIMEH |
| 0305 | 3E80 | 541 | MVI A,80H |
| 0307 | D3FC | 542 | OUT TIMEL |
| 0309 | 3EC0 | 543 | MVI A,0C0H |
| 030B | D3F8 | 544 | OUT KOADR |
| 030D | F1 | 545 | SPEI1: POP PSW |
| 030E | F5 | 546 | PUSH PSW |
| 030F | 77 | 547 | MOV M,A |
| 0310 | 7E | 548 | MOV A,M |
| 0311 | CD1803 | 549 | CALL DISPLA |
| 0314 | E3 | 550 | XTHL |
| 0315 | BC | 551 | CMP H |
| 0316 | E1 | 552 | POP H |
| 0317 | C9 | 553 | RET |
| | | 554 | ***** |
| | | 555 | ; PROGRAMA DISPLA |
| | | 556 | ***** |
| 0318 | F5 | 557 | DISPLA: PUSH PSW |
| 0319 | C5 | 558 | PUSH B |
| 031A | 4F | 559 | MOV C,A |
| 031B | 0640 | 560 | MVI B,40H |
| 031D | 7C | 561 | MOV A,H |
| 031E | CD3203 | 562 | CALL BYAUS |
| 0321 | 7D | 563 | MOV A,L |
| 0322 | CD3203 | 564 | CALL BYAUS |
| 0325 | 3E48 | 565 | MVI A,48H |
| 0327 | CDBA02 | 566 | CALL AUS ;TRAER "=" |
| 032A | 79 | 567 | MOV A,C |
| 032B | CD3203 | 568 | CALL BYAUS ;DATENBYTE |
| 032E | C1 | 569 | POP B |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|------------------------|
| 032F | F1 | 570 | POP PSW |
| 0330 | C9 | 571 | RET |
| | | 572 | |
| | | 573 | ; ++++++ |
| | | 574 | ; PROGRAMA BYTE, BYAUS |
| | | 575 | ; ++++++ |
| 0331 | 7E | 576 | BYTE: MOV A, M |
| 0332 | F5 | 577 | BYAUS: PUSH PSW |
| 0333 | 0F | 578 | RRC |
| 0334 | 0F | 579 | RRC |
| 0335 | 0F | 580 | RRC |
| 0336 | 0F | 581 | RRC |
| 0337 | CDDB02 | 582 | CALL UMW |
| 033A | CDBA02 | 583 | CALL AUS |
| 033D | F1 | 584 | POP PSW |
| 033E | CDDB02 | 585 | CALL UMW |
| 0341 | CDBA02 | 586 | CALL AUS |
| 0344 | C9 | 587 | RET |
| | | 588 | |
| | | 589 | ; ++++++ |
| | | 590 | ; PROGRAMA DSUB |
| | | 591 | ; ++++++↑ |
| 0345 | C5 | 592 | DSUB: PUSH B |
| 0346 | 37 | 593 | STC |
| 0347 | 79 | 594 | MOV A, C |
| 0348 | 2F | 595 | CMA |
| 0349 | 8D | 596 | ADC L |
| 034A | 6F | 597 | MOV L, A |
| 034B | 78 | 598 | MOV A, B |
| 034C | 2F | 599 | CMA |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|--|
| 034D | 8C | 600 | ADC H |
| 034E | 67 | 601 | MOV H,A |
| 034F | 3F | 602 | CMC |
| 0350 | C1 | 603 | POP B |
| 0351 | C9 | 604 | RET |
| | | 605 | ;***** |
| | | 606 | ; PROGRAMA GLEICH |
| | | 607 | ;***** |
| 0352 | 7A | 608 | GLEICH: MOV A,D |
| 0353 | BC | 609 | CMP H |
| 0354 | C0 | 610 | RNZ |
| 0355 | 7B | 611 | MOV A,E |
| 0356 | BD | 612 | CMP L |
| 0357 | C9 | 613 | RET |
| | | 614 | ;***** |
| | | 615 | ; PROGRAMAS: AANF,EANF,BANF,AUANF,EUANF,DATANF |
| | | 616 | ;***** |
| 0358 | 21CEFF | 617 | AANF: LXI H,AA |
| 035B | 017777 | 618 | LXI B,7777H |
| 035E | C3C603 | 619 | JMP PARAM |
| 0361 | 21D0FF | 620 | EANF: LXI H,EA |
| 0364 | 017779 | 621 | LXI B,7977H |
| 0367 | C3C603 | 622 | JMP PARAM |
| 036A | 21D2FF | 623 | BANF: LXI H,BA |
| 036D | 01777C | 624 | LXI B,7C77H |
| 0370 | C3C603 | 625 | JMP PARAM |
| 0373 | 21D8FF | 626 | AUANF: LXI H,AU |
| 0376 | 013E77 | 627 | LXI B,773EH |
| 0379 | C3C603 | 628 | JMP PARAM |
| 037C | 21DAFF | 629 | EUANF: LXI H,EU |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|---------------------|
| 037F | 013E79 | 630 | LXI B,793EH |
| 0382 | C3C603 | 631 | JMP PARAM |
| 0385 | 21DCFF | 632 | ADRANF: LXI H,ADR-1 |
| 0388 | 010000 | 633 | LXI B,0000H |
| 038B | 117773 | 634 | LXI D,7377H |
| 038E | C39A03 | 635 | JMP PAAN |
| 0391 | 21DBFF | 636 | DATANF: LXI H,DAT-1 |
| 0394 | 010000 | 637 | LXI B,0000H |
| 0397 | 117C5E | 638 | LXI D,5E7CH |
| | | 639 | ;+++++ |
| 039A | 78 | 640 | PAAN: MOV A,B |
| 039B | 0640 | 641 | MVI B,40H |
| 039D | CDBA02 | 642 | CALL AUS |
| 03A0 | 79 | 643 | MOV A,C |
| 03A1 | CDBA02 | 644 | CALL AUS |
| 03A4 | 7A | 645 | MOV A,D |
| 03A5 | CDBA02 | 646 | CALL AUS |
| 03A8 | 7B | 647 | MOV A,E |
| 03A9 | CDBA02 | 648 | CALL AUS |
| 03AC | 3E48 | 649 | MVI A,48H ; "=" |
| 03AE | CDBA02 | 650 | CALL AUS |
| 03B1 | 3AE1FF | 651 | LDA DEFAULT |
| 03B4 | 07 | 652 | RLC |
| 03B5 | DABB03 | 653 | JC PAANF1 |
| 03B8 | 23 | 654 | INX H |
| 03B9 | 3600 | 655 | MVI M,0 |
| 03BB | CD3103 | 656 | PAANF1: CALL BYTE |
| 03BE | 2B | 657 | DCX H |
| 03BF | 1E7C | 658 | MVI E,7CH |
| 03C1 | 1602 | 659 | MVI D,02H |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|-------------------|
| 03C3 | C3EF03 | 660 | JMP PARAM3 |
| | | 661 | ;+++++ |
| | | 662 | ;+++++ |
| | | 663 | ;+++++ |
| 03C6 | 78 | 664 | PARAM: MOV A,B |
| 03C7 | 0640 | 665 | MVI B,40H |
| 03C9 | CDBA02 | 666 | CALL AUS |
| 03CC | 79 | 667 | MOV A,C |
| 03CD | CDBA02 | 668 | CALL AUS |
| 03D0 | 3E48 | 669 | MVI A,48H |
| 03D2 | CDBA02 | 670 | CALL AUS |
| 03D5 | 3AE1FF | 671 | LDA DEFAULT |
| 03D8 | 07 | 672 | RLC |
| 03D9 | DAE103 | 673 | JC PARAM2 |
| 03DC | 97 | 674 | SUB A |
| 03DD | 77 | 675 | MOV M,A |
| 03DE | 23 | 676 | INX H |
| 03DF | 77 | 677 | MOV M,A |
| 03E0 | 2B | 678 | DCX H |
| 03E1 | 23 | 679 | PARAM2: INX H |
| 03E2 | 0608 | 680 | MVI B,08H |
| 03E4 | CD3103 | 681 | CALL BYTE |
| 03E7 | 2B | 682 | DCX H |
| 03E8 | CD3103 | 683 | CALL BYTE |
| 03EB | 1E70 | 684 | MVI E,70H |
| 03ED | 1608 | 685 | MVI D,08H |
| 03EF | 3ADEFF | 686 | PARAM3: LDA MODUS |
| 03F2 | F610 | 687 | ORI 10H |
| 03F4 | 32DEFF | 688 | STA MODUS |
| 03F7 | CD8001 | 689 | CALL BEDIEN |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|-------------------------|
| 03FA | C9 | 690 | RET |
| | | 691 | |
| | | 692 | ;***** |
| | | 693 | ; PROGRAMAMA SDEF,RDEF. |
| | | 694 | ;***** |
| 03FB | 3EFF | 695 | SDEF: MVI A,0FFH |
| 03FD | 32E1FF | 696 | STA DEFAULT |
| 0400 | C9 | 697 | RET |
| 0401 | 97 | 698 | RDEF: SUB A |
| 0402 | 32E1FF | 699 | STA DEFAULT |
| 0405 | C9 | 700 | RET |
| | | 701 | ;***** |
| | | 702 | ; PROGRAMAMA K |
| | | 703 | ;***** |
| 0406 | 4F | 704 | K: MOV C,A |
| 0407 | 0603 | 705 | MVI B:3 |
| 0409 | FEC3 | 706 | CPI 0C3H |
| 040B | C8 | 707 | RZ |
| 040C | FECD | 708 | CPI 0CDH |
| 040E | C8 | 709 | RZ |
| 040F | E6E7 | 710 | ANI 0E7H |
| 0411 | FE22 | 711 | CPI 22H |
| 0413 | C8 | 712 | RZ |
| 0414 | E6C7 | 713 | ANI 0C7H |
| 0416 | FEC2 | 714 | CPI 0C2H |
| 0418 | C8 | 715 | RZ |
| 0419 | FEC4 | 716 | CPI 0C4H |
| 041B | C8 | 717 | RZ |
| 041C | 79 | 718 | MOV A,C |
| 041D | E6CF | 719 | ANI 0CFH |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|------------------|
| 041F | FE01 | 720 | CPI 01H |
| 0421 | CA3D04 | 721 | JZ KEND |
| 0424 | 0602 | 722 | MVI B, 2 |
| 0426 | 79 | 723 | MOV A, C |
| 0427 | E6F7 | 724 | ANI 0F7H |
| 0429 | FED3 | 725 | CPI 0D3H |
| 042B | CA3D04 | 726 | JZ KEND |
| 042E | E6C7 | 727 | ANI 0C7H |
| 0430 | FE06 | 728 | CPI 06H |
| 0432 | CA3D04 | 729 | JZ KEND |
| 0435 | FEC6 | 730 | CPI 0C6H |
| 0437 | CA3D04 | 731 | JZ KEND |
| 043A | 0601 | 732 | K1: MVI B, 1 |
| 043C | 97 | 733 | SUB A |
| 043D | B8 | 734 | KEND: CMP B |
| 043E | C9 | 735 | RET |
| | | 736 | ;+++++ |
| | | 737 | ; PROGRAMA JUMP |
| | | 738 | ;+++++ |
| 043F | 4F | 739 | JUMP: MOV C, A |
| 0440 | E630 | 740 | ANI 30H |
| 0442 | 0640 | 741 | MVI B, 40H |
| 0444 | CA5504 | 742 | JZ JUMP1 |
| 0447 | 0680 | 743 | MVI B, 80H |
| 0449 | EA5504 | 744 | JPE JUMP1 |
| 044C | FE20 | 745 | CPI 20H |
| 044E | 0604 | 746 | MVI B, 04H |
| 0450 | CA5504 | 747 | JZ JUMP1 |
| | | 748 | |
| 0453 | 0601 | 749 | MVI B, 01H |

167

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|---|
| 0455 | 3AB4FF | 750 | JUMP1: LDA FRET |
| 0458 | A0 | 751 | ANA B |
| 0459 | 47 | 752 | MOV B,A |
| 045A | 79 | 753 | MOV A,C |
| 045B | E608 | 754 | ANI 09H |
| 045D | 80 | 755 | ADD B |
| 045E | 37 | 756 | STC |
| 045F | E8 | 757 | RPE |
| 0460 | 3F | 758 | CMC |
| 0461 | C9 | 759 | RET |
| | | 760 | ;***** |
| | | 761 | ;PROGRAMA PARA EL TECLADO |
| | | 762 | ;FUNCION BORRADO DISPLAY |
| | | 763 | ;***** |
| | | 764 | |
| 07BC | | 765 | ORG 7BCH |
| | | 766 | |
| | | 767 | ;***** |
| | | 768 | ;PROGRAMA ANWAZL |
| | | 769 | ;FUNCION:INICIALIZACION DISPLAY |
| | | 770 | ;***** |
| 07BC | 97 | 771 | ANWAZL: SUB A |
| 07BD | 0608 | 772 | MVI B,B |
| 07BF | 05 | 773 | SLAZL: DCR B |
| 07C0 | CDC707 | 774 | CALL ANWAUS |
| 07C3 | C2BF07 | 775 | JNZ SLAZL |
| 07C6 | C9 | 776 | RET |
| | | 777 | ;***** |
| | | 778 | ;PROGRAMA ANWAUS |
| | | 779 | ;FUNCION:VISUALIZACION DEL SIGNO DESCRITO |

| LOC | OBJ | LINE | SOURCE STATEMENT |
|------|--------|------|--------------------------|
| | | 780 | ; POR LA PALABRA DISPLAY |
| | | 781 | ; ++++++ |
| 07C7 | F5 | 782 | ANWAUS: PUSH PSW |
| 07C8 | 3E80 | 783 | MVI A,80H |
| 07CA | 80 | 784 | ADD B |
| 07CB | D3EF | 785 | OUT 0EFH |
| 07CD | F1 | 786 | POP PSW |
| 07CE | D3EE | 787 | OUT 0EEH |
| 07D0 | C9 | 788 | RET |
| | | 789 | |
| | | 790 | ; ++++++ |
| | | 791 | ; PROGRAMA ANWEIN |
| | | 792 | ; ++++++ |
| 07D1 | 3EC2 | 793 | ANWEIN: MVI A,0C2H |
| 07D3 | D3EF | 794 | OUT 0EFH |
| 07D5 | 3E40 | 795 | MVI A,40H |
| 07D7 | D3EF | 796 | OUT 0EFH |
| 07D9 | DBEF | 797 | WAREIN: IN 0EFH |
| 07DB | E60F | 798 | ANI 0FH |
| 07DD | CAD907 | 799 | JZ WAREIN |
| 07E0 | DBEE | 800 | IN 0EEH |
| 07E2 | C9 | 801 | RET |
| | | 802 | |
| | | 803 | END |

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0

MODULE PAGE 28

USER SYMBOLS

| | | | | | | | | | | | | | |
|--------|--------|--------|--------|--------|--------|---------|--------|--------|--------|--------|--------|--------|--------|
| AA | A FFCE | AANF | A 0358 | ABSP | A 0220 | ADR | A FFDD | ADRANF | A 0385 | ANFA1 | A 0078 | ANFANG | A 0055 |
| ANWAUS | A 07C7 | ANWAZL | A 07BC | ANWEIN | A 07D1 | ARET | A FFB5 | AU | A FFD8 | AUANF | A 0373 | AUS | A 028A |
| AUS1 | A 02BC | BA | A FFD2 | BANF | A 036A | BEDIEN | A 0180 | BEFREG | A FFC2 | BL1 | A 02B0 | BL2 | A 02B3 |
| BLIN | A 029B | BRET | A FFB7 | BTBL | A 01E9 | BYAUS | A 0332 | BYTE | A 0331 | CLOE | A 0195 | CLRALL | A 004F |
| CMD0 | A 0081 | CMD0A | A 0083 | CMD1 | A 00AD | CMD2 | A 00D2 | CMD3 | A 011C | CRET | A FFB6 | CURE | A 027C |
| DAT | A FFDC | DATANF | A 0391 | DATEA | A 00EE | DEFAULT | A FFE1 | DISPLA | A 0318 | DRET | A FFB9 | DSUB | A 0345 |
| EA | A FFD0 | EANF | A 0361 | ECHO | A 0219 | ERET | A FFB8 | EU | A FFDA | EUANF | A 037C | FILL1 | A 00C1 |
| FRET | A FFB4 | GLEICH | A 0352 | H1 | A FFD4 | H2 | A FFD6 | HEX0 | A 02EA | HEXZEI | A 01F1 | HRET | A FFB8 |
| | | | | | | | | | | | | | |
| IRET | A FFC0 | JUMP | A 043F | JUMP1 | A 0455 | K | A 0406 | K1 | A 043A | KEND | A 043D | KOADR | A 00F8 |
| KOG2 | A 02D2 | KOGEN | A 02CF | KOMZE | A FFDF | KOSEA | A 00EF | LEFT | A 028A | LEFT1 | A 028B | LES | A 01A7 |
| LESBE | A 026F | LOES | A 0049 | LRET | A FFBA | M1 | A 00A7 | MEMAD | A FFCB | MODUS | A FFDE | MONRAM | A FFB4 |
| NEXADR | A 0037 | PAAN | A 039A | PAANF1 | A 03BB | PACK | A 0225 | PACK1 | A 0231 | PACK2 | A 0237 | PACK3 | A 0240 |
| PACK4 | A 0245 | PACK5 | A 0254 | PARAM | A 03C6 | PARAM2 | A 03E1 | PARAM3 | A 03EF | PRET | A FFBC | RAMHZ | A FFE0 |
| RDEF | A 0401 | RECHTS | A 027D | REGAD | A FFCA | RIGH1 | A 0295 | RIGHT | A 0294 | SDEF | A 03FB | SLAZL | A 07BF |
| SPEI1 | A 030D | SPEICH | A 02FA | SRET | A FFB8 | START | A 0000 | START1 | A 0037 | STKINI | A 0000 | TABZ | A 01D9 |
| TAST | A 0183 | TIMEH | A 00FD | TIMEL | A 00FC | UMR1 | A 0138 | UMR2 | A 013F | UMR3 | A 014D | UMREND | A 0179 |
| UMS1 | A 00FA | UMS3 | A 00FE | UMS4 | A 0117 | UMW | A 02DB | WAREIN | A 07D9 | | | | |

ASSEMBLY COMPLETE, NO ERRORS

LISTADO DE EPROM
HABILITADORA DE CI

ORG 484D

5FH

5FH

3FH

3FH

ORG 492D

5FH

5FH

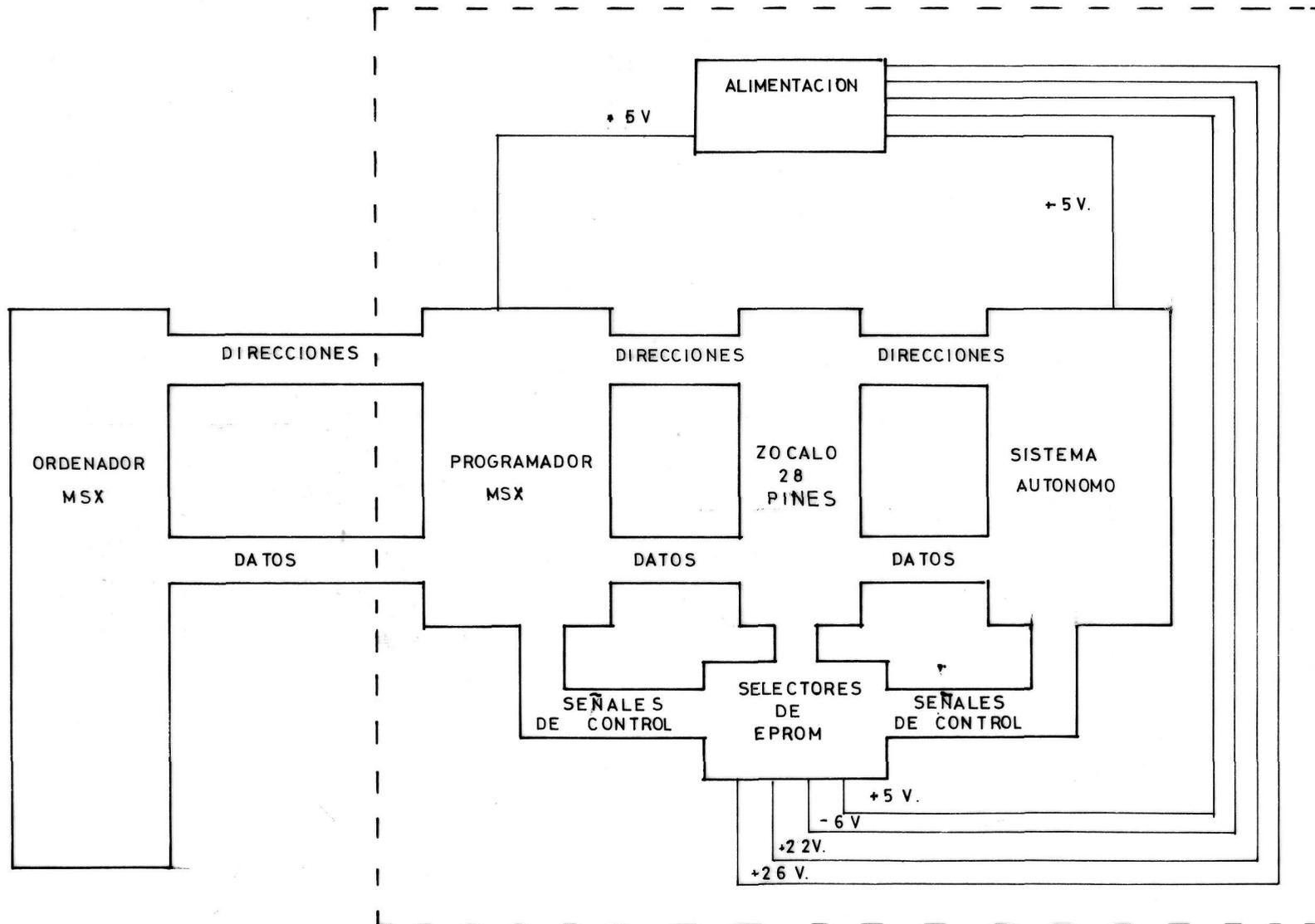
3FH

3FH

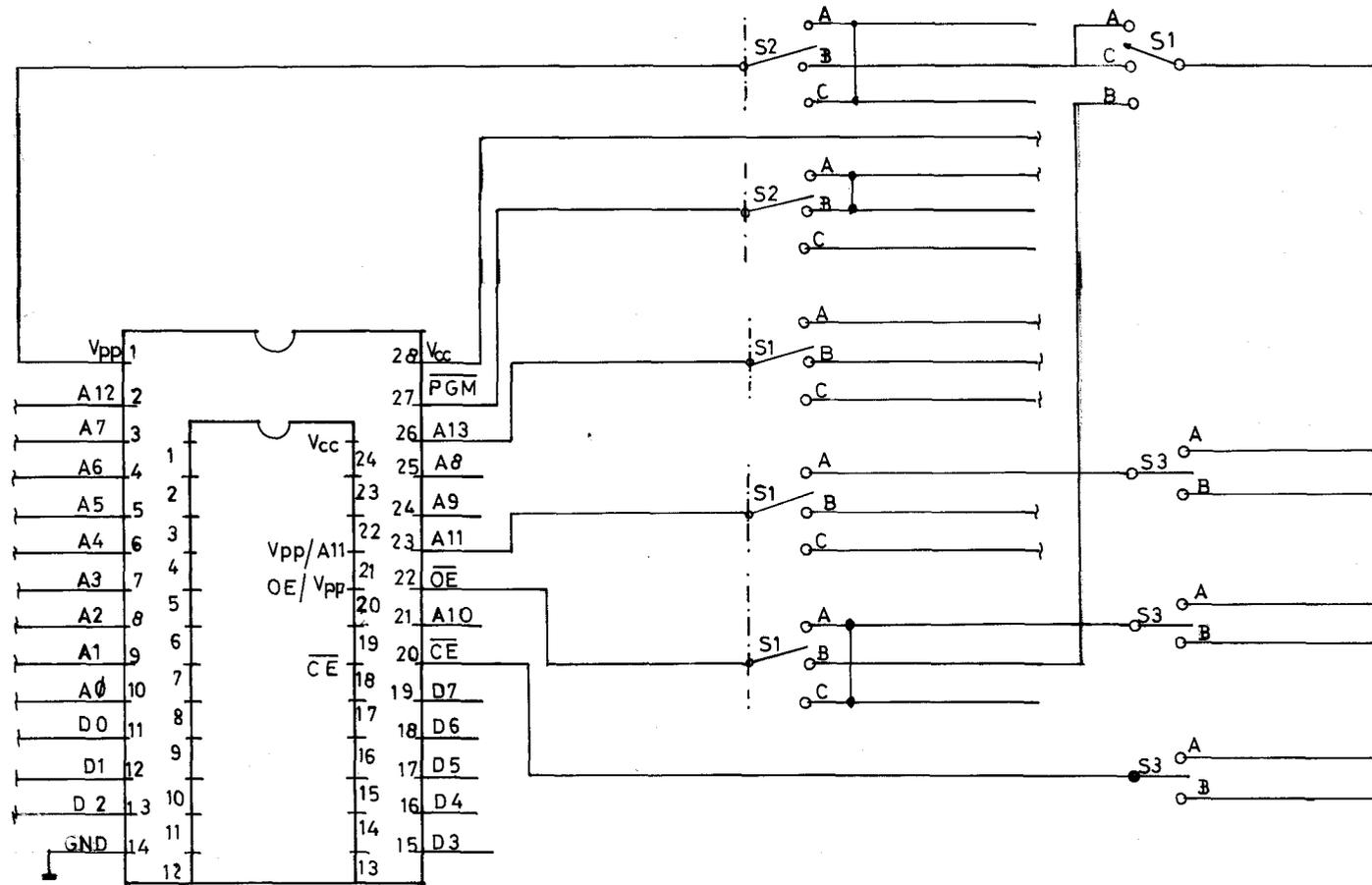
75H

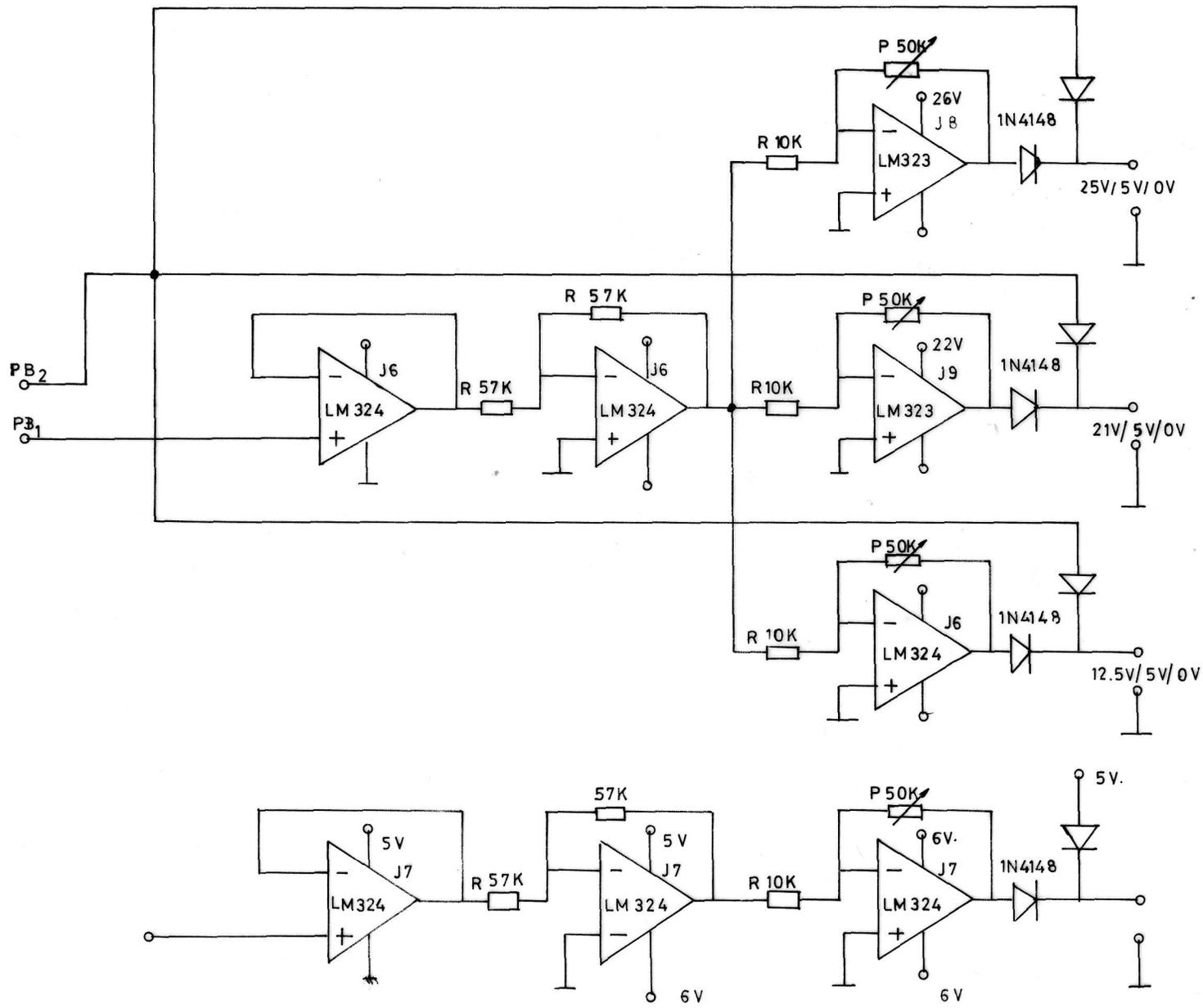
END

DIAGRAMA DE BLOQUE DEL PROGRAMADOR



ESQUEMA CONMUTADORES





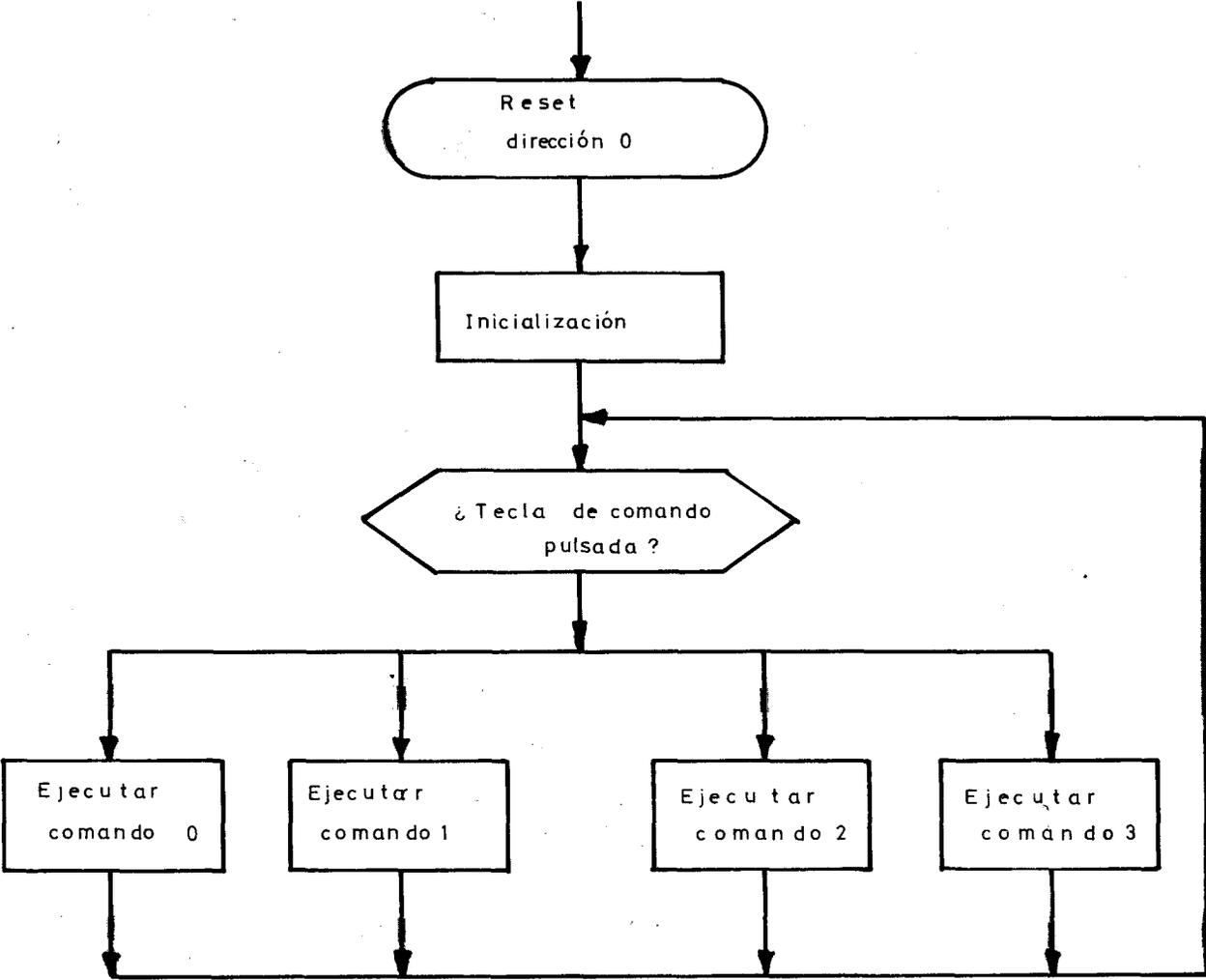
ESQUEMA CONTROLADOR DE TENSIONES

SELECTOR DE EPROM

| CONECTOR EPROM | 1 PO S I C I O N | 2 PO S I C I O N |
|-------------------|---------------------|---------------------|
| 2716 | A | X |
| 2732A | B | X |
| P2732 | B | X |
| 2764 | C | B |
| 2764A | C | A |
| P2764 | C | B |
| P2764A | C | B |
| 27128 | C | B |
| 27128A | C | B |
| 27256 | C | C |

X: Indiferente

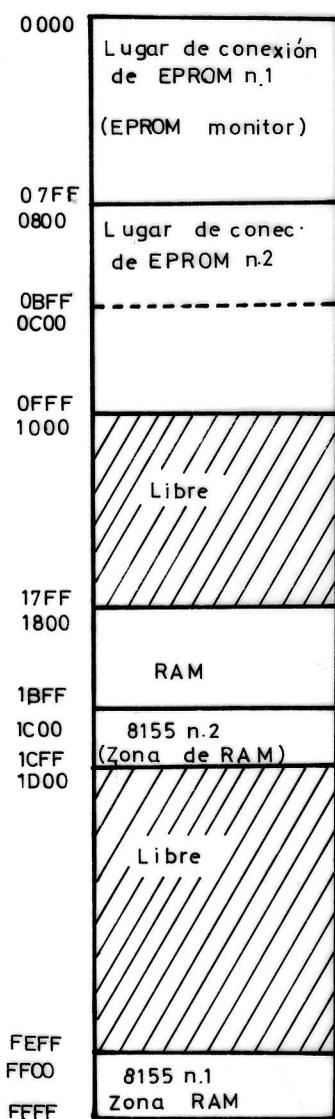
Diagrama de flujo del funcionamiento general del programa monitor



Mapas de direccionamiento

SISTEMA AUTONOMO

Mapa de direccionamiento de memoria



Mapa de direccionamiento de E/S

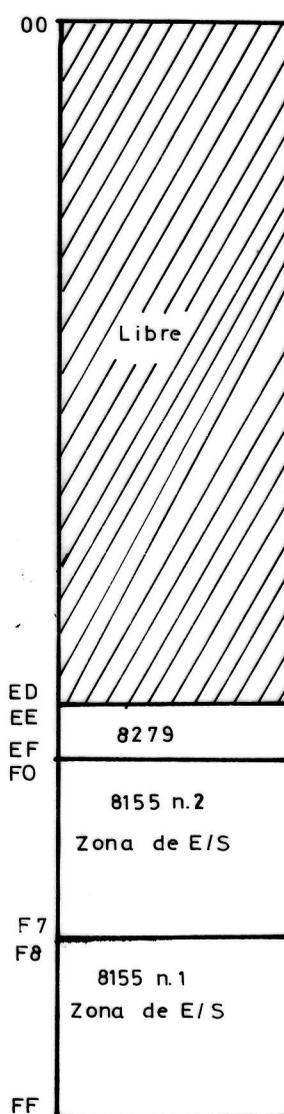
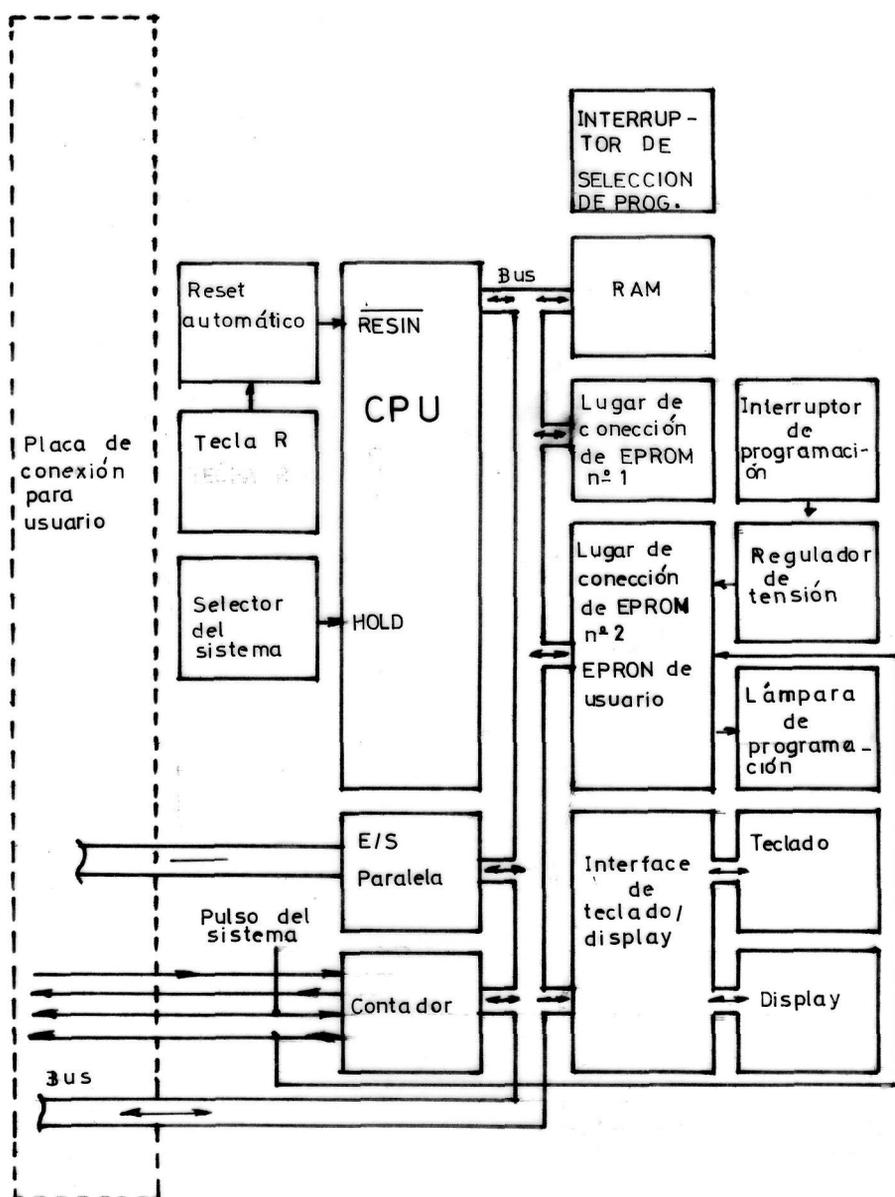
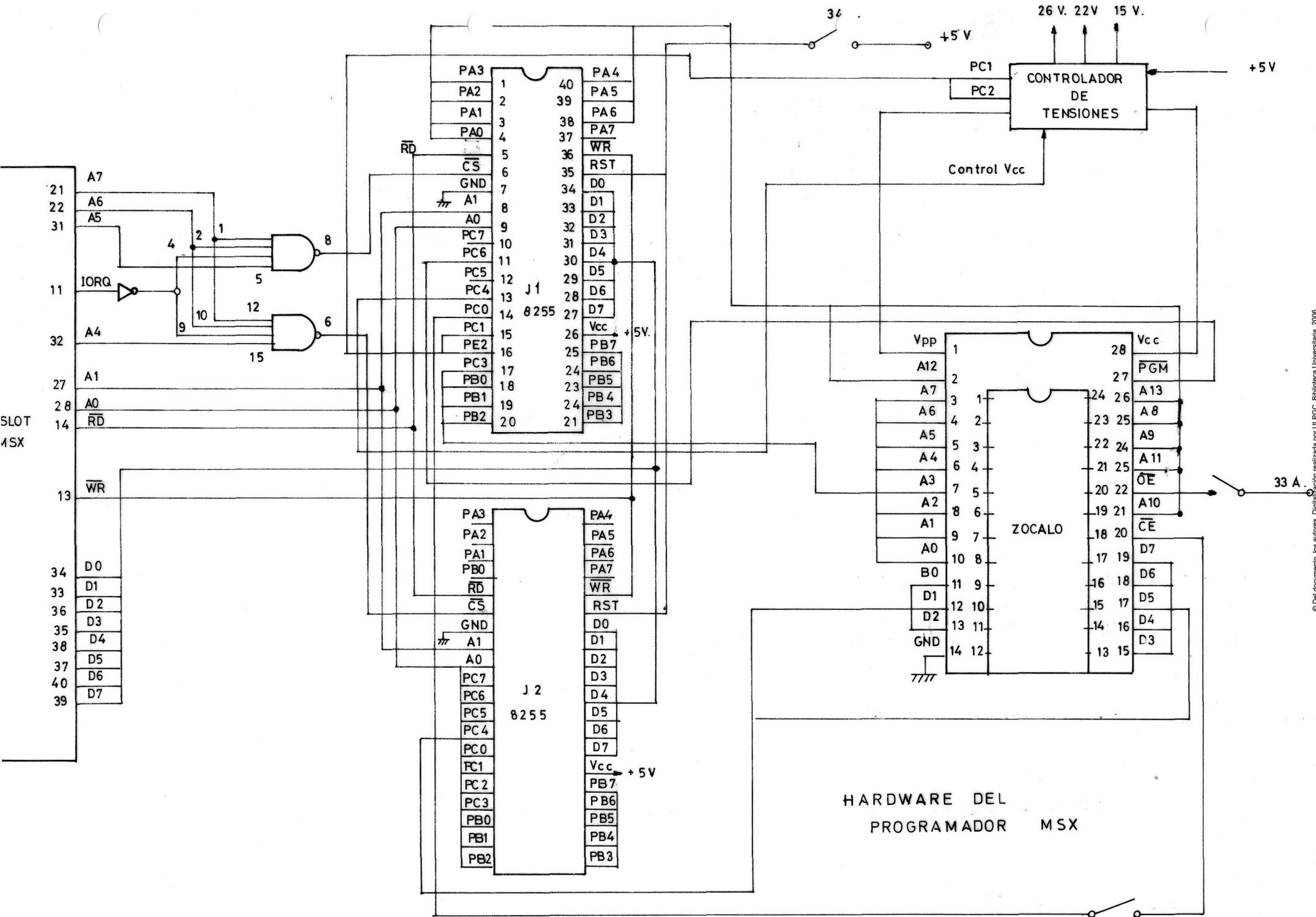
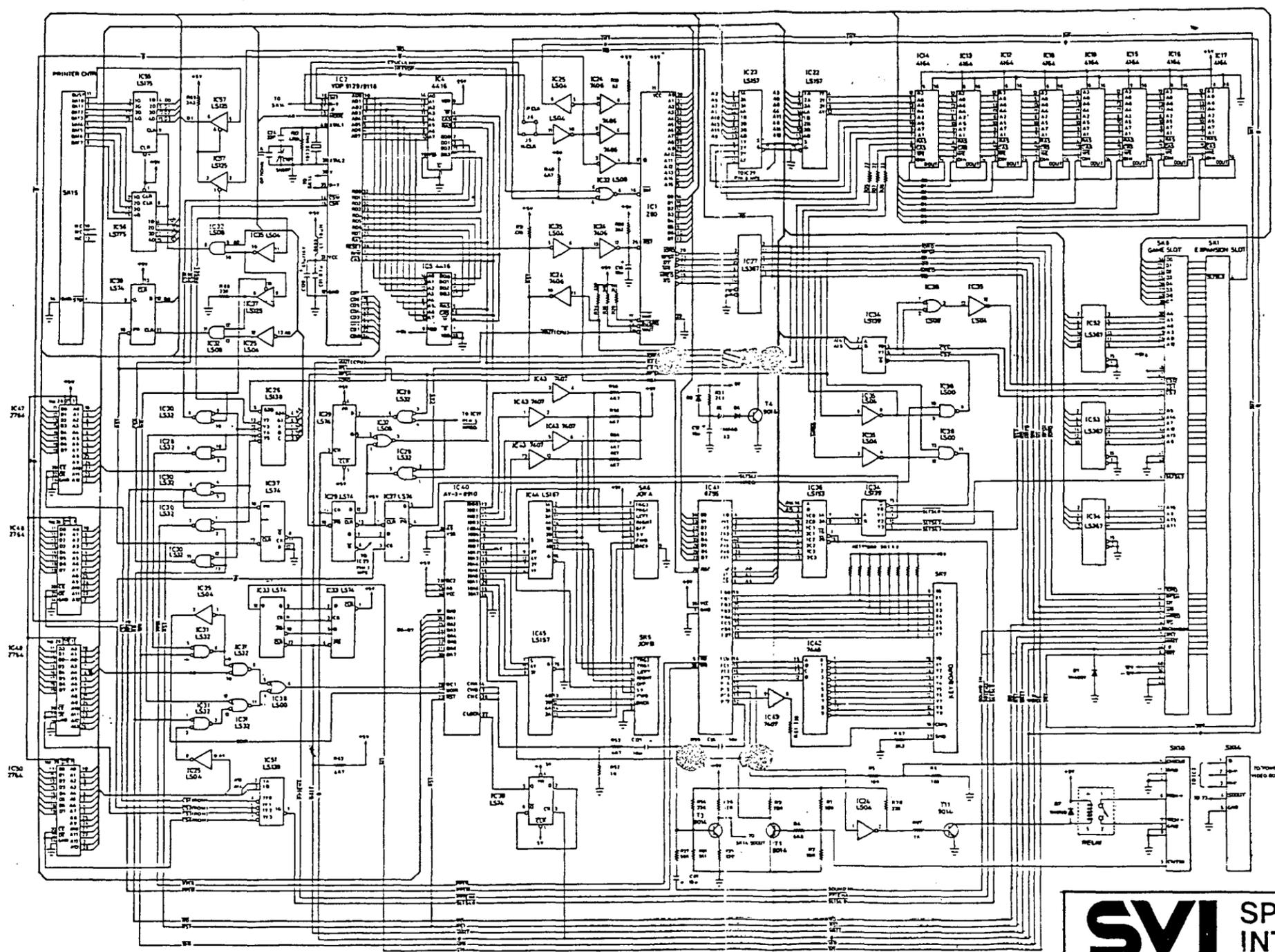


Diagrama de bloques del sistema mínimo





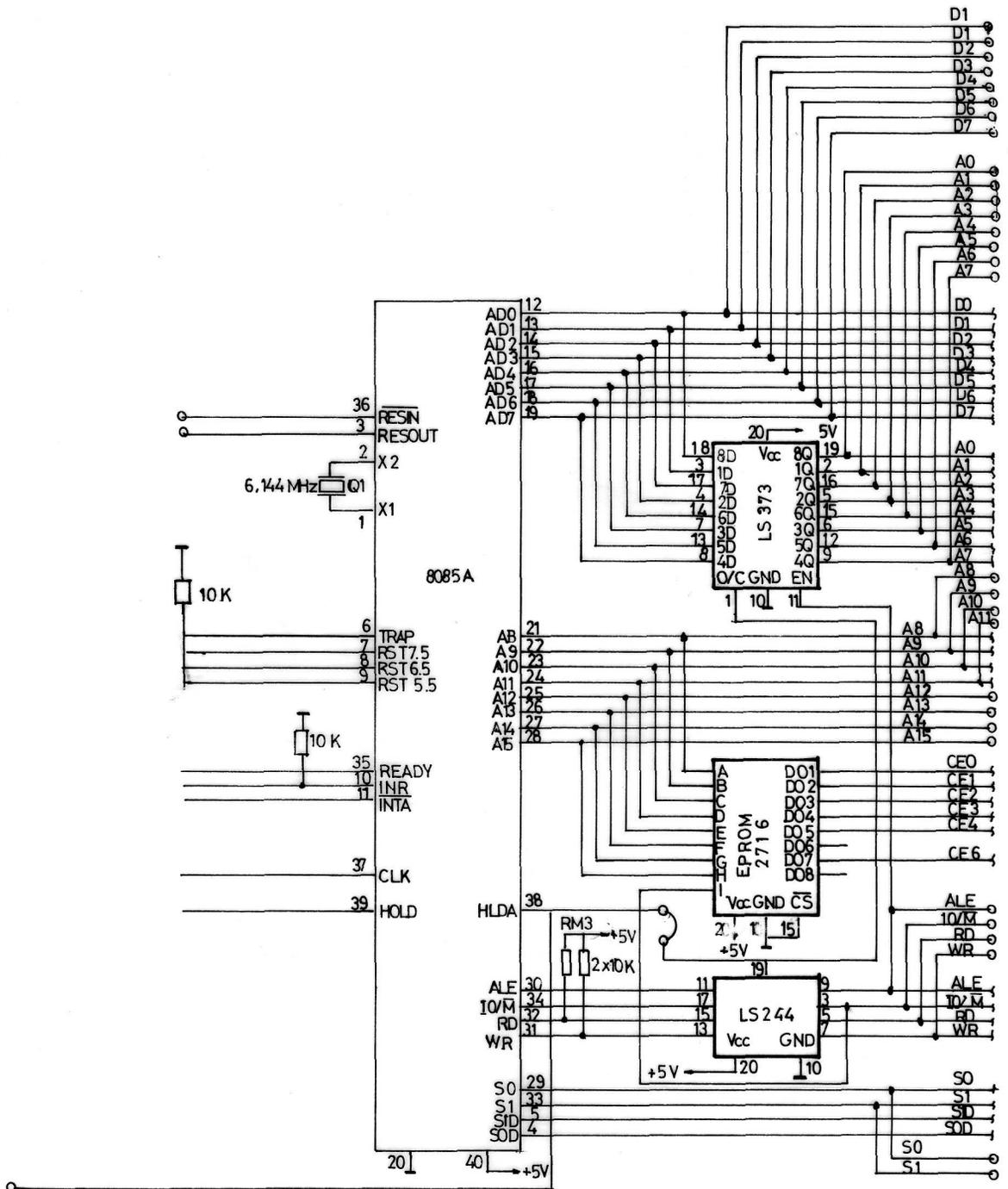


1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100

SVI SPECTRAVIDEO INTERNATIONAL LIMITED

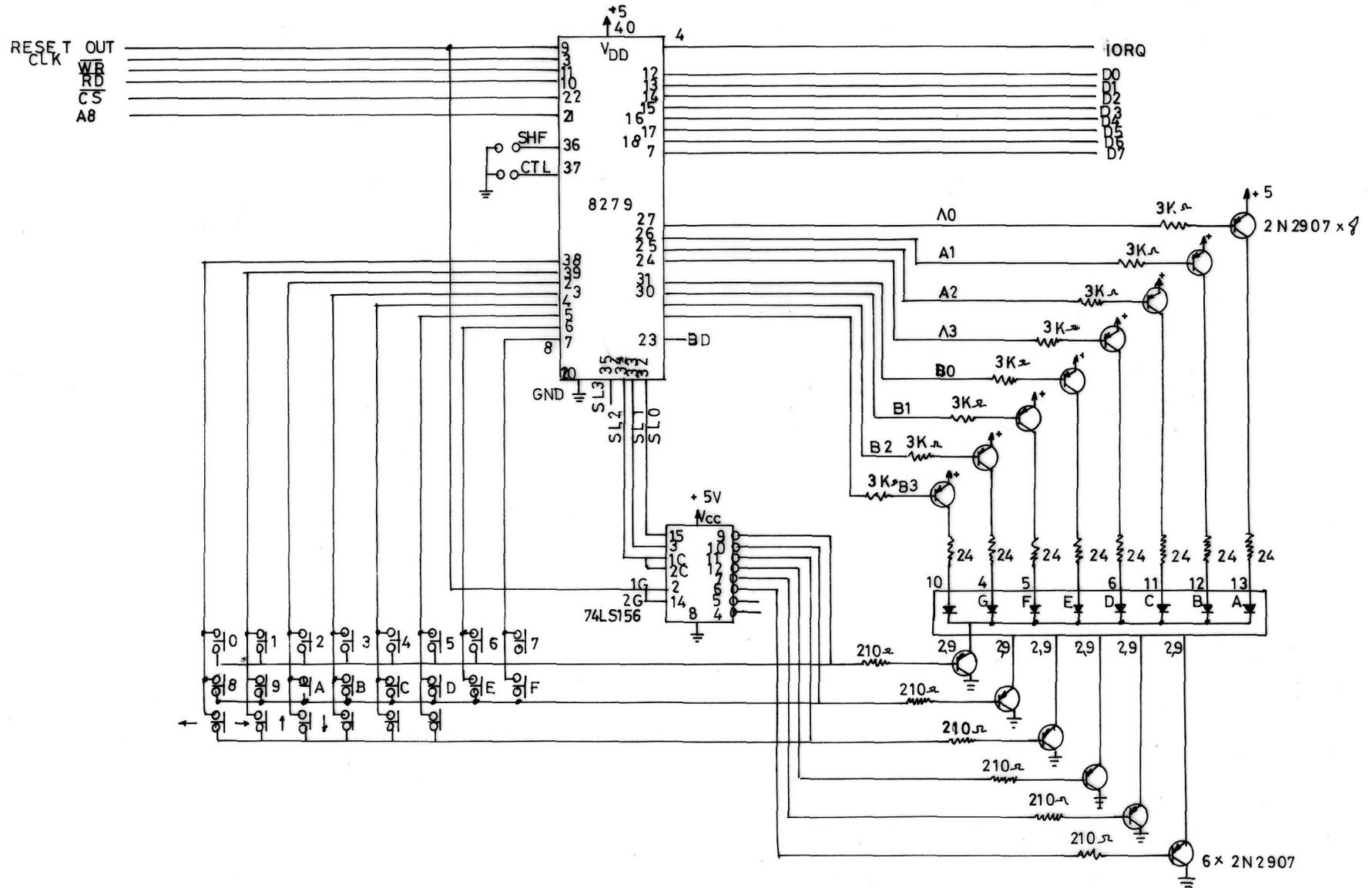
CONTROLLED ENGINEERING DOCUMENT
 THIS DOCUMENT CONTAINS CONFIDENTIAL INFORMATION WHICH IS THE EXCLUSIVE PROPERTY OF SPECTRAVIDEO INTERNATIONAL LTD.

| | |
|---|---|
| TITLE <i>MSX 728 SCHEMATIC DIAGRAM (MAIN BOARD) - EPROM VERSION -</i> | VERIFIED BY <i>KIMBA LAU</i> DATE <i>29-1-1985</i> |
| DOCUMENT NO <i>STM-728-018</i> | VERSION <i>1.0</i> SHEET <i>1</i> NC OF <i>1</i> |



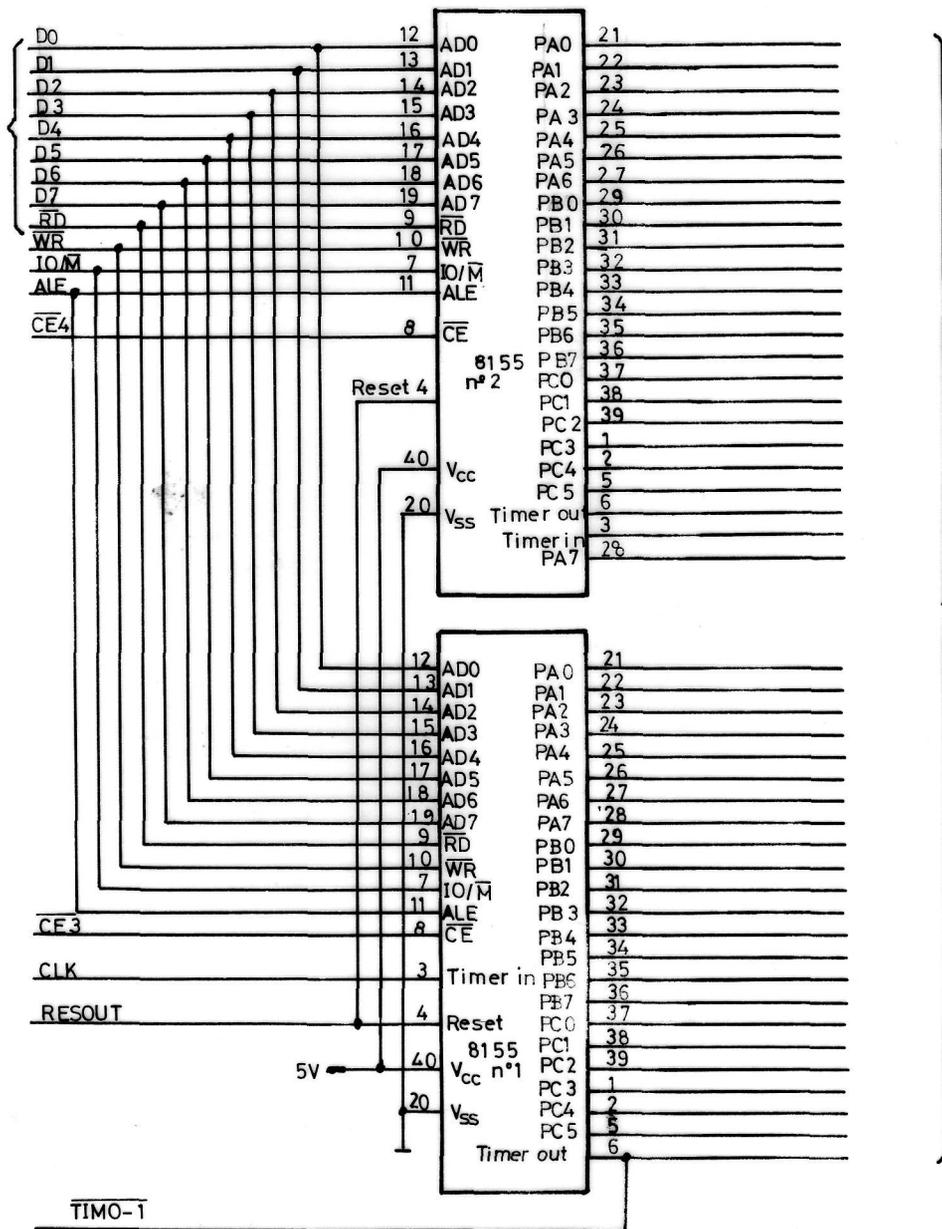
ESQUEMA UNIDAD CENTRAL SISTEMA AUTONOMO

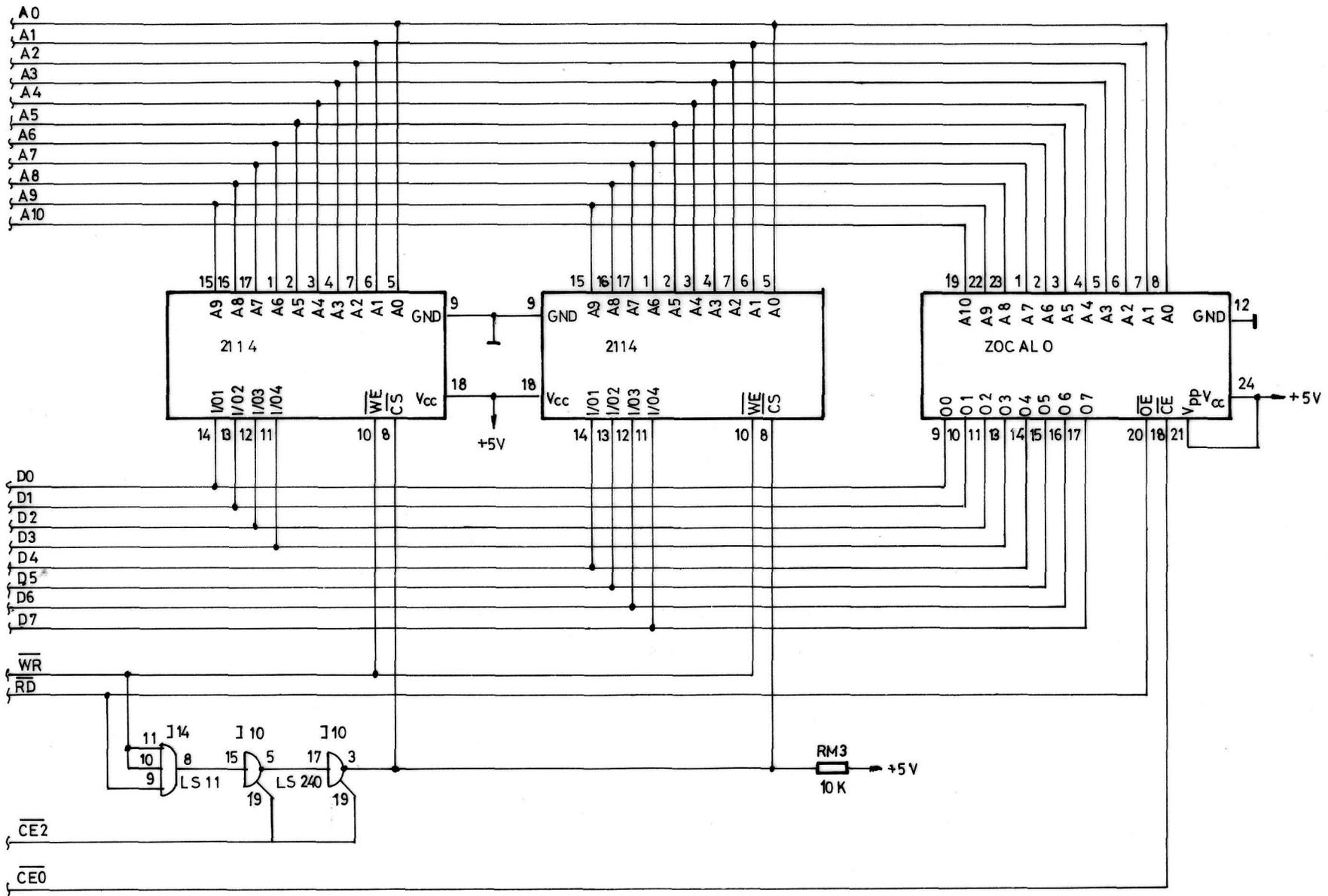
CONTROLADOR DE TECLADO Y DISPLAYS



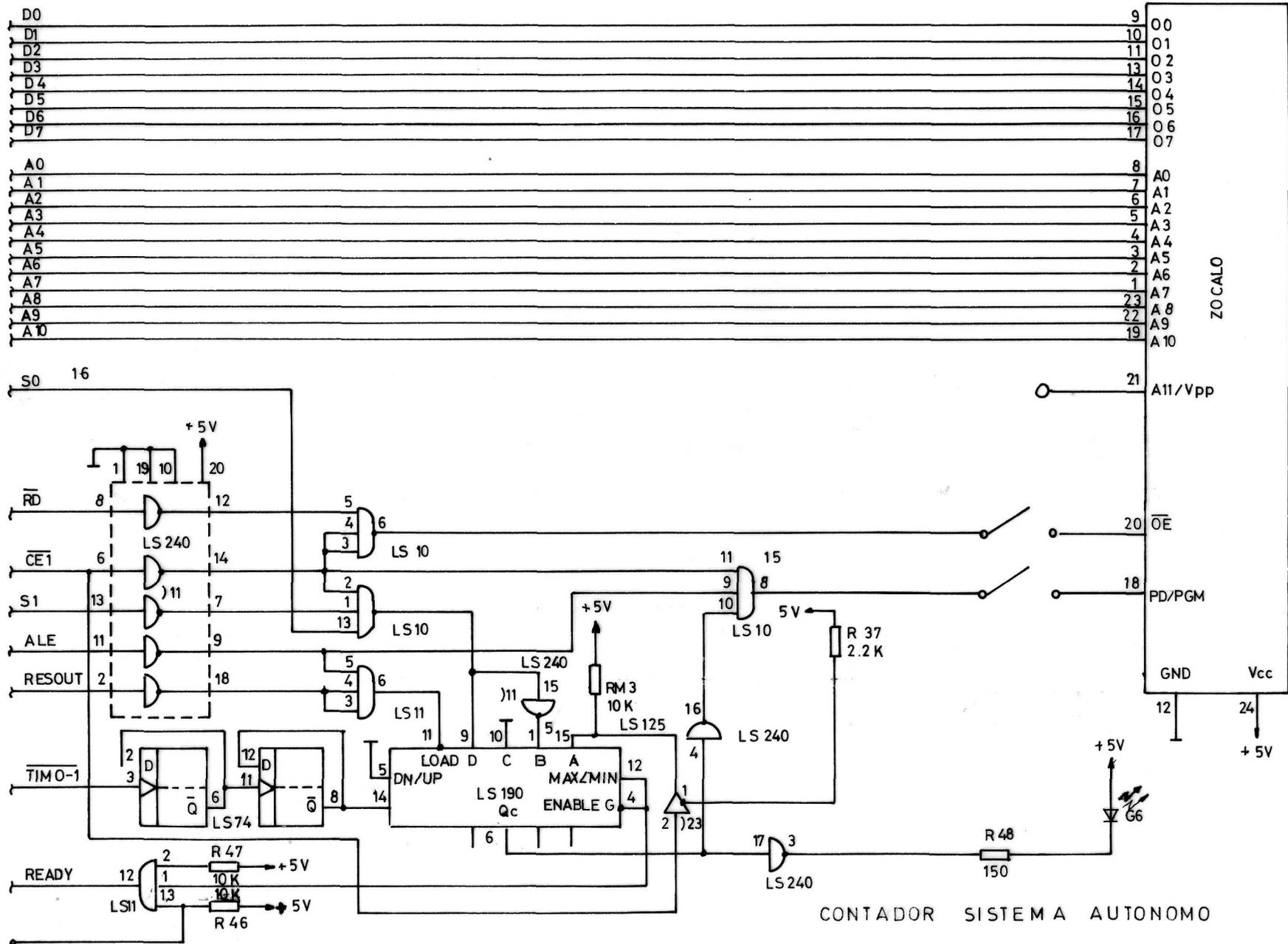
Puertos de expansión

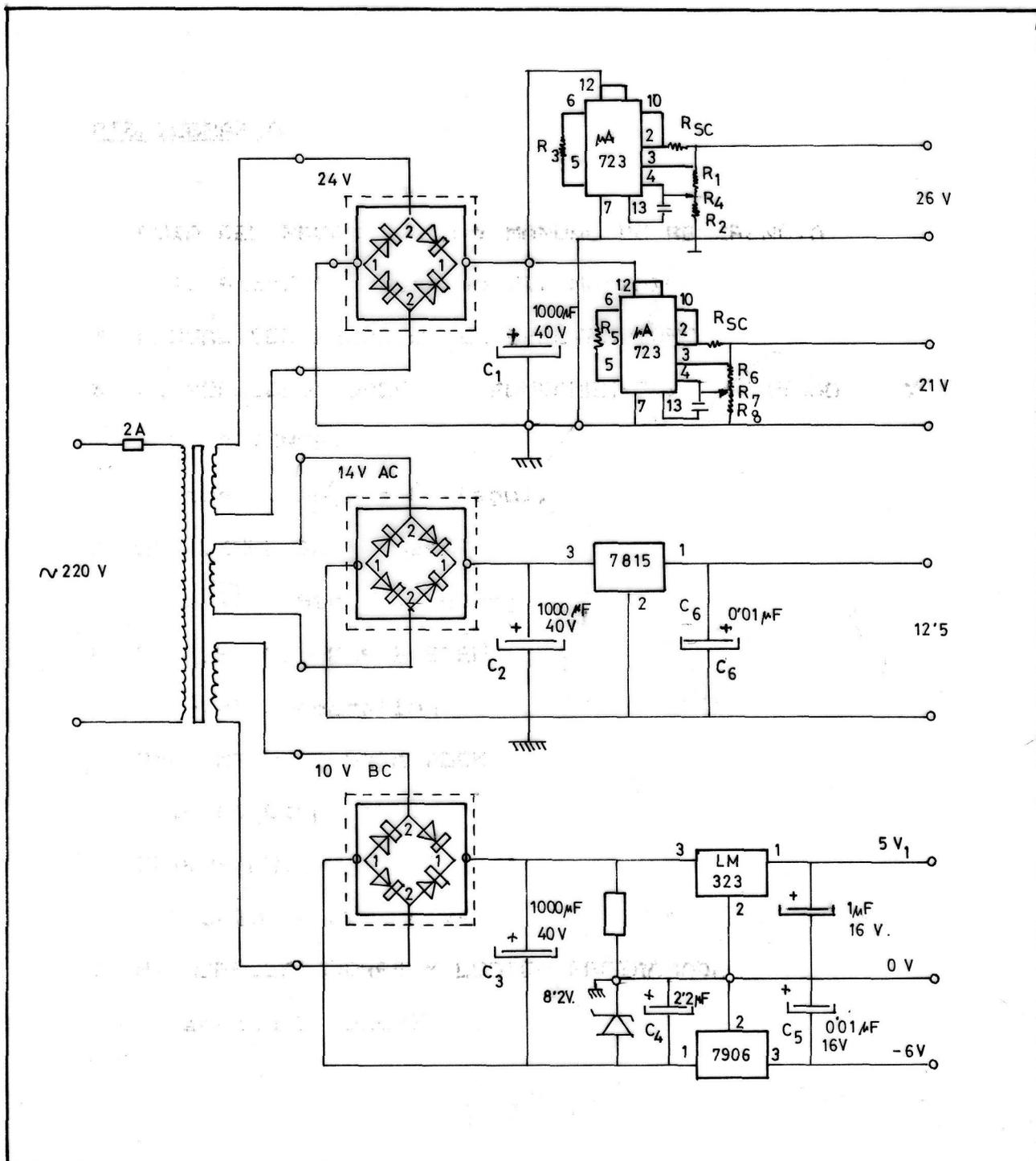
DEL SISTEMA AUTONOMO





ESQUEMA DE MEMORIA RAM SISTEMA AUTONOMO





ALIMENTACION

BIBLIOGRAFIA

- * GUIA DEL PROGRAMADOR Y MANUAL DE REFERENCIA
T. Sato/ P. Hapstone /I. Muriel
- * MANUAL DEL USUARIO DEL SPECTRAVIDEO
- * MICROPROCESADORES. FUNDAMENTOS, DISEÑO Y
APLICACIONES.
José M. Angulo Usategui.
- * MSX LENGUAJE MAQUINA
Hullin- Stra. Ssenburg
- * MICROPROCESSERS SYSTEM
Intel Corporation
- * ONLY MEMORY. DATA BOOK
Intel Corporation
- * EXTENSIONES MSX
Elektor n.73 y n.74
- * MICROPROCESADORES Y LOGICA PROGRAMADA
Kerneth L. Short