

N: 42
27/04/94



ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON - UNAM

I N G E N I E R I A

"DISEÑO DE UN CIRCUITO SIMULADOR DE MEMORIA DE SOLO LECTURA
(ROM) UTILIZANDO COMO INTERFASE UNA COMPUTADORA PERSONAL"

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA
P R E S E N T A :

LUIS ZACARIAS SANDOVAL

ASESOR:

Ing. Eleazar M. Pineda Díaz

ENEP



ARAGON

San Juan de Aragón, Edo. de Méx., 1994

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGÓN
DIRECCION

LUIS ZACARIAS SANDOVAL
P r e s e n t e .

En contestación a su solicitud de fecha 7 de diciembre de 1993, relativa a la autorización que se le debe conceder para que el señor profesor, Ing. ELEAZAR M. PINEDA DIAZ pueda dirigirle el trabajo de Tesis denominado "DISEÑO DE UN CIRCUITO SIMULADOR DE MEMORIA DE SOLO LECTURA (ROM) UTILIZANDO COMO INTERFASE UNA COMPUTADORA PERSONAL", con fundamento en el punto 6 y siguientes del Reglamento para Exámenes Profesionales en esta Escuela, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobada su solicitud.

Aprovecho la ocasión para reiterarle mi distinguida consideración.

ATENTAMENTE
"POR MI RAZA HABLARA EL ESPIRITU"
San Juan de Aragón, Edo. de Méx., abril 15 de 1994
EL DIRECTOR


M EN CLAUDIO C. MERRIFIELD CASTRO

Nota: La aceptación del tema de tesis y asesor de la misma fue registrado en la Unidad Académica de esta Escuela con fecha 21 de enero de 1994.

c c p Lic. Alberto Ibarra Rosas, Jefe de la Unidad Académica.
c c p Ing. Raúl Barrón Vera, Jefe de Carrera de Ingeniería
Mecánica Eléctrica.
c c p Ing. Eleazar M. Pineda Díaz, Asesor de Tesis.

AGRADECIMIENTOS

Le Agradesco a Dios por darme la Vida y por brindarme la oportunidad de concluir mis estudios profesionales.

El más grande de los reconocimientos para mis padres Josefina Sandoval A. y Lucio Zacarías Z. por haberme dado su Amor, Cariño y apoyo para lograr esta gran meta que es darme una profesion, del cual estoy orgulloso de mi familia.

A MI MADRE: JOSEFINA SANDOVAL A.

Por todo el Amor, Cariño, Comprension y Apoyo durante toda mi vida, principalmente en mis estudios profesionales.

A MI PADRE: LUCIO ZACARIAS Z.

Por enseñarme el valor de la Vida, por formarme en una persona provechoza y por darme esta gran herencia que es la Educación.

A MIS HERMANOS: Minerva, Ambrocio, Lucio, Rosa María, Israel, Ismael y Eduardo.

Por brindarme su Amor, Cariño, Comprensión y Apoyo en todo momento de nuestras vidas.

Toda mi familia reciba las gracias por su apoyo brindado para terminar este trabajo.

A MI ASESOR: ING. Margarito E. Pineda Díaz.

Por todo el apoyo, sugerencias y dedicación prestada para la realización de este trabajo.

AL GRUPO DEL JURADO: Quienes con sus observaciones y sugerencias han hecho de este uno de mis mejores trabajos. Les agradezco su participación incondicional y profesional.

A MIS COMPAÑEROS Y AMIGOS: Por brindarme su amistad y apoyo en todas las etapas de mi vida.

Mi agradecimiento a la Universidad Nacional Autónoma de México y a la Escuela Nacional de Estudios Profesionales Aragón por brindarme la oportunidad de una formación Cultural y Profesional.

Mi agradecimiento a los Ing. Francisco Javier Ramírez J. y Marco Antonio Torres B. por el asesoramiento y apoyo brindado para la realización de éste trabajo.

Mi agradecimiento al Instituto Nacional de Investigaciones Nucleares por el apoyo y material brindado para la realización de éste proyecto.

**"DISEÑO DE UN CIRCUITO SIMULADOR DE MEMORIA DE SOLO LECTURA
(ROM) UTILIZANDO COMO INTERFASE UNA COMPUTADORA PERSONAL"**

I N D I C E

	Pag.
INTRODUCCION	1
CAPITULO 1 ANTECEDENTES	6
1.1 PROPOSITO Y JUSTIFICACION DE LA TESIS	9
1.2 PLANTEAMIENTO DEL DISEÑO	10
1.3 REQUERIMIENTOS A CUBRIR Y CARACTERISTICAS QUE DEBE TENER . EL CIRCUITO SIMULADOR DE ROM	13
1.4 ORGANIZACION DE LA COMPUTADORA	14
1.5 PANORAMA DE LA EVOLUCION DE LAS PC'S	18
1.6 EL PUERTO SERIE Y PARALELO	22
1.7 LA TARJETA MADRE	24
1.8 EL CONECTOR DE EXPANSION	25
1.9 DESCRIPCION DE LAS TERMINALES DEL CONECTOR DE EXPANSION .	27
1.10 MAPA DE DIRECCIONAMIENTOS DE LA MEMORIA DE LA PC PARA EL CONECTOR DE EXPANSION.	31
CAPITULO 2 DISEÑO DEL CIRCUITO SIMULADOR DE ROM	34
2.1 ESTRUCTURA DEL SISTEMA	34
2.2 DISEÑO DEL CIRCUITO	36
2.2.1 DISEÑO DEL HARDWARE	36
2.2.1.1 ALTERNATIVAS DE SOLUCION	37
2.2.1.2 SELECCION DE COMPONENTES	39
2.2.1.3 BASES DE DISEÑO	42
2.2.1.4 DIAGRAMA A BLOQUES DEL CIRCUITO SIMULADOR DE ROM	43
2.2.1.5 CONDICIONES DE OPERACION DE LOS ELEMENTOS	53
2.2.2 DISEÑO DEL SOFTWARE	55
2.2.2.1 DIAGRAMA DE FLUJO DEL PROGRAMA	55
2.2.2.2 DESARROLLO DE LA PROGRAMACION DEL CIRCUITO	58
SIMULADOR DE ROM	58

	Pag.
CAPITULO 3 CONSTRUCCION DEL CIRCUITO	68
3.1 DISEÑO DEL CIRCUITO IMPRESO	68
3.2 CONSTRUCCION DEL CIRCUITO IMPRESO	70
3.3 INTERRUPTOR Y CABLE PLANO	73
3.4 CONECTOR DB-25 PARA LA TARJETA SIMULADOR DE ROM	76
 CAPITULO 4 MONTAJE Y PRUEBAS	 79
4.1 MONTAJE	79
4.2 PRUEBAS	80
 CAPITULO 5 RESULTADOS Y CONCLUSIONES	 84
5.1 RESULTADOS	84
5.2 ALCANCES	93
5.3 CONCLUSIONES	95
 BIBLIOGRAFIA	 97
 APENDICES	 98
A) LISTA DE COMPONENTES	99
B) HOJA DE DATOS DE FABRICANTES DE CI	101
C) MANUAL DE OPERACION	109
D) LINEAMIENTOS PARA ESCRIBIR UN PROGRAMA EN LENGUAJE ENSAMBLADOR	116
E) ESPECIFICACION ISA PARA LAS TARJETAS IMPRESAS	119
F) GLOSARIO DE TERMINOS	121

INTRODUCCION

El Microprocesador en las últimas décadas a tomado una gran importancia debido a que es un pequeño pero extremadamente complejo dispositivo de gran escala de integración, que maneja y procesa datos aritméticos bajo el control de un programa.

En el microprocesador se encuentra la unidad aritmética y de control, contenidas en la unidad de procesamiento central (CPU). El microprocesador controla todas las unidades del sistema empleando las unidades de control. Cuenta con un "bus" de direcciones para seleccionar un lugar en la memoria. El "bus" de datos constituye una vía de dos sentidos para la transferencia de datos entre la unidad de procesamiento y memoria del cual se envían o reciben datos.

El microprocesador siempre tiene una secuencia, recibe información, decodifica y ejecuta, es decir que recibe las instrucciones de la memoria del programa, después la decodifica (traduce la instrucción) y ejecuta la orden.

El microprocesador no sólo se utiliza en las computadoras, si no en una gran variedad de aplicaciones como son automoviles, videojuegos, sistemas de alarma, sistemas de comunicación etc.

Un microprocesador típico no contiene memoria y no es capaz de realizar funciones de entrada y salida (E/S) de información en una computadora o en un sistema en prueba, para ello es necesario conectar otros circuitos integrados (CI) que realicen estas funciones. Estos circuitos son llamados memorias.

Hay dos tipos de memorias que pueden ayudarle al microprocesador en estas funciones, la RAM y la ROM.

La memoria de acceso aleatorio (RAM) almacena datos o programas temporales.

La memoria de sólo lectura (ROM) en la cual se encuentra normalmente el programa permanente que hace funcionar al microprocesador.

Este tipo de memoria ROM la encontramos en cuatro formas:

La ROM estándar es una memoria de almacenamiento de información y su contenido no es alterado, además almacena instrucciones del programa que siempre son disponibles para el microprocesador.

La PROM (ROM programable), se programa permanentemente por el usuario y es programada solo una vez con las instrucciones ideales.

La EPROM (ROM programable y borrable), es programada y borrada por el usuario con rayos ultravioleta de alta intensidad. Se aplican por la parte de su pequeña ventana.

La EAPROM (memoria de sólo lectura eléctricamente alterable) programado por el usuario y borrado eléctricamente con equipo especial.

El microprocesador opera como parte de un sistema en el desarrollo de diseños o proyectos, junto con algún tipo de memoria y circuitos de compuertas básicas.

En la realización de algún proyecto se requiere de dos cosas fundamentales: rapidez en el desarrollo del proyecto y eficiencia en la operación; por lo que se requiere minimizar el tiempo y elevar la calidad de los trabajos, en los cuales se utilicen microprocesadores.

El presente trabajo muestra el diseño, construcción y pruebas de un Circuito Simulador de Memoria ROM (memoria de sólo lectura), que servirá como herramienta de uso común en la programación del microprocesador o cualquier sistema en el que se utilice una EPROM, ya que el circuito simula la programación de esta memoria para que sea verificada.

El simulador de ROM está constituido por una tarjeta de circuito impreso que realiza una interfase paralela con el "bus" o "conector de expansión" de la computadora tipo XT o AT, compatible con IBM. Esta tarjeta se instala dentro del conector de expansión de la PC, la cual se encuentra ubicada en la parte izquierda de la tarjeta madre en el interior de la computadora. Todo lo anterior se describe en el capítulo 1.

El circuito es capaz de simular una memoria ROM con capacidad de almacenamiento en memoria de 8 Kbytes, con la ventaja de modificar en ella su contenido por programación en caso de requerir alguna corrección. Esta tarjeta toma información de un archivo escrito en ASCII, la interpreta y la deposita en una localidad de memoria específica del simulador. Dicha información es utilizada por el sistema donde se va a instalar la memoria ROM que se está probando. Dicho sistema utiliza algún tipo de microprocesador, memoria, decodificadores o compuertas básicas. (el conjunto de estos elementos se le conoce como sistema mínimo).

El circuito simulador de ROM tiene la finalidad de acelerar el proceso de desarrollo en los programas para sistemas que utiliza microprocesadores y memorias ROM o EPROM.

El simulador además de que agiliza la programación, ayuda al diseñador en la corrección de errores, ya que utilizando el simulador puede verificar si el programa a simular realiza las funciones adecuadas antes de grabar a la memoria ROM (memoria sólo de lectura) o una EPROM (memoria de sólo lectura borrable).

Si se encuentra algún error en la simulación del programa, se deben realizar las correcciones apropiadas antes de grabar definitivamente un programa, minimizando así el tiempo en el desarrollo del sistema mínimo.

El trabajo de esta tesis consta de 5 capítulos y esta integrada de la siguiente forma:

En el capítulo 1 se mencionan los requerimientos, características que debe tener el circuito, justificación y propósito de la tesis, así como los antecedentes para llevar a cabo el proyecto, es decir se contempla un panorama general de los conceptos básicos sobre la computadora y la evolución del microprocesador; así como la parte teórica sobre el puerto serie y paralelo, la tarjeta madre, las señales del conector de expansión de la PC. Es necesario conocer también el mapa de direccionamiento de la memoria de la PC para llevar a cabo la interfase con la tarjeta simulador de ROM, por lo que se da esta información en el capítulo 1.

En el capítulo 2 se habla sobre el diseño de la tarjeta. En este capítulo se trata desde la estructura general, las alternativas de solución, la selección de componentes y el diseño del circuito tanto del "hardware" como del "software", es decir los elementos que integran al circuito, así como el programa que controla el circuito simulador de ROM.

El capítulo 3 comprende el diseño del circuito impreso, es decir la distribución de los componentes, la ruta de intercomunicación entre los dispositivos que integran la tarjeta y su construcción.

El capítulo 4 describe el procedimiento para el montaje de los elementos que integran la tarjeta y las pruebas de carácter general para verificar que no existan cortos circuitos, falsos contactos y pistas o rutas de comunicación abiertas. Así como pruebas de funcionamiento de la tarjeta, verificando que el programa a simular funcione adecuadamente.

El capítulo 5 muestra los resultados a los que se llegó y las conclusiones del diseño.

Además se anexa en el apéndice el manual de operación para usar adecuadamente la tarjeta simulador de ROM y un glosario de terminos de las palabras poco comunes, el cual se recomienda que el lector las consulte.

C A P I T U L O 1

A N T E C E D E N T E S

En las últimas décadas la electrónica digital se ha desarrollado ampliamente gracias al material semiconductor del Silicio y sobre todo al microprocesador. El microprocesador es un pequeño pero extremadamente complejo dispositivo de muy alta escala de integración (VLSI) programable. Por integración se entiende a la gran escala de arreglos de circuitos interconectados en un sólo semiconductor o circuito integrado. Dispositivo que tiene la capacidad de procesamiento de las grandes computadoras.

Los microprocesadores son la base de un nuevo tipo de dispositivos inteligentes, su utilidad en la actualidad es muy variada, se encuentran en diversos productos: desde los juguetes de los niños hasta los automóviles, desde los electrodomésticos hasta las microcomputadoras, y desde la aplicación en equipo industrial hasta los robots.

Un microprocesador típico no contiene memoria y no es capaz de realizar funciones de entrada y salida de alguna operación por sí sólo, es necesario conectar otros circuitos que ayuden para la transferencia de información, comúnmente se utilizan buffers. Este circuito integrado ayuda a transferir señales de una sección de un sistema a otro, se usa para compensar la velocidad del flujo de datos de un dispositivo a otro. Además se requiere de un circuito en el cual se almacene información y se tenga disponible para indicarle al microprocesador lo que debe realizar.

Este último circuito se llama memoria y tiene la característica de almacenar información. Las memorias semiconductoras se dividen en dos grupos: memoria RAM (memoria de lectura/escritura) y memoria ROM (memoria de sólo lectura).

La memoria de lectura/escritura es una memoria que puede ser fácilmente programada, borrada y reprogramada por el usuario. La programación se denomina escribir en memoria. Copiar datos de la memoria sin destruir el contenido, se denomina leer en memoria. La memoria de lectura/escritura se denomina RAM (memoria de acceso aleatorio), es volátil, lo cual significa que se perderá la información que contiene si se desconecta la alimentación del CI. Las RAM se utilizan para almacenar datos o programas temporales. Las RAM existen de varias capacidades de acuerdo a su aplicación; las hay de 16 kbits, 2, 8, y 32 Kbytes, de acuerdo a su capacidad las tenemos de 16, 22, 24, y 28 terminales de conexión.

Las memorias RAM se subdividen en dos grupos. Si la memoria de lectura/escritura contiene circuitos tipo "cerrojos" como celdas de memoria, se denomina RAM estática (SRAM), si la celda de memoria está basada en una capacitancia se denomina RAM dinámica (DRAM). En estas memorias el contenido se pierde después de un tiempo de 5 ó 10 ms. Para poder conservar su contenido es necesario efectuar un refresco periódicamente cada 2 ms. El proceso de refresco consiste en leer el contenido y volver a grabarlo en las mismas posiciones. La RAM estática no necesita refrescarse y mantiene su información binaria, indefinidamente, todo el tiempo que esté alimentado el CI. Las RAM dinámicas tienen mayor capacidad y menor consumo de potencia que las estáticas.

Una última memoria semiconductoras más moderna es la RAM no volátil (NVRAM), la cual combina RAM estática y PROM borrables eléctricamente (EEPROM).

El otro tipo de memoria es la ROM, el cual se encuentra en cuatro formas:

La ROM estándar que es programada por el fabricante. La PROM (memoria de sólo lectura programable) puede ser programada permanentemente por el usuario o distribuidor utilizando un equipo especial. Puede ser programada sólo una vez. La EPROM (memoria de sólo lectura programable borrable), puede ser programada y borrada por el usuario. Los datos almacenados en la EPROM pueden ser borrados aplicando luz ultravioleta de alta intensidad, a través de la ventana especial que tienen en la parte superior. La EAPROM (memoria de sólo lectura alterable eléctricamente) que puede ser borrada y programada por el usuario con equipo especial; la EAPROM se borra eléctricamente y no con luz ultravioleta.

La ROM, PROM, EPROM, y EAPROM son memorias no volátiles permanentes y no pierden sus datos cuando se desconecta la alimentación del CI.

Las ROM y EPROM son utilizadas más frecuentemente para almacenar instrucciones de programación para los microprocesadores o programas de algún sistema en el cual se deseen datos permanentes y no se requieran modificaciones.

Las memorias no volátiles (ROM, PROM, EPROM, y EAPROM) existen de varias capacidades de acuerdo a su aplicación, las hay desde 32, 256, y 1024 Bytes hasta de 2, 4, 8, 16, 32, 64, y 128 Kbytes. De acuerdo a su capacidad las encontramos desde 8, 14, 16 terminales hasta 24, 28, 32, y 40 terminales.

Al utilizar microprocesadores en un diseño para un proyecto específico se requiere un conjunto de circuitos integrados auxiliares que pueden ser compuertas básicas, algún tipo de memoria, codificadores, buffers, etc.

Al conjunto de todos estos elementos que se ocupan para un fin determinado (diseño de algún proyecto específico) se le denomina sistema mínimo.

1.1 PROPOSITO Y JUSTIFICACION DE LA TESIS

El proposito de la tesis es diseñar un circuito simulador de memoria de sólo lectura (ROM), el cual permita la simulación, verificación y prueba de la programación de memorias ROM o EPROM aplicadas a sistemas en los cuales se utilicen microprocesadores; que además permita agilizar la programación, minimizando el tiempo en el desarrollo de las memorias EPROM.

El presente trabajo también surge como respuesta a la necesidad que existe en los laboratorios de electrónica de contar con una herramienta que ayude a los diseñadores y a cualquier alumno de electrónica para simular, verificar y probar la programación de memorias EPROM.

La simulación de los programas y la verificación de las instrucciones para la aplicación de los microprocesadores son de gran importancia en las etapas de experimentación para cualquier diseño, en donde se utilicen memorias ROM o EPROM, ya que al programar una memoria EPROM, la programación le indica al microprocesador lo que debe realizar dentro de un sistema, pero si la programación presenta algún error la memoria se tendrá que borrar con luz ultravioleta, grabar nuevamente y comprobar que las instrucciones son correctas, este ciclo de borrar y grabar la memoria es incómodo, además se pierde tiempo. Para ello la tarjeta simulador de ROM se usará y simulará la programación de la EPROM trabajando en ella, haciendo las pruebas pertinentes, y correcciones de la programación en caso de error en la propia tarjeta, así hasta que la programación sea correcta para que se grave definitivamente en la EPROM del sistema final.

1.2 PLANTEAMIENTO DEL DISEÑO

En el desarrollo de los diseños con microprocesadores se utiliza por lo general buffers, este circuito integrado permite transferir señales de una sección de un sistema a otro, proporcionando mayor capacidad de manejo de corriente y compensa la velocidad del flujo de los datos de un dispositivo a otro. Se utiliza comúnmente una memoria ROM o EPROM para la programación del microprocesador, así como de otros circuitos auxiliares como compuertas básicas, decodificadores, etc.

se utiliza además la computadora personal PC y una persona que se encarga de programar la memoria ROM o EPROM, utilizando algunos paquetes (procesador de palabras) y algún lenguaje de programación.

Al utilizar microprocesadores en un diseño para un proyecto específico, además de plantear los requerimientos se lleva a cabo una programación, en la cual se indican las instrucciones que realizará el programa en el microprocesador para el control del sistema diseñado, pero para concluir un trabajo de este índole se lleva tiempo en la programación y en las pruebas pertinentes a dicho diseño.

Comúnmente se utilizan memorias EPROM para la programación de los microprocesadores. En la mayoría de las ocasiones al realizar las pruebas de programación en una EPROM cuando el programa ha tenido algún error, la memoria EPROM se tendrá que borrar con luz ultravioleta para realizar las correcciones en la programación, grabar nuevamente la memoria y volver a realizar las pruebas, hasta concluir y verificar que el diseño y la programación cumplan con los requerimientos.

El grabar una memoria EPROM y borrarla cuando ha tenido algún error implica pérdida de tiempo, el quitar y poner el CI de una EPROM es muy incómodo, ya que se tendrá que quitar la memoria del porta integrado (base del integrado), y trasladarlo al borrador de memorias con mucha precaución utilizando una pulsera electrostática para la protección de la memoria, además de cuidar el tiempo de borrado comunmente de 15 a 20 minutos, ya que si pasa de este tiempo puede destruir a la memoria.

Por lo que el planteamiento del problema es diseñar un circuito electrónico Simulador de memorias ROM para que simule a la memoria tipo EPROM y pueda ser programada en lugar de la memoria real, con lo cual facilite la prueba y depuración de los programas.

Esta tarjeta tiene la finalidad de minimizar tiempo en el desarrollo de la programación en memorias EPROM, y en la programación de los microprocesadores que utilicen esta memoria. Esta tarjeta servirá como herramienta de uso comun para la programación de memorias EPROM y en este circuito simulador de ROM se harán las pruebas pertinentes del programa de cualquier diseño, donde se utilice este tipo de memoria.

Si la programación a simular de este tipo de memoria es incorrecta, se realizarán las respectivas correcciones en la propia tarjeta simulador de ROM, si la programación a simular es correcta se grabará definitivamente en la ROM o EPROM del sistema final.

La ubicación del simulador de memoria ROM dentro de un sistema de desarrollo con microprocesadores se muestra en la fig. 1.

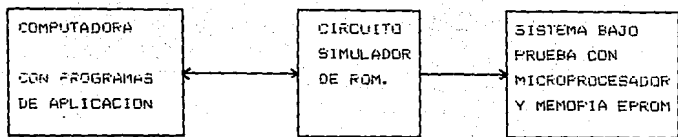


Fig. 1 Sistema de desarrollo con microprocesadores.

En la fig. 1 se muestra la relación que tendrán los programas de aplicación contenidos en la computadora personal PC, (Programa AVEMAC51, AVSIM51, XASM51.COM y archivos con extensión HEX que son programas para un sistema con microprocesador; el circuito que va a simular los programas para la EPROM (Simulador de ROM), y el sistema que contiene al microprocesador y a la memoria EPROM que será programada una vez que sea checada la programación.

En el primer bloque encontramos a la computadora personal PC en el cual se elabora el programa para la memoria EPROM para una aplicación específica. Este programa se realiza en lenguaje ensamblador, utilizando para esto un editor de texto. Después por medio del programa XASM51.COM se convierte el programa de lenguaje ensamblador a un archivo tipo "ASCII" (American National Standard Code for Information), este código interpreta la información del programa con letras mayúsculas y números (del A al F y del 0 al 9).

En el simulador de ROM se prueba y se verifica si las instrucciones del programa a simular son correctas, para esto se realizará un programa que interprete el archivo de tipo ASCII a binario (información representada en 0 y 1), el circuito simulador de ROM tiene la finalidad de simular a la EPROM del sistema bajo prueba. El programa a simular se almacena en el circuito simulador y una vez que se haya probado se transfiere la información al sistema bajo prueba (sistema con microprocesador).

El tercer bloque es el sistema que se está probando y contiene en ella el microprocesador y la memoria a simular, en este sistema se verifica si realmente la programación a simular es correcta y que todo el sistema funciona adecuadamente, si no es así se tendrá que corregir el error por programación y una vez que es corregido el programa se vuelve a probar hasta que se pueda grabar definitivamente en la memoria EPROM que se va a ocupar en el diseño para un proyecto específico.

1.3 REQUERIMIENTOS A CUBRIR Y CARACTERISTICAS QUE DEBE TENER EL CIRCUITO SIMULADOR DE ROM

Para plantear el diseño se analizó los requerimientos para facilitar la programación de los microprocesadores y hacer correcciones inmediatas en la programación de la EPROM, por lo que se consideró que se necesita de un circuito electrónico que simule a una memoria EPROM y que tenga las siguientes características y especificaciones:

1 El Simulador de memoria EPROM debe estar contenido en una tarjeta de circuito impreso para que se instale en el conector de expansión de 62 terminales de la PC tipo XT (computadora de tecnología extendida basado en el microprocesador 8088), o AT (computadora de tecnología avanzada basado en el microprocesador 80286), compatible con IBM.

2 El circuito deberá ser capaz de simular una memoria EPROM con capacidad de 8 Kbytes, y con la ventaja de modificar en ella la programación en caso de realizar alguna corrección.

3 El simulador de EPROM tomará información de un archivo tipo ASCII, la interpretará y la depositará en una localidad de memoria específica, contenida en la memoria de la tarjeta.

4 El circuito tendrá la finalidad de acelerar el proceso de desarrollo en los programas para un sistema, en el cual se utilicen microprocesadores. La tarjeta simulará el programa de una EPROM y se probará físicamente la operación del sistema con la prioridad de corregir rápidamente en caso de tener algún error y así anorrar tiempo en la programación de los microprocesadores y de las memorias EPROM.

5 La tarjeta de circuito impreso debe estar bajo las dimensiones que especifica ISA (Industrial Standard Architecture), para interfases estandar con la PC. (que debe estar dentro de las siguientes medidas: 33.02 cm de largo, 12 cm de altura y 1.2 cm de ancho).

1.4 ORGANIZACION DE LA COMPUTADORA

Las computadoras se han venido utilizando desde 1950. En un principio eran máquinas que ocupaban gran espacio y con alto costo, por lo que solamente las grandes empresas y el gobierno las utilizaban. El tamaño y la forma de la computadora ha cambiado gracias a un nuevo dispositivo denominado microprocesador.

El microprocesador es un circuito integrado (CI) que tiene gran capacidad de procesamiento en información. El microprocesador es de muy alta escala de integración (VLSI), que realiza las tareas de la unidad central de procesamiento de una computadora.

Una computadora contiene un microprocesador y como mínimo algún tipo de memoria semiconductor. En la actualidad la computadora constituye una herramienta muy útil aplicada en los diversos campos de la ciencia, la industria y sobre todo en el desarrollo de diseños y programación en proyectos de investigación.

Una computadora es una máquina electrónica digital, capaz de almacenar una gran cantidad de información y actuar de acuerdo con las instrucciones dadas. El sistema básico de la computadora consta de cinco unidades: la unidad de entrada, las unidades aritmética y de control (contenidas en la CPU, o unidad de procesamiento central), la unidad de memoria y la unidad de salida.

La unidad de memoria consta de dos tipos; memoria de lectura-escritura RAM (memoria de acceso aleatorio), este tipo de memoria es volátil es decir al quitarle el suministro de energía pierde toda la información. comunmente en este tipo de memoria se almacena información de datos temporales. El otro tipo de memoria denominado sólo de lectura (ROM) almacena información permanente que sólo puede leerse, tiene la característica de no borrar la información aun cuando se le quite el suministro de energía.

Esta organización es denominada arquitectura de la computadora. (ver la fig. 2).

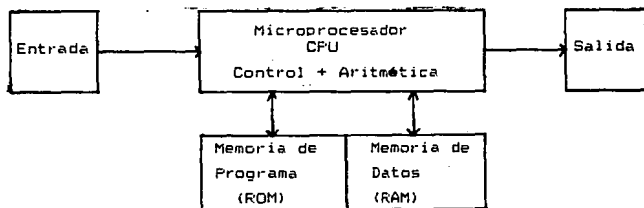


Fig. 2 Organización general de una computadora

La computadora funciona de la siguiente forma: los programas y los datos son introducidos en la unidad de procesamiento central CPU y transferidos a sus respectivas posiciones de memoria. La CPU lee la primera instrucción de la memoria del programa y la ejecuta. Las instrucciones pueden ser tan simples como sumar dos números, transferir un dato, sacar un dato, etc.

Cuando finaliza el manejo de los datos, los resultados son transferidos a la salida de la computadora, las funciones de la CPU son debidas a las instrucciones almacenadas en el programa de memoria.

En el aspecto funcional la máquina tiene tres componentes principales: memoria, procesador y dispositivos periféricos de entrada y salida. La memoria es el lugar donde se almacena la información e instrucciones. El procesador tiene la capacidad de interpretar las instrucciones que toma de la memoria. Los dispositivos de entrada y salida toman la información de elementos externos a la máquina y proporcionan resultados a periféricos fuera de la PC.

Los dispositivos de entrada son: el teclado de las terminales, las unidades de disco y las unidades de cinta; los de salida son la pantalla de las terminales, las impresoras, las unidades de disco y las de cinta. Además los conectores serie, paralelo y el conector de expansión de 62 terminales de la PC también son utilizados como dispositivo de entrada y salida.

La organización de la computadora esta constituida por un microprocesador en donde se encuentra la CPU. La computadora contiene las siguientes secciones básicas: la unidad de entrada, las unidades de control y aritmética que se encuentran en el microprocesador, además la unidad de memoria y la unidad de salida. Estas partes se interconectan en conjunto a través de las líneas de control, "bus" de datos y "bus" de direcciones: esta organización se muestra en la fig. 3.

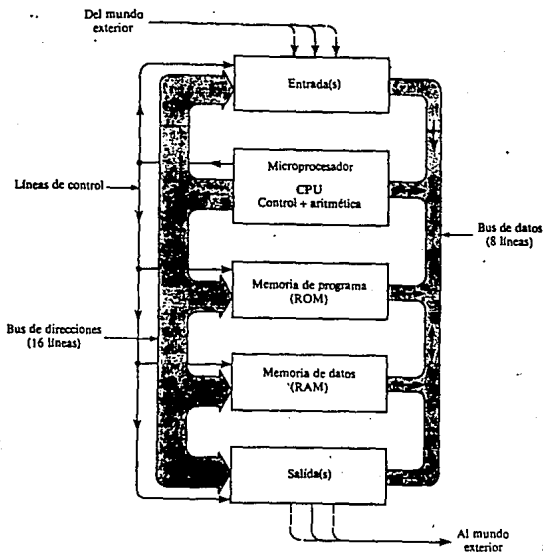


Fig. 3 Organización típica de una computadora

El bloque de la entrada recibe información del mundo exterior que es introducida a la computadora por medio del teclado de la PC y las unidades de disco.

En el segundo bloque se encuentra el microprocesador, en esta sección se controla todas las unidades del sistema utilizando las líneas de control.

El "bus" de direcciones (16 conductores en paralelo) seleccionan una cierta posición de memoria. El "bus" de datos (ocho conductores en paralelo) es un camino de doble dirección que se utiliza para introducir y sacar datos de la unidad microprocesadora.

La computadora contiene dos tipos de memoria: si un programa se va almacenar de forma permanente, la información se coloca en un dispositivo de memoria denominado memoria de sólo lectura ROM (tercer bloque). La ROM es una memoria programada permanentemente, en forma de CI.

En el cuarto bloque se representa la memoria RAM, en esta memoria se almacenan los datos de tipo temporal, es un dispositivo integrado denominado memoria de lectura/escritura, o memoria de acceso aleatorio.

La unidad de salida es el bloque que envía la información hacia el mundo exterior.

La computadora cuenta además con "conectores" de expansión que permiten establecer una interfase con la arquitectura de la PC (comunicación de la computadora con un dispositivo periférico) para desarrollar aplicaciones de manera general o específica.

1.5 PANORAMA DE LA EVOLUCION DEL MICROPROCESADOR Y LAS PC'S

El primer microprocesador fue desarrollado por Intel Corporation en 1971. Intel introdujo ese año el 4004 de 4 bits y el 8008 de 8 bits. En 1974 Intel introdujo el microprocesador 8080, (de 40 terminales), esta fue la segunda generación de microprocesadores, que maneja palabras de datos de 8 bits y tienen 16 líneas de dirección, además ofrece al usuario posibilidades de propósito general.

El microprocesador Intel 8085 (de 40 terminales), es una versión mejorada del 8080. Integra el reloj, control del sistema y prioridad en las interrupciones. Una tendencia en la evolución de los microprocesadores ha sido integrar más funciones en menos CI

En 1978 Intel desarrolló el 8086 y el 8088 (de 40 terminales), el microprocesador 8088 permite compatibilidad con dispositivos actuales de entrada y salida. La arquitectura del 8088 lo hizo que fuera uno de los microprocesadores más avanzados de su época, tenía tantas características potentes que IBM basó su primera generación de computadoras personales (PC) en el 8086. En este mismo tiempo apareció el CI 8087 (de 40 terminales), coprocesador matemático de números reales, dedicado a realizar procesos de datos numéricos a alta velocidad y cálculos matemáticos de alta precisión.

Las primeras computadoras personales pensadas para uso general se basaron en el microprocesador 8088 de 8 bits, comenzando así la innovación de las computadoras. Este microprocesador contiene la CPU que es la parte inteligente para controlar a la computadora, recibe o envía la información de datos en representaciones de 8 bits y su sistema operativo realiza una sola tarea en un momento determinado.

Otros microprocesadores de la generación 8080/8085 son los de Intel 80186, 80188, y 80286, (de 68 terminales) desarrollados en la década de los ochentas.

La aparición de computadoras manejadas por microprocesadores de 16 bits, dio la posibilidad de tener equipos más potentes. Las computadoras basadas en microprocesadores de 16 bits, como el 8086, tratan la información en multibits de 16 bits, en vez de 8, estos ofrecen una precisión mucho más alta, la aplicación del 8086 le permite a la computadora un juego de instrucciones más amplias siendo versátil, este microprocesador carga las instrucciones con mayor velocidad que el 8080, el procesador de 16 bits puede enviar o recibir un dato de 16 bits en una única transferencia, mientras que el procesador de 8 bits necesita para lo mismo dos operaciones distintas, los microprocesadores de 16 bits soportan varias tareas a la vez y son mejores que los de 8 bits.

El "software" para computadoras con microprocesadores de 16 bits, gana eficiencia por ser más fácil representar y operar grandes cantidades de números de 16 bits. El número de instrucciones que se utilizan pueden ser cuatro veces mayor que en uno de 8 bits. En estas computadoras se aumenta el poder operativo y se tiene un control más eficiente de casi todos los dispositivos periféricos de entrada/salida.

La cantidad de memoria que puede utilizar directamente una computadora de 16 bits es enorme comparada con la de 8 bits. El microprocesador de 16 bits puede acceder a un millón de bytes de memoria, mientras que uno de 8 bits sólo puede acceder directamente 64 Kbytes.

A mediados de 1984 Intel realizó un salto importante en el diseño de arquitectura con el desarrollo del microprocesador 80286. Esta CPU de nueva generación fue diseñada principalmente para aplicaciones que requerían altas prestaciones de velocidad en la transmisión de datos.

La computadora personal basado en el microprocesador 80286 de 16 bits, tiene las siguientes características: Mecanismos de protección y soporte a memoria muy amplia, su intervalo de velocidad esta entre 6 MHz y 12 MHz; en memoria RAM su intervalo es de 512 Kbytes como mínimo, hasta 1 Mbytes, en memoria ROM tiene como mínimo 64 Kbytes, con disco duro de 20 a 30 Mbytes, en disco flexible tiene desde 1.2 a 1.44 Mbytes, con 8 "buses" de expansión. Este tipo de procesador supera en calidad, función y velocidad a los microprocesadores 8080, 8088, y 8086.

El 80286 es compatible con el 8088/8086 debido a su conjunto común de modos de direccionamiento e instrucciones básicas, la arquitectura base soporta lenguajes de alto nivel como Pascal y "C", ya que el diseño del conjunto de registros esta bien adaptado al código generado por el compilador.

El 80286 soporta diferentes tipos de datos de alta capacidad y valores en forma exponencial. El diseño soporta un direccionamiento eficiente de estructuras complejas de datos, como arreglos, registros y arreglos de registros.

La arquitectura de memoria del 80286 soporta técnicas de programación que permite dividir la memoria en segmentos, proporciona un gran espacio de direcciones, tiene una memoria real que consta de 16 Mbytes de RAM. Este espacio permite al procesador guardar en memoria programas muy grandes y sus correspondientes estructuras de datos permiten acceder a una alta velocidad. El 80286 se diseñó para soportar aplicaciones de multiusuarios y multitarea en tiempo real, es decir realizar varias tareas simultáneamente en el mismo tiempo en una computadora.

El microprocesador que le sigue al 80286 tiene funciones más amplias y más rápidas en la transmisión de datos, este microprocesador es el 80386 (de 132 terminales).

El microprocesador 80386 es un procesador de 32 bits, trabaja con datos de 32 bits y puede acceder a 4 Gbytes de memoria, diseñado para soportar sistemas operativos optimizados para multitarea con registros de 32 bits, el 80386 soporta direcciones y tipos de datos de 32 bits, incluye mecanismos de protección para soportar sistemas operativos y "hardware" avanzados de multitarea.

Sus características son las siguientes: Su velocidad esta en el intervalo de 16 a 32 MHz, memoria RAM de 1 a 4 Mbytes, expandible a 16 Mbytes, con ROM de 1 a 2 Mbytes, disco flexible de 2 Mbytes y disco duro de 40 a 70 Mbytes, con 8 "conectores" de expansión; 1 de 32 bits, 6 de 16 bits y 1 de 8 bits.

Las modernas computadoras 486 están basadas en el microprocesador 80486 de 32 bits (con 16B terminales), que posee mecanismos que soportan estructuras de cálculo más potentes que los procesadores de 8 y 16 bits. Mientras que los procesadores de 8 y 16 bits realizan los cálculos en un sistema, los nuevos microprocesadores distribuyen las funciones a realizar en subfunciones que se ejecutan en CI especialmente diseñados para ello, como el coprocesador 80387.

Hoy en día hay CI especiales que realizan operaciones y funciones trigonométricas, e incluso funciones especiales de entrada y salida, sin necesidad del procesador central, son microprocesadores completamente dedicados a tareas especiales.

Las nuevas computadoras de microprocesadores de 32 bits, poseen algoritmos potentes, de detección y recuperación de error; es decir si hay algún error en el programa, será capaz de arreglarlo automáticamente, haciendo que los programas no se bloqueen y que el sistema no deje de funcionar. Los microprocesadores de 32 bits son tres veces más rápidos que los de 16 bits, la transferencia de información es tan rápida como los "buses" lo puedan transmitir (nanosegundos).

Estas computadoras son más potentes, aumentan la inteligencia de los dispositivos del procesador a una memoria del orden de Gbytes, estas computadoras tienen pantalla de color tridimensionales, que pueden acceder a redes de comunicación avanzada, permitiendo acceder bases de datos muy grandes.

1.6 EL PUERTO SERIE Y PARALELO

Todo proceso en el interior de una computadora existen dispositivos que posibilitan el diálogo entre la CPU y las unidades periféricas, a estos dispositivos se les llama puertos de entrada y salida. (input/output port)

Cada una de estos puertos está controlado por un circuito especial llamado controlador.

Los puertos de entrada y salida se usan para la comunicación entre la CPU y el mundo externo, así una computadora tendrá un controlador para cada dispositivo externo a la CPU. además la información que pasa a través de estos puertos de entrada y salida es controlada y estructurada por el mismo controlador, para así entregarla en condiciones óptimas a la CPU o a la unidad externa.

Se conoce como interfase o interconexión al dispositivo mediante el cual existe un diálogo entre la CPU y la unidad externa. (en nuestro caso el circuito simulador de ROM).

Las interfases prescindiendo de su grado de complejidad se dividen en interfases paralelas e interfases seriales.

El concepto de interfase serial o paralela se refiere a la conexión existente entre la puerta de entrada y salida y el periférico. Es decir, se refiere a la condición en que viajan los datos entre el controlador de interfase y el dispositivo externo. Las interfases más comunes son:

EIA RS232-C : interfase serial de uso general, para la conexión de impresoras, "plotters" (periféricos o equipos que efectúan graficas y dibujos), "modems", etc.

IEEE488 Interfase paralela de uso general principalmente para aparatos o instrumentos de medición.

Con la interfase serial los bits de información viajan uno después del otro y por la misma línea se envía toda la información, es decir en serie.

En una interfase serial los datos que la CPU envía llegan al controlador en forma paralela por el "bus" de datos, el controlador los memoriza y posteriormente a cada pulso de reloj los va transmitiendo uno tras otro. La información viaja a través de un sólo hilo.

De igual manera si el controlador recibe información de un periférico externo, estos datos le llegan en forma serial, posteriormente cuando la palabra o el byte se ha completado lo envía en forma paralela a la CPU por medio del "bus" de datos.

Con la interfase paralela todos los bits de información viajan simultáneamente hacia el periférico. En esta interfase, en cada pulso de reloj se envía un byte al periférico. De la misma manera el controlador de la interfase paralela recibe los datos en forma paralela de la unidad conectada al mismo. Este tipo de conexión es recomendable para distancias cortas y cuando se requiere de altas velocidades de transmisión de información.

1.7 LA TARJETA MADRE

La tarjeta madre es la parte principal de la PC donde se encuentra la unidad de procesamiento central del sistema (CPU), que realiza las funciones inteligentes de control, las funciones aritméticas y lógicas. Así como el adecuado funcionamiento de la transferencia de datos a los demás dispositivos del sistema para su óptimo funcionamiento. En ella se encuentran también memorias de almacenamiento de información, temporizadores, controladores y circuitos integrados (CI) que permiten la transferencia de entrada y salida de los datos.

La tarjeta madre contiene cinco conectores de expansión para una computadora PC tipo XT, que son utilizadas para la conexión de dispositivos periféricos. Una de estas no tiene salida al panel posterior de la máquina porque se destina para la tarjeta controladora del disco duro, cuenta además con cuatro bases en las que se puede instalar más memoria en circuitos integrados de 256 kbytes de RAM.

La tarjeta madre de la PC tipo AT mejora sus características ya que cuenta con un microprocesador más potente (microprocesador 80286). Cada uno de sus elementos tiene mayor capacidad para el procesamiento de la información; la transferencia de información es más rápida, por lo que hace más eficiente su funcionamiento. Además cuenta con ocho conectores de expansión que pueden ser usados para los dispositivos periféricos, seis de estas tienen dos bases, la primera base es de 62 terminales compatibles con XT y la otra base tiene 36 terminales para expandir la capacidad del sistema. Adicionalmente contiene dos bancos de bases para expandir la memoria RAM de 16 bits; cada CI utilizado es de 256 Kbyte de RAM.

1.8 EL CONECTOR DE EXPANSION

La computadora XT cuenta con 5 conectores de expansión y cada una de estas cuenta con 62 terminales por las cuales se realiza la transmisión de las señales periféricas hacia la CPU de la computadora, realizando el acceso directo de memoria (DMA).

Para el manejo de instrucciones, cuenta con líneas de datos, de direcciones y de control.

El conector de expansión cuenta con:

- 8 líneas del "bus" de datos, bidireccionales
- 20 líneas de direcciones
- 6 niveles de interrupción
- 3 canales para líneas de control DMA
- Un canal para pruebas de línea
- 4 líneas de polarización de voltaje directo
- Líneas de control para lectura de memoria
- Líneas para entrada / salida de lectura o escritura.

Ver la tabla 1 (Terminales del conector de expansión)

LADO DE SOLDADURA Nombre de la señal	Terminales		LADO DE COMPONENTES Nombre de la señal
GND	B1	A1	-I/O CH CK
+RESET DRV	B2	A2	+D7
+5V	B3	A3	+D6
+IRQ2	B4	A4	+D5
-5v	B5	A5	+D4
+DRQ2	B6	A6	+D3
-12V	B7	A7	+D2
NO USADO..	B8	A8	+D1
+12V	B9	A9	+D0
GND	B10	A10	+I/O CH RDY
-MEMW	B11	A11	+AEN
-MEMR	B12	A12	+A19
-IOW	B13	A13	+A18
-IOR	B14	A14	+A17
-DACK3	B15	A15	+A16
+DRQ3	B16	A16	+A15
-DACK1	B17	A17	+A14
+DRQ1	B18	A18	+A13
-DACK0	B19	A19	+A12
CLOCK	B20	A20	+A11
+IRQ7	B21	A21	+A10
+IRQ6	B22	A22	+A9
+IRQ5	B23	A23	+A8
+IRQ4	B24	A24	+A7
+IRQ3	B25	A25	+A6
-DACK2	B26	A26	+A5
+T/C	B27	A27	+A4
+ALE	B28	A28	+A3
+5V	B29	A29	+A2
+OSC	B30	A30	+A1
GND	B31	A31	+A0

Tabla 1 Terminales de entrada y salida del conector de expansión

Todas las señales manejan voltajes de 0 a 5 volts típico en TTL (Lógica transistor transistor).

En la tabla 1 se muestran las señales disponibles para realizar la interfase con la PC, conteniendo las líneas de dirección, de datos y de control. El conector de expansión cuenta con dos líneas de 31 señales cada una, la tabla muestra las señales que contiene en el lado de soldadura y en el lado donde van los componentes para una tarjeta que realiza la interfase con la PC. Las terminales son comentadas en 1.9.

1.9 DESCRIPCION DE LAS TERMINALES DEL CONECTOR DE EXPANSION

Señal D0 a D7 que se encuentra en las terminales A9-A2 :

Son ocho líneas que constituyen el "bus" de datos bidireccional de 8 "bits" para la transferencia de datos, comandos e información entre el microprocesador 8088, memoria y puertos. Para una computadora XT, D0 es el bit menos significativo (LSB) y D7 el bit más significativo (MSB). Estas líneas son activas en uno lógico.

Señal A0 a A19 que se encuentra en las terminales A31-A12 :

Son las líneas del "bus" de direcciones de los "bits" 0 al 19 que se utilizan para direccionar la memoria y los dispositivos de entrada/salida en el sistema, estas líneas pueden ser generadas o manejadas por el controlador de DMA (Direct Memory Access) o por el microprocesador 8088. Son activas en uno lógico.

Señal -I/O CH CK que se encuentra en la terminal A1 :

(I/O Channel Check): Es la señal generada por una tarjeta externa en el conector de expansión que informa a la CPU de una falla. esta línea provee al microprocesador una condición de error de paridad sobre la memoria o dispositivos instalados en el conector de expansión. cuando es activada en cero lógico un error ha ocurrido.

Señal +I/O CH RDY que se encuentra en la terminal A10 :

(I/O Channel Ready) Esta es una señal de entrada usada para extender la longitud del ciclo del "bus" para que la memoria o algún dispositivo de entrada o salida más lentos de lo que se requiere pueda responder al ciclo normal de cuatro pulsos de reloj del sistema.

Señal de RESET DRV que se encuentra en la terminal B2 :

Esta señal es usada para restablecer o inicializar el sistema lógico en el encendido de la máquina o durante una falla de suministro de energía. esta activo en uno lógico.

Señal IRQ 2-7 que se encuentran en la terminal B4, B25-21 :

(Interrupt Request Channels) Estas señales de entrada son canales de petición de interrupción para el microprocesador 8088 provenientes del "bus" del sistema. Estas señales van directamente al circuito manejador de interrupciones B259 para interrumpir el control del proceso de la tarjeta. Estas líneas son usadas como señales al microprocesador indicando que un dispositivo de E/S requiere atención.

Señal -MEMW que se encuentra en la terminal B11 :

(Memory Write) Señal de salida baja usado para escribir datos del "bus" del sistema a la memoria. Esta señal es controlada por el 8288 (XT) y el CPU la habilita en bajo cuando realiza una escritura en la memoria.

Señal -MEMR que se encuentra en la terminal B12 :

(Memory Read) Es la señal de salida usada para leer los datos en memoria. esta señal es controlada por el 8288. Señal que el CPU habilita en bajo para realizar una lectura a memoria.

Señal -IOW que se encuentra en la terminal B13 :

(I/O Write) Esta señal es de salida y controlada por el 8288. El CPU activa en bajo esta señal para realizar una escritura hacia un dispositivo periférico del sistema.

Señal -IOR que se encuentra en la terminal B14 :

(I/O Read) Señal de salida del 8288 utilizada para indicar las entradas o salidas de los puertos. El CPU activa en bajo esta señal para realizar una lectura hacia un dispositivo periférico del sistema.

Señal +DRQ 1-3 que se encuentran en la terminal B18, B6, B16 :

DRQ1 a DRQ3 (DMA requests) Estas líneas de entrada son canales asíncronos y de petición. Se usan por los dispositivos periféricos para obtener servicio del DMA.

Señal -DACK 0-3, que se encuentran en la terminal B19, B17, B26, B15 :

DACK0 a DACK3 (DMA acknowledge signals) Son señales activas de reconocimiento del DMA, notifican a dispositivos periféricos individuales cuando se ha otorgado un servicio del DMA y refresca la memoria dinámica.

Señal CLOCK que se encuentra en la terminal B20 :

Reloj del sistema con una frecuencia de 4.77 MHz, la cual es una tercera parte de la frecuencia del oscilador. Esta es la frecuencia de operación del microprocesador 8088, este tiene un período de 210 nanosegundos y un ciclo de trabajo del 33% producido por el generador de reloj.

señal +T/C que se encuentra en la terminal B27 :

(Terminal/count) Esta línea provee un pulso cuando se presenta la terminación de un servicio del DMA.

Señal +ALE que se encuentra en la terminal B28 :

(adres latch enable) Esta es una señal de salida proporcionada por el controlador del "bus" 8286. Indica que el "bus" de direcciones es válido para el comienzo de un ciclo. Esta señal esta en activo alto justo antes de que el "bus" de direcciones sea válido y cae a nivel bajo justo después que el "bus" de direcciones ha sido válido.

Señal AEN que se encuentra en la terminal A11 :

(Addres enable) Esta señal es usada por el controlador DMA, inhibe al controlador del "bus" 8288 y a los acondicionadores de las direcciones para permitir que el DMA asuma el control del "BUS".

La señal de AEN puede ser utilizada para habilitar a otros dispositivos de E/S, indicando que una dirección ha sido generada y es válida cuando está activa en uno lógico. El controlador DMA tiene el control del "bus" de direcciones, "bus" de datos y los comandos de lectura y escritura (memoria y dispositivos de E/S).

Señal de OSC que se encuentra en la terminal B30 :

Señal de salida cuya frecuencia es equivalente a 14.31818 MHz, con un período de 70 nanosegundos y utilizado para la circuitería de video.

El conector de expansión cuenta con fuentes de alimentación proporcionadas en las líneas B1 y B31 (tierras), los voltajes son suministrados en las siguientes líneas: + 5 VDC en terminal B3 y B29, - 5 VDC en la terminal B5, + 12 VDC en la terminal B9 y - 12 V DC en la terminal B7.

1.10 MAPA DE DIRECCIONAMIENTO DE LA MEMORIA DE LA PC PARA EL CONECTOR DE EXPANSION

La computadora Personal (PC) contiene un mapa de memoria y de direccionamientos para el conector de expansión, dividido de la siguiente forma: el mapa de direccionamiento del puerto de entradas y salidas para el conector de expansión esta formado por una primera parte con 512 puertos direccionables contenida entre 0000H al 01FFF en hexadecimal, esta es asignada para el sistema principal de la tarjeta (sistema madre de la tarjeta). En esta parte reside el sistema base de la tarjeta. Estas direcciones del puerto son usados para direccionar la CPU del 8088 y el soporte de dispositivos que integran a la tarjeta madre. La segunda parte es el espacio de las direcciones de 0200H al 03FF con 512 puertos direccionables que estan disponibles en la tarjeta para los 5 conectores de expansión; este espacio de dirección es usado para la decodificación de direcciones del sistema.

En el mapa de memoria de la PC están designadas regiones especiales para una función específica del sistema y regiones disponibles para la interfase con otros sistemas. (Ver la tabla 3).

A continuación se muestran el mapeo de las direcciones asignadas para cada elemento y el espacio disponible con todas las características del conector de expansión. (Ver la tabla 2 y 3). Estas distribuciones es necesario conocerlas para utilizar las direcciones adecuadas en las adliaciones específicas.

INTERVALO HEXADECIMAL	APLICACION
000- 00F	Controlador de acceso directo a memoria
020- 021	Circuito DMA 8237A -5
040- 043	Control de interrupciones programables
060- 063	Circuito 8259A
080- 083	Temporizador 8253-5
0A0- 0A3	Controlador de Interfase Programable
0C0- 0C3	8255A-5
0E0- 0E3	Registro de interrupción no mascarable
0A4- 0A7	Reservado
0C4- 0C7	Reservado
0E4- 0E7	Reservado
100- 1FF	No se utiliza
200- 20F	Control de juegos
210- 217	Unidad de expansión
220- 24F	Reservado
278- 27F	Reservado
2F0- 2F7	Reservado
2F8- 3FF	Comunicación asíncrona (2)
300- 31F	Tarjeta prototipo
320- 32F	Disco duro
378- 37F	Impresora
380- 38C	Comunicación SDLG
380- 389	Comunicación síncrona binaria (2)
3A0- 3A9	Comunicación síncrona binaria (1)
3B0- 3BF	Monitor monocromático e impresora
3C0- 3CF	Reservado
3D0- 3DF	Gráficas a color
3E0- 3F7	Reservado
3F0- 3F7	Disco
3F8- 3FF	Comunicación asíncrona (1)

Tabla 2 Mapeo de direccionamiento del conector de expansión de la PC XT.

00000	16K x 9	RAM EN LA UNIDAD DEL SISTEMA BASE
	16K x 9	
	16K x 9	
	16K x 9	
10000	576 KB	EXPANSION DE MEMORIA RAM
9FFFF A0000	NO USADO	64 KB.
B0000	28KB	128 KB RESERVADOS PARA EL BUFFER DE PANTALLA.
B7FFF B8000	16KB	
BCFFF	16KB	
BC000	NO USADO	
BD000	RESERVADO PARA ROM 192 KB.	
F0000	8KB	CONECTOR PARA ROM
	8KB	
	8K ROM	
	8K ROM	
	8K ROM	
	8K ROM	
	8K ROM	INTERPRETE BASICO
FFFFF	16 Bytes	BIOS ROM AREA RESERVADA PARA EL SOBB, PRUEBAS DE AUTOENCENDIDO.

Tabla 3 Mapa de Memoria de la IBM PC.

C A P I T U L O 2

DISEÑO DEL CIRCUITO SIMULADOR DE ROM

En este capítulo se encuentra el desarrollo del diseño para el circuito simulador de ROM. Se analiza las alternativas de solución y la selección de los componentes. Además el diseño de los elementos que integran al circuito ("hardware") y del programa que gobierna y hace funcionar adecuadamente al circuito ("software").

2.1 ESTRUCTURA DEL SISTEMA

Para realizar el diseño del circuito Simulador de ROM se consideran brevemente varias etapas, estructurándose por bloques y analizando e integrando los componentes hasta obtener el diseño final. Estas etapas se presentan en la fig.4.



Fig 4 Etapas que integran la estructura del circuito simulador de ROM.

Para desarrollar el diseño del circuito simulador, previamente se tuvo que entender el objetivo del circuito, el requerimiento que debe cumplir el simulador para agilizar la programación de memorias EPROM, para sistemas que utilizan microprocesadores, así como los criterios que debe cumplir el circuito.

Ya se mencionaron las señales que tiene el conector de expansión, el cual nos proporciona las señales de control, datos y direccionamiento de la PC para realizar la interfase. Así como el mapa de memoria disponible para realizar la intercomunicación (líneas de entrada y salida), para poder trabajar con la tarjeta del circuito simulador de ROM.

La estructura del sistema se basa considerando las etapas de la fig. 4. Para tener una visión de la estructura general del sistema simulador de ROM y realizar la interfase con la PC se presenta el siguiente diagrama a bloques:

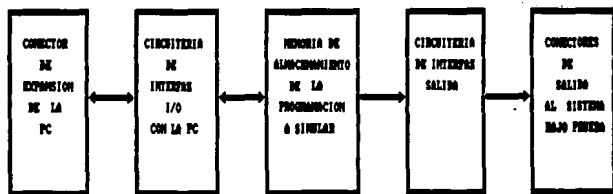


fig. 5 Diagrama a bloques de la estructura general del sistema simulador de ROM.

A continuación se analiza el diseño del circuito tanto del hardware (distribución de elementos físicos), como el diseño del software (programa que gobierna al circuito).

2.2 DISEÑO DEL CIRCUITO

El diseño del circuito se dividió en dos secciones la parte del "hardware", que corresponde a la circuitería de la tarjeta (elementos físicos) y el diseño del "software" que corresponde a la programación de la tarjeta. Estas dos secciones del diseño son dependientes, ya que si una de las dos tiene algún error la tarjeta simulador de ROM no funcionará.

2.2.1 DISEÑO DEL HARDWARE

Para realizar el diseño del "hardware" se tiene que conocer las señales que nos proporciona el conector de expansión de la PC, así como su mapa de memoria; con el fin de localizar aquellas líneas de direccionamiento que se van a ocupar para la tarjeta. Esta información se analizó en el capítulo anterior.

Las Señales útiles provenientes del conector de expansión de la PC son las líneas que constituyen el "bus" de datos bidireccional (que se encuentran en las terminales A2-A7), las líneas que constituyen al "bus" de direcciones (que se encuentran en las terminales A12-A31) que se utilizan para direccionar la memoria y dispositivos de entrada y salida. Además de las líneas de control AEN, MEMW, MEMR, (que se encuentran en la terminal A11, B11, y B12 respectivamente), y las líneas que proporcionan la fuente de alimentación GND y el voltaje de + 5 V. (que se encuentran en la terminal B1, B31 para GND y B3, B29 para el voltaje de + 5 V.)

Para realizar el diseño del "Hardware" se consideran alternativas de solución y selección de componentes los cuales se especifican a continuación.

2.2.1.1 ALTERNATIVAS DE SOLUCION

De las alternativas para escoger los elementos para realizar la tarjeta simulador de ROM se tomará aquella que cubra con los requerimientos dados en la sección 1.3, de acuerdo a la disponibilidad de materiales adquiridos fácilmente en el mercado, así como la facilidad de manejo, versatilidad, bajo costo y utilizando los elementos mínimos posibles.

Para lograr la comunicación entre la computadora y el circuito simulador de ROM, hay que interconectar el "bus" de datos, de direcciones y de control.

Por lo que se requiere de dispositivos que interactuen y faciliten la comunicación con las señales del conector de expansión de la computadora. Se requiere de dispositivos retenedores (latches) que permitan capturar y tomar datos para ser reconocidos tanto por el Simulador como para la PC. Así como de un dispositivo para direccionar la memoria, de una memoria donde almacenar la programación a simular y de conectores para unir el simulador de ROM con el sistema bajo prueba.

Las alternativas de los elementos que por sus características se pueden utilizar se muestran a continuación de acuerdo para cada aplicación.

___ Para la circuitería de interfaz tanto para la entrada como la salida podemos utilizar los siguientes elementos.

CI 74LS373 Memoria temporal tipo D octal ("latches") es un dispositivo retenedor, la salida sigue a la entrada cuando su terminal de control está en bajo y la terminal de habilitación está en alto.

CI 74LS244 Acondicionador (buffer) dispositivo de interfaz cuya función es la de aislar, así como proporcionar una mayor capacidad de manejo de corriente, sin ejercer ninguna función lógica sobre la misma. El buffer tiene una entrada de habilitación activa en baja y el dato no se invierte cuando pasa a través del buffer. Cuando el buffer del bus está inhabilitado, la salida está flotando y no tiene efecto en la línea del bus.

En la condición de inhabilitación la salida de la puerta no absorbe ni da ninguna corriente a la línea del bus, esta aislado por lo que se dice que el buffer del bus tiene salida de tres estados.

CI 74LS245 Bus de transferencia que nos permite la comunicación bidireccional.

CI PPI 8255 dispositivo programable de entrada-salida de propósito general.

Para almacenar el programa a simular se debe usar una memoria temporal tipo RAM por que sólo se guardará la información en el momento de simular el programa para los microprocesadores por lo que podemos utilizar alguna de las siguientes memorias:

Memoria 2114A RAM estática de INTEL de 4096 bits organizada como 1024 palabras de 4 bits con tecnología HMOS. El acceso de datos no necesita esperar los tiempos de preparación de direcciones. interfaz sencillo y compatible con TTL.

memoria MCM6064 RAM estática organizada de 8192 palabras de 8 bits, con tecnología CMOS, comoatible con TTL y con un tiempo de acceso de 100 ns máximo.

Memoria MCM6164 RAM estática organizada por 8192 palabras de 8 bits, tecnología CMOS, compatible con TTL y con un tiempo de acceso de 45 a 55 ns máximo.

Memoria MCM6264 RAM estática de alta velocidad de 8192 palabras de 8 bits, tecnología CMOS, compatible con TTL y con un tiempo de acceso rápido de 35 a 45 ns máximo.

Memoria NM100496 RAM estática organizada de 16,384 palabras de 4 bits tecnología CMOS y compatible con TTL.

Para la decodificación del espacio de direcciones a ocupar podemos utilizar dos CI 74LS85 que son comparadores de 4 bits, ó un CI 74LS688 comparador de 8 bits.

Los conectores más apropiados para la transmisión de datos son los del tipo DB-25 (conectores para un cable plano con el mismo tipo de terminal).

2.2.1.2 SELECCION DE COMPONENTES.

Para la selección de los componentes se toma en cuenta los siguientes parámetros:

- a) El consumo de potencia
- b) La compatibilidad con TTL (lógica transistor transistor)
- c) Que contenga un "bus" de datos de 8 bit
- d) Aplicación para interfases
- e) Configuración ideal en terminales de entrada-salida
- f) Costo bajo.

El cuadro comparativo se encuentra en la tabla 4.

Parámetros	CI	CI	CI	CI
	74LS373	74LS244	74LS245	PP18255
Consumo de potencia bajo	X	X	X	
Compatibilidad con TTL	X	X	X	X
Líneas de protocolo	X	X	X	X
"Bus" de datos de 8 bits	X	X	X	X
Aplicación para interfase con la tarjeta de la PC	X	X	X	X
Comunicación asíncrona bidireccional entre "bus"			X	X
Costo bajo	X	X	X	X
Configuración ideal en terminales de entrada-salida.	X		X	X

Tabla 4 Cuadro de comparación para los dispositivos de interfase de entrada/salida.

Como se puede observar del cuadro comparativo se llega a la conclusión que la mejor alternativa para realizar la circuitería de interfaz (segundo bloque de la fig. 5) con la PC es el CI 74LS245, porque facilita la comunicación asíncrona bidireccional y es aplicable para la interfase con el conector de expansión de la PC, además de su bajo costo y la configuración ideal en sus terminales de entrada y salida.

Se deberá utilizar además una memoria RAM (memoria de acceso aleatorio) donde se deposita el programa que se quiere probar y simular, para escoger la memoria más adecuada se presenta el siguiente cuadro de comparación de memorias RAM estáticas para analizar las características de cada una de ellas.

Parámetros	CI	CI	CI	CI	CI
	2114	MCM6064	MCM6164	MCM6264	NM100496
Consumo de potencia bajo	X	X	X	X	X
Compatibilidad TTL	X	X	X	X	X
Bus de datos de 8 bits		X	X	X	
Capacidad de memoria en Kbytes	5	8	8	8	8
Tiempo de acceso en ns.	75	100	45	35	20
No requiere reloj para la habilitación de señales de control.	X	X	X	X	X
Costo bajo	X	X	X	X	X

Tabla 5 Cuadro comparativo de Memorias RAM estáticas

De acuerdo a la tabla 5 comparación de las memorias RAM estáticas se observa que la mejor opción es la RAM MCM 6264 porque es una memoria de alta velocidad con capacidad de 8 Kbytes, con un "bus" de datos de 8 bits con un tiempo de acceso a memoria de 35 ns. compatible con TTL y además de que su valor es de bajo costo.

Para simular a la memoria ROM y tomar la información de un archivo tipo ASCII, interpretarla y depositarla a una localidad de memoria del simulador es necesario comunicarse con la PC, esto se hace a través del conector de expansión de la PC y las pistas de la tarjeta con lo cual quedan enlazados los "buses" de datos, direcciones y de control.

2.2.1.3 BASES DE DISEÑO

Para construir la tarjeta simulador de ROM se considerarán las siguientes bases de diseño:

- a) Realizar el diagrama esquemático del circuito, construir el prototipo en tarjeta protoboard y hacer las pruebas en éste, antes de realizar definitivamente la tarjeta de circuito impreso.
- b) Utilizar los elementos más apropiados considerando su potencia, compatibilidad con TTL y sobre todo el bajo costo. Para escoger la mejor alternativa se tomaron los criterios de la tabla 4 y tabla 5.
- c) Se requiere tener un consumo de energía para la tarjeta no mayor a 2 W y a 400 mA, para no sobrepasar las características eléctricas de las fuentes de alimentación de la computadora.
- d) Se debe distribuir los componentes cuidando su estética y la distancia más corta en las rutas, pistas ó venas de intercomunicación de todas las partes que la integran.
- e) Se usará un ancho mayor de ruta en las líneas de tierra y alimentación para que soporten el paso de la corriente.

f) El diseño del circuito definitivo estará en una tarjeta de circuito impreso de dos caras el cual está en el intervalo de dimensiones que especifica ISA. (Para mayor información ver el apéndice E).

2.2.1.4 DIAGRAMA A BLOQUES PARA EL DISEÑO DEL CIRCUITO PROPUESTO

Para llevar a cabo el diseño de la circuitería del simulador de ROM se propone el siguiente diagrama a bloques, el cual presenta el esquema general del circuito simulador de ROM. Ver fig. 6.

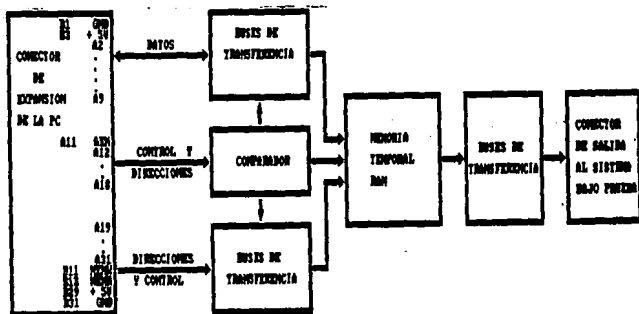


Fig. 6 Diagrama a bloques en forma general del circuito simulador de ROM propuesto.

A continuación se explica cada uno de los bloques de la figura 6.

En el primer bloque se presentan las señales del conector de expansión que proporciona la PC para realizar la comunicación con la tarjeta simulador de ROM. Las señales útiles que ocupamos son el "bus de datos" (A2 - A9) los cuales constituyen 8 líneas de conexión bidireccional realizando la interfaz con el CI 74LS245 bus de transferencia. Del "bus de direcciones" (que se encuentran en las terminales del A12 - A31) las señales (A12 - A18) son los más significativos y se utilizan para direccionar la memoria del microprocesador de la PC para establecer el intervalo de memoria disponibles para la tarjeta simulador de ROM.

Las señales que se encuentran en las terminales A19 a la A31 son 12 líneas de direcciones. Las señales de control AEN, MEMW y MEMR (que se encuentran en las terminales A11, B11 y B12). así como las señales que proporcionan la fuente de alimentación GND y el voltaje de +5 V. (que se encuentran en las terminales B1, B31 y B3, B29 respectivamente) son las señales que ocupamos para llevar a cabo la interfaz con la PC. Todas estas señales son útiles para la circuitería de interfaz los cuales se interconectan con los CI 74LS245 buses de transferencia y el CI 74LS688 comparador de 8 bits, como se observa en la fig. 6.

Para simular a la memoria ROM y tomar la información de un archivo tipo ASCII, interpretarla y depositarla a una localidad de memoria del simulador es necesario comunicar los "buses" de datos, direcciones y de control. Para esto utilizamos "buses" de transferencia, para transferir las señales de la PC a la tarjeta simulador de ROM; estos se encuentran en el circuito integrado 74LS245.

Para la transferencia de las señales se utilizan tres circuitos integrados 74LS245 Buffers de transferencia bidireccional para la circuitería de entrada y otros tres para la circuitería de salida.

Los bloques de los buses de transferencia (circuito integrado 74LS245) de la fig. 6 son dispositivos retenedores (latches) reconocidos por el simulador y por el CPU de la PC. Este dispositivo mejora y reduce el ruido de histéresis en el "bus" de entrada. Este circuito integrado nos permite la comunicación de información del programa que provienen del "bus" de datos, el "bus" de direcciones y de control de la PC. A través del "bus" de transferencia fluye la información del programa ya interpretado en forma binaria, almacenándose esta información en la memoria RAM MCM6264, donde posteriormente se transfiere hacia el sistema bajo prueba.

El circuito integrado 74LS245 bus de transferencia tiene dos modos de comunicación en los buses de datos, la transmisión de los datos puede ser del "bus" A al B o viceversa, dependiendo de la terminal de control "G" y de la terminal de habilitación DIR. El bus de datos del conector de expansión se conecta al bus de datos de la memoria RAM MCM6264, pasando previamente por el circuito integrado 74LS245 bus de transferencia bidireccional.

Para la decodificación del intervalo de direcciones que se va a ocupar en el mapa de memoria del microprocesador de la PC, se utiliza el circuito integrado 74LS688 comparador de 8 bits.

Las direcciones AA13 al AA19 que se encuentran en las terminales A12 al A18 del conector de la PC están conectados a las terminales de entrada "Q" del comparador, por la entrada "P" del comparador se conecta un DIP switch de 8 posiciones en paralelo con resistencias de 1 kohms (arreglo resistivo).

Cuando el valor en los interruptores del DIP switch es igual al valor del bus de direcciones provenientes del conector de la PC y cuando el control AEN le suministra una señal activo bajo, el comparador manda en la terminal de salida una señal activo bajo, el comparador activa a su vez otros dispositivos como la memoria RAM MCM6264 y los buses de transferencia de entrada por lo que este elemento controla la transferencia de datos y la lectura o escritura del programa en la RAM del circuito.

En la figura 7 se muestra la decodificación utilizada para el circuito simulador de ROM.

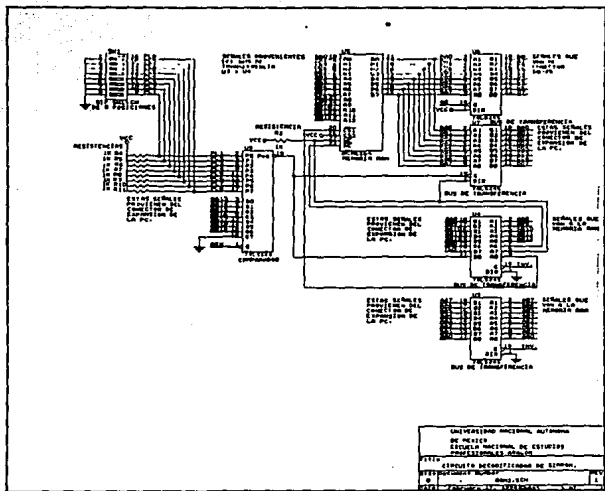


Fig. 7 Circuito decodificador para el simulador de ROM.

Las posiciones del DIP switch de 8 posiciones deben estar bien direccionadas con el valor AB H (hexadecimal) para establecer adecuadamente la comunicación con la PC, el cual junto con el arreglo resistivo define el nivel lógico de ceros y unos en las terminales de entrada del comparador con el fin de fijar la dirección adecuada y realizar la transferencia de los datos correctos, ya que en caso contrario puede causar daños irreversibles a la PC. Para las posiciones del DIP switch ver la fig. 8.



Fig. 8 Direccionamiento del DIP switch de 8 posiciones.

Se requiere además una memoria temporal para guardar la información a simular (programa para el microprocesador bajo prueba), de acuerdo a las alternativas y cuadro de comparación de la tabla 5 se escogió una memoria RAM estática MCM6264 de alta velocidad con capacidad de 8 Kbytes de memoria para almacenar las instrucciones del programa de la ROM o EPROM y así poder simular esta memoria.

También se utilizó los siguientes circuitos integrados: 74LS00 (NAND), 74LS32 (OR), y 74LS14 (inversor), para designar los pulsos de control en los CI de transferencia.

Se utilizó un interruptor de un polo 2 tiros marca AUGAT, un conector DB-25 hembra marca PRODEL, un conector macho E.I.S (sistema económico de interconexión) marca AMP de 5 terminales, un metro de cable plano de 30 hilos marca PRODEL, un diodo emisor de luz (LED) y una base cold de 24 terminales. Todos estos elementos se utilizan para realizar la tarjeta simulador de ROM, simular la memoria ROM o EPROM y transferir esta información al sistema bajo prueba.

La fig. 9 muestra el diagrama a bloques del circuito simulador de ROM en una forma más detallada, la relación que tiene cada elemento y su interconexión con todo el sistema, permitiéndonos entender la transmisión de las señales del "bus" de datos, de direcciones y de control de cada elemento, hacia las señales que existe en el conector de la PC.

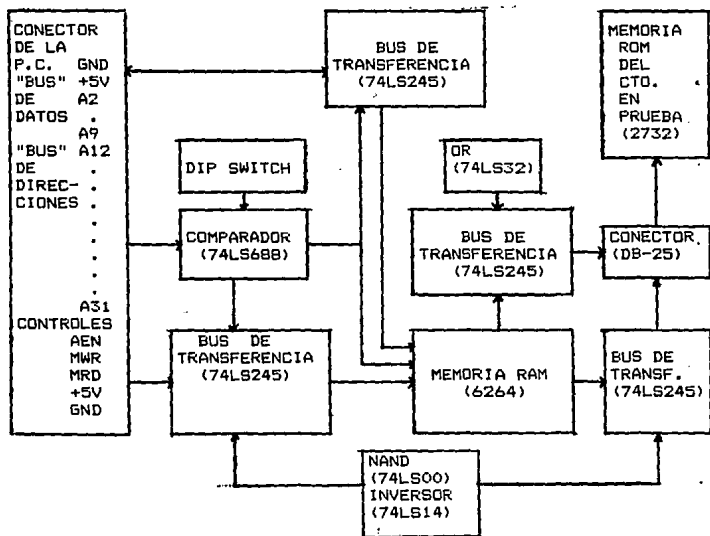


Fig. 9 Diagrama a bloques, más detallado del circuito simulador de ROM.

Para transferir las instrucciones del programa a simular al sistema bajo prueba se utiliza el conector hembra DB-25 y el cable plano de 30 hilos. Como el LED (diodo emisor de luz) indicador y el interruptor para direccionar las señales (direccionar la señal de la PC hacia tarjeta simulador o de la tarjeta simulador hacia el sistema bajo prueba) se encuentra en la tarjeta. se tuvo que trasladar estas señales hacia una caja de aluminio para mayor comodidad, poder controlar y visualizar apartir de este. Para esto se utiliza un conector EIS (sistema económico de interrupción) de 5 terminales por donde se trasladan estas señales.

En la fig. 10 se muestra el diagrama a bloques del sistema simulador de ROM, el LED indicador y el interruptor para direccionar el sentido de la transferencia de las señales.

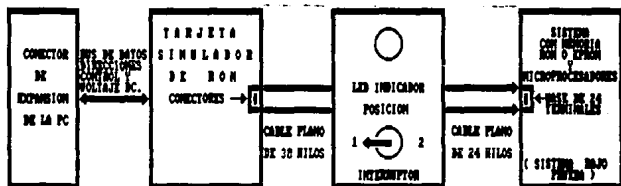


Fig. 10 Diagrama abloques del sistema simulador de ROM, el LED indicador y el interruptor que determina la dirección de las señales.

Se utiliza un interruptor de un polo 2 tiros para operar el simulador. En la posición 1 del interruptor (encendido el LED indicador) la tarjeta esta preparada para cargar el programa a simular y la transferencia de información es de la PC a la tarjeta simulador de ROM.

En la posición 2 del interruptor (apagado el LED indicador) la transferencia de información es de la tarjeta simulador de ROM hacia la base de 24 terminales donde se simula y se encuentra la memoria ROM o EPROM (sistema bajo prueba) ver fig. 10. Debe tomarse en cuenta que se tendrá que cargar previamente el programa del simulador de ROM, dicho procedimiento se comenta más adelante.

El diseño del circuito simulador de ROM es un emulador de EPROM que realiza las funciones de la EPROM dentro de un circuito bajo prueba de modo que no se requiere programar ningún componente (memoria ROM o EPROM) hasta estar seguro que el circuito bajo prueba funciona adecuadamente con las instrucciones del programa de esta memoria.

La información se deposita en el simulador de ROM (Emulador) desde la PC. Al contar con un simulador de ROM se ahorra tiempo en el desarrollo de programas para sistemas mínimos en el cual utilicen memorias ROM, EPROM y microprocesadores.

El simulador se puede describir como una memoria de doble puerto en donde la PC es uno de los puertos y el sistema bajo prueba es el otro. En esta memoria la PC puede leer y escribir datos mientras que el sistema bajo prueba sólo puede leer.

El conector de expansión para una computadora XT y una AT proporciona cuatro niveles de voltaje en corriente directa. La Tabla 6 proporciona la información de las características máximas y mínimas de voltaje, la corriente, la potencia y las terminales en donde se encuentran.

No. de terminal	(V dc)	Voltaje (Máx.)	Voltaje (Mín.)	Corriente (A)	Potencia (W)
B3 y B29	+ 5	5.25	4.80	7.0	35.0
B1 y B31	- 5	5.50	4.60	0.3	1.5
B9	+12	12.60	11.52	2.0	24.0
E7	-12	13.2	10.92	0.25	3.0

Tabla 6 Voltajes proporcionados por el conector de expansión

Para la polarización de la tarjeta del circuito simulador se utiliza el voltaje de + 5 V, ya que los circuitos integrados que constituyen el simulador requieren este voltaje de alimentación de acuerdo a las características eléctricas del fabricante.

Para el diseño del circuito y la realización del diagrama esquemático se utilizó el paquete Orcad, este paquete es muy útil para realizar diagramas eléctricos, ya que cuenta con librerías e instrucciones que permiten realizar el diagrama esquemático de los dispositivos eléctricos como resistencias, diodos, capacitores, memorias, microprocesadores etc. Además tiene comandos para transferir los elementos a otro archivo donde facilita la realización del diagrama de circuito impreso.

La fig. 11 muestra el diagrama electrónico del circuito simulador de ROM.

2.2.1.5 CONDICIONES DE OPERACION DE LOS ELEMENTOS

A continuación se describen las condiciones de operación para cada uno de los elementos que integran el circuito simulador de ROM, los modos de control para que la RAM opere en forma de escritura, ver la Tabla 7, y para lectura ver la Tabla 8.

CIRCUITO INTEGRADO	CONTROL	ENTRADA	SALIDA
U5 MEMORIA RAM MCM6264	$\overline{CS1}$ CS2 \overline{OE} \overline{W} L H X L	ENTRADA DE DATOS	
U9 COMPARADOR 74LS688	G L	ENTRADAS P = 0	L
BUS DE TRANSFERENCIA 74LS245 U3, U4 Y U7 U1 Y U2 U6	G DIR L L L H H X	TRANSMISION DE DATOS DEL BUS B AL A. DEL BUS A AL B. AISLADO	

Tabla 7 Condiciones de operación en los controles de cada elemento para escribir en la RAM (grabar datos).

CIRCUITOS INTEGRADOS	CONTROL	ENTRADA	SALIDA
U5 MEMORIA RAM MCM6264	$\overline{CS1}$ $\overline{CS2}$ \overline{OE} \overline{W} L H L H		SALIDA DE DATOS
U9 COMPARADOR 74LS688	G L	ENTRADA A = B	L
BUS DE TRANSFERENCIA 74LS245 U1 Y U2 U3 Y U4 U6 Y U7	G DIR L L L H H X	TRANSMISION DE DATOS DEL BUS B AL A. DEL BUS A AL B AISLADO.	

Tabla 8 Condiciones de operación en los controles de cada elemento para leer en la memoria (simular la ROM).

NOTA:

$\overline{CS1}$, $\overline{CS2}$, \overline{OE} , \overline{W} , G, y DIR son terminales de control y habilitan a la memoria RAM y al bus de transferencia, observar las terminales de control de la tabla 7 y 8.

A, B, F, y U son terminales de entrada. además las siguientes letras representan niveles de voltaje:

L = Nivel activo bajo (cero lógico)

H = Nivel activo alto (uno lógico)

X = Condición de No importa.

2.2.2 DISEÑO DEL SOFTWARE

El diseño del "software" consiste en el desarrollo de la programación para que el circuito simulador de ROM funcione adecuadamente. La programación se realiza en lenguaje "C" y toma información de un archivo tipo ASCII, la interoreta y la deposita en una localidad de memoria específica del microprocesador de la PC.

"C" es un lenguaje de programación de empleo general, caracterizado por poseer un moderno control de flujo y estructuras de datos, así como un rico conjunto de operadores. No está especializado para una área de aplicación en particular. Su carencia de restricciones y su generalidad lo hace más eficaz y conveniente para muchas tareas principalmente para la programación de microprocesadores e interfaces con la PC.

La información de los archivos consiste en instrucciones y direcciones que se deben depositar directamente en la ROM de un sistema bajo prueba, así esta información se simula en la tarjeta y se verifica su correcto funcionamiento o se corrige si tiene algún error. Si la información de estas instrucciones son correctas se grabarán en forma definitiva en la ROM del sistema bajo prueba.

2.2.2.1 DIAGRAMA DE FLUJO DEL PROGRAMA

El diagrama de flujo del programa para el manejo del circuito simulador de ROM y las funciones que realiza se muestra en la fig. 12.

A continuación se escribe en una forma general cada bloque:

DIAGRAMA DE FLUJO DEL PROGRAMA SIMULADOR DE ROM EN LENGUAJE " C "

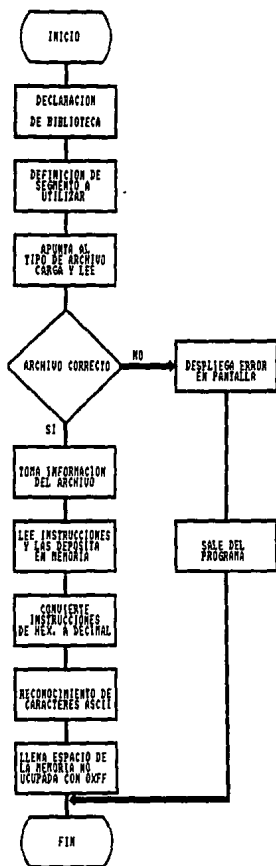


Fig. 12 Diagrama de flujo del programa simulador de ROM.

Iniciamos entrando al lenguaje "C" desde el directorio principal teciendo D:\CD PC.

Desde dentro del lenguaje "C" iniciamos el programa con encabezados que define la declaración de los archivos de biblioteca de "C" que ocupamos.

El tercer bloque define el segmento de inicio de la memoria del microprocesador de la PC que ocuparemos, declarando un nombre simbólico que se ocupará en el archivo fuente (programa principal del simulador de ROM).

Una vez realizado lo anterior como primera función del programa carga y lee el nombre del archivo a simular de la entrada estándar.

Ya cargado el programa revisa si el archivo es correcto, si no lo es despliega en pantalla un mensaje de error, sale del programa y finaliza. Si es correcto el nombre del archivo pasa a la siguiente función en la cual se encarga de tomar la información del archivo a simular.

Posteriormente lee las instrucciones, las recibe y convierte las instrucciones de hexadecimal a decimal.

La siguiente función reconoce los caracteres ASCII del programa a simular y por último el programa realiza la función de llenar con "ff" el espacio de memoria no ocupada.

En el desarrollo del programa se da mayor información sobre el programa que hace funcionar al circuito simulador de ROM especificando los comandos, las instrucciones y variables que se ocupan.

2.2.2.2 DESARROLLO DE LA PROGRAMACION DEL CIRCUITO SIMULADOR DE ROM.

La programación del circuito simulador de ROM realiza las funciones de tomar la información de un archivo tipo ASCII, interpretarla y depositarla en una localidad de memoria específica del microprocesador de la PC.

Para convertir el programa del archivo tipo ASCII (hexadecimal) a binario, (porque la información binaria (unos y ceros) la pueden reconocer los circuitos integrados), lo interprete, lo deposite en una localidad de memoria del microprocesador de la PC y simule la memoria ROM en la tarjeta, se desarrolló el programa del simulador de ROM, llamado SIMROM.EXE utilizando el compilador TURBO C versión 2.0 de Borland.

El programa SIMROM.EXE significa el programa fuente o principal, ejecutable en la PC, es la parte inteligente de la tarjeta Simulador de ROM, el cual la hace funcionar adecuadamente gancole una serie de instrucciones a la PC y al circuito simulador.

La programación esta constituida por comandos, encabezados, operadores, variables y funciones, los cuales se explican a continuación:

El programa simulador de ROM inicia con 5 encabezados "#include", los cuales son archivos que contienen las definiciones de las funciones que se utilizan en un módulo o serie de módulos para la creación de un programa, estas son funciones de biblioteca ó librería del subdirectorio de TURBO "C" y su formato es el siguiente: #include <nombre_de_archivo>

Existen 3 encabezados "#define", que son extensiones que nos permiten dar un nombre simbólico a una constante que se va a utilizar varias veces en un programa, permitiendo una ejecución más rápida de esta declaración.

```
void limpia_mem(void);  
void copia_archivo(FILE *ap);  
void main (void);  
unsigned int atohex(char num[]);  
int valor (char num);
```

Las declaraciones anteriores son funciones que posteriormente se llaman en el programa para realizar el bloque de esa función.

void es una declaración que indica que no regresa ningún valor al ejecutarse la función.

unsigned int atohex(char num[]); Es una función de tipo entero que acepta valores positivos y tiene un argumento de arreglo tipo carácter.

El programa de la tarjeta simulador de ROM se muestra en la siguiente hoja y después se explica las variables y funciones que se utilizan.

En lenguaje "C" todas las variables y funciones deben ser declaradas antes de usarse, para poder ser reconocidos por el programa.

En lenguaje "C" todo programa tiene como primera función a void main(), después aparece una llave que se abre "{", ésta le indica al compilador que en ese punto empieza un bloque de archivos y termina donde se encuentra una llave que cierra "}".

EL programa esta dividido en bloques cada uno de los cuales realiza las siguientes funciones especiales:

a) En el primer bloque el programa SIMROM.EXE realiza la función de leer el nombre de los archivos desde el teclado y llama la función "copia archivo"; además verifica e informa si existen errores al abrir o cerrar el archivo.

En este bloque está la función `void main(void)`, la cual indica el inicio del programa. La llave "`{`" indica el inicio del bloque de esa función.

El comando `FILE *ap` es un apuntador para el tipo de archivo que sirve como enlace entre el programa y un archivo.

El comando `char nombre [64]` es un arreglo que permite al compilador calcular el espacio que debe reservar en memoria para guardar todos los elementos que forman el arreglo.

`Printf()` es una función de biblioteca que manda el contenido que hay en ella a la pantalla de la PC.

La función `gets(nombre)` permite leer lo que se tiene en la función `printf()`.

La función `ap = fopen()` abre y lee un archivo, además realiza el enlace del programa con el archivo.

El comando `if(condición)` es una proposición condicional que se ejecuta si ésta es verdadera, si no lo es pasa a la siguiente proposición.

`Exit()` es una función de biblioteca que origina la terminación de una iteración, bloque y del programa.

`copia_archivo(ap)` es la función encargada de copiar el archivo del programa a simular.

El comando `return` es la función de retorno.

b) En el segundo bloque el programa realiza la Función de tomar la información del archivo, decodificar y depositar en memoria la información de una línea.

Este bloque se inicia con una declaración de las variables que se van a utilizar, con la especificación de los arreglos para reservar el espacio en memoria, los cuales son los siguientes:

```
char cont[2];
unsigned int con=0;
char direc[4];
unsigned int dire=0;
char codig[2];
unsigned int codi=0;
char far#memoria;
```

char significa variable tipo caracter y unsigned int con, dire, y codi, significa asignación de enteros positivos con inicialización en cero.

void copia_archivo(FILE #ap) es la función de copiar un archivo.

"(" y ")" indica el inicio y fin del bloque.

char c y unsigned int i son declaraciones de variables tipo caracter y entero positivo utilizada para la función de ese bloque.

La asignación memoria =MK_FP(MEM_SEG, MEM_OFF) define la localidad de memoria en la que será depositada la información del archivo, ésta contiene el programa a simular.

Limpia mem() es la función que limpia la memoria antes o después de utilizarla.

While (condición) proposición es una función de iteración de ciclos, en la cual se ejecuta la proposición cuando la condición es verdadera.

La función printf() imprime en la pantalla de la PC el contenido que hay en ella.

Cont[i]= (char)fgetc(ap) lee el número de instrucciones que hay en el apuntador de la línea y lo deposita en la variable cont.

El comando `atohex` convierte la información de hexadecimal a decimal.

```
for(inicialización, condición, incremento)
    {proposición}
```

es una función de iteración de ciclo en la cual se ejecuta la proposición cuando la condición es verdadera, si no lo es pasa al siguiente instrucción del programa.

c) En el tercer bloque se realiza la función que recibe un arreglo de elementos de tipo carácter (`char`), y regresa el equivalente de hexadecimal a decimal.

`Unsigned int atohex(char num[])` es la función que recibe un arreglo de elementos tipo carácter, regresando su equivalente de hexadecimal a decimal.

`unsigned int val=0` e `int i=0` son declaraciones y asignación de variables enteras utilizadas para este bloque de la función.

Se utiliza la iteración `for` (inicialización, condición, incremento), (proposición), se ejecuta la proposición cuando la condición es verdadera. si no lo es pasa a la siguiente instrucción.

`Exit()` es una función de biblioteca que origina la terminación de una iteración, un bloque o un programa.

`return val` es el valor de retorno y la función regresa a su llamada.

d) En el cuarto bloque el programa realiza la función que se encarga de reconocer un carácter ASCII es decir acepta el valor en hexadecimal y convierte su equivalente decimal para caracteres en hexadecimal.

En este bloque se utiliza la función if - else que es una iteración del ciclo y da la instrucción del control de flujo, es decir que indica el camino que seguirá el programa y su forma es :

```
if (condición)
    proposición;
else if(condición)
    proposición;
    ...
else
    proposición;
```

Si la condición del primer if es cierta se ejecuta su respectiva proposición, si no, pasa a la siguiente if y se evalúa su condición. Si es verdadera se ejecuta si no lo es pasa a la siguiente, así hasta evaluar la última condición y ejecutar la última proposición.

e) La última función que realiza el programa es la función encargada de llenar con ff el espacio de memoria desde la localidad A000:E000 hasta la localidad A000:FFFF. En este bloque se utiliza la función limpia_mem() y la función de iteración for(inicialización, condición) {proposición}.

El programa para el circuito simulador, SIMROM.EXE debe ser cargado en la computadora donde se va a instalar la tarjeta Simulador de ROM.

En seguida se presenta el programa SIMROM.EXE que hace funcionar a la tarjeta simulador de ROM.

/* LISTADO DEL PROGRAMA */

```

/*****
 *
 *      UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
 *
 *      ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON
 *
 *
 *
 *      PROGRAMA DEL CIRCUITO SIMULADOR DE ROM " SIMROM.C "
 *
 *****/

```

```

#include <stdio.h>           /*Librerias*/
#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#include <math.h>

```

```

#define MEM_SEG      0xA000 /*Segmento de inicio de la memoria*/
#define MEM_OFF      0xE000 /*Segmento del "offset"*/
#define TAMANO       0x2000 /*Nombre simbólico que tendrá la cte.*/
                          /*que será sustituido por el archivo */
                          /*fuente*/

```

```

void limpia_mem(void);      /*Declaración de funciones y variables*/
void copia_archivo(FILE *ap);
void main(void);

```

```

unsigned int atoi(char num[]);
int valor(char num);

```

```

/*****
 *
 *      FUNCION QUE LEE EL NOMBRE DEL ARCHIVO DESDE EL TECLADO Y LLAMA LA
 *      FUNCION COPIA ARCHIVO, ADEMÁS VERIFICA E INFORMA SI EXISTE ERROR
 *      AL ABRIR O CERRAR EL ARCHIVO
 *
 *****/

```

```

void main(void)             /*Inicio del programa*/
{
    FILE *ap;               /*Apuntador al tipo de archivo*/
    char nombre[64];
    printf("\n Nombre del archivo: ");
    gets(nombre);           /*Lectura del nombre de archivo*/

    ap = fopen(nombre, "r"); /*Abre archivo para lectura*/
    if (ap == NULL)
    { printf("\nError en el nombre del archivo"); /*Corta y sale del programa si hay*/
      exit (1);             /*error de inicializacion del apuntador*/
    }

    copia_archivo(ap);      /*Copia archivo*/
    if (fclose(ap))
        printf("\nerror al cerrar archivo");
    return;
}

```

```

/*****
 * ESTOS ARREGLOS SE DECLARAN FUERA PORQUE SE VAN A USAR EN VARIAS FUNCIONES
 *
 *****/

char cont[2];                               /*Declaración de variables e */
unsigned int con=0;                          /*inicialización*/
char direc[4];
unsigned int dire=0;
char codig[2];
unsigned int codi=0;
char far*memoria;                          /*Declaración de apuntador lejano*/
/* a memoria */

/*****
 *
 * FUNCION ENCARGADA DE TOMAR LA INFORMACION DEL ARCHIVO
 * PARA DECODIFICARLA Y DEPOSITARLA EN MEMORIA
 *****/

void copia_archivo(FILE *ap)                /*Apunta al archivo*/
{
char c;                                     /*Declaración de variables*/
unsigned int i;
memoria=MK_FP(MEM_SEG, MEM_OFF);          /*Definición de la localidad de */
/*memoria en que será depositada */
/*la información*/
limpia_mem();                             /*Llamado a la función que limpia*/
/*la memoria antes de usarla*/

while((c=(char) fgetc(ap)) != EOF)
{
if(c!=':')                               /*Marca el inicio de una línea*/
{
printf("el primer caracter es diferente de :");
return;
}

cont[1]=(char) fgetc(ap);                /*Lee el número de instrucciones que*/
cont[0]=(char) fgetc(ap);                /*hay en el apuntador de la línea*/
/*y lo deposita en la variable cont.*/
cont[2]='\0';                             /*Marca el final de caracteres de la*/
/*cadena*/
printf("\n%s\n", cont);                  /*Imprime arreglo alfanumérico */
con=atohex(cont);                        /*Convierte de hexadecimal a decimal*/
printf("\n%u\n", con);                    /*Imprime entero decimal*/

direc[3]=(char) fgetc(ap);                /*Lee los digitos que dan la dirección*/
direc[2]=(char) fgetc(ap);                /*y lo deposita en la variable direc.*/
direc[1]=(char) fgetc(ap);
direc[0]=(char) fgetc(ap);
direc[4]='\0';                             /*Marca el final de la cadena de */
/*caracteres*/
dire=atohex(direc);                       /*Convierte de hex. a decimal */
printf("se tienen %u instrucciones para la dirección %u\n", con, dire);

c=(char) fgetc(ap);                       /*Lee apuntador*/
c=(char) fgetc(ap);
if(c=='!')                                /*Verifica código de Fin de archivo*/
{
printf("fin de archivo");
return;
}
}

```



```

/*****
 *
 *      DEPOSITANDO EN MEMORIA LA INFORMACION DE UNA LINEA
 *
 *****/

printf("\nel corrimiento es de %u\n",dire);

for(i=0;i<con;          /*Deposita en memoria [con] datos*/
{
    codig[1]=(char)fgetc(ap); /*Lee el apuntador y lo deposita en */
    codig[0]=(char)fgetc(ap); /*codig*/
    codig[2]='\0';          /*Marca el final de la cadena */

    codi=atohex(codig);    /*Convierte de hex. a decimal*/
    *(memoria+dire+i)=codi; /*Apunta a memoria - direcci3n + 1 */
    i++;                  /*en la variable codi. incrementando*/
}
c=(char)fgetc(ap);      /*Caracter de verificaci3n de suma*/
c=(char)fgetc(ap);     /*verificaci3n total */
c=(char)fgetc(ap);     /*caracter de fin de l3nea */

}

printf("salio por un lado desconocido");
return;

}

/*****
 *
 * FUNCION QUE RECIBE UN ARREGLO DE ELEMENTOS TIPO char Y REGRESA
 * SU EQUIVALENTE DE HEXADECIMAL A DECIMAL.
 *
 *****/

unsigned int atohex(char num[])
{
    unsigned int val=0;
    int i=0;
    for(i=0;num[i]!='\0';)
    {
        if(valor(num[i])==-1)
        {
            printf("\n CHARACTER ERROR %c\n",num[i]);
            exit(1);
        }
        val=val+(valor(num[i])*pow(16, i));
        i++;
    }

    return val;
}

```

```

/*****
 *
 *   FUNCION QUE RECONGCE UN CARACTER ASCII Y REGRESA SU EQUIVALENTE
 *   DECIMAL PARA CARACTERES EN HEXADECIMAL
 *
 *****/

```

```

int valor(char num)
{

```

```

    if(num=='0')
        return (0);
    if(num=='1')
        return (1);
    if(num=='2')
        return (2);
    if(num=='3')
        return (3);
    if(num=='4')
        return (4);
    if(num=='5')
        return (5);
    if(num=='6')
        return (6);
    if(num=='7')
        return (7);
    if(num=='8')
        return (8);
    if(num=='9')
        return (9);
    if(num=='A')
        return (10);
    if(num=='B')
        return (11);
    if(num=='C')
        return (12);
    if(num=='D')
        return (13);
    if(num=='E')
        return (14);
    if(num=='F')
        return (15);
    else
        return (-1);
}

```

```

/*****
 *
 *   ESTA FUNCION SE ENCARGA DE LLENAR CON 0xFF EL ESPACIO DE MEMORIA
 *   DESDE LA LOCALIDAD A000:E000 HASTA LA LOCALIDAD A000:FFFF
 *
 *****/

```

```

void limpia_mem(void)
{

```

```

    int i=0;
    for(i=0; i<=TAMAND; )
    { *(memoria+i)=0xFF;
      i++;
    }
    return;
}

```

```

/*FIN DEL PROGRAMA*/

```

C A P I T U L O 3

CONSTRUCCION DEL CIRCUITO SIMULADOR

Después del análisis y alternativas para desarrollar el circuito simulador se diseñó el diagrama esquemático del circuito con el auxilio del paquete ORCAD SDT. se va a construir un prototipo y a realizar las pruebas en la computadora junto con la programación respectiva. Al verificar su funcionalidad en la tarjeta experimental, se llevará a cabo el diseño y construcción de la tarjeta en circuito impreso.

3.1 DISEÑO DEL CIRCUITO IMPRESO

Para realizar el diseño del circuito impreso, la distribución de los componentes y las rutas de intercomunicación con los dispositivos que integran la tarjeta se va a utilizar el paquete ORCAD PCB.

Otro criterio para el diseño es buscar la distancia más corta de interconexión de todas las partes que integran la tarjeta, considerando el uso de una tarjeta impresa en las dos caras, así como la máxima proximidad entre las señales de entrada y salida de cada componente que la integra.

Las líneas de tierra y de alimentación de + 5 V se desarrollan usando el ancho mayor de ruta posible ya que deben soportar el paso de la corriente.

Los diagramas de la tarjeta impresa se muestran en la fig. 13 y 14. En ella se pueden observar las líneas de interconexión de todos los elementos que integran al circuito simulador de ROM, la fig. 13 es la tarjeta del lado donde van los componentes por lo que las líneas no se unen a las terminales de cada elemento, además se observa en ella que las líneas más anchas son líneas de alimentación (GND y el voltaje de + 5V). La fig. 14 es el diagrama donde van las pistas de interconexión que unen los circuitos del simulador, en esta cara es el lado de soldadura, ya que las líneas de interconexión se unen en las terminales de cada elemento de la tarjeta y por donde se soldan los componentes, también se observa en esta cara de la tarjeta que las líneas de alimentación (GND y el voltaje de + 5 V) son de un ancho mayor que las demás.

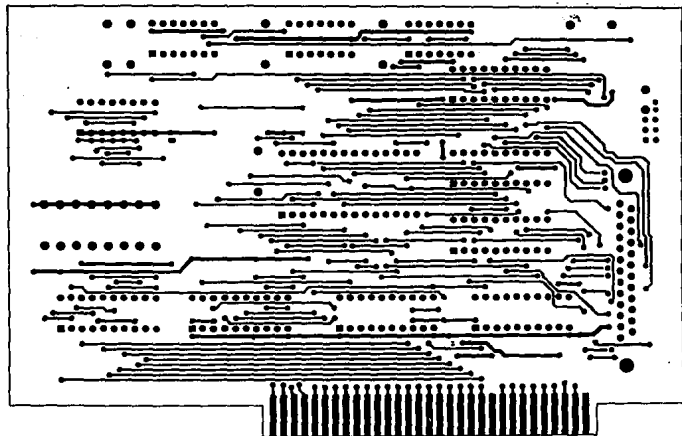


Fig. 13 Circuito impreso de la tarjeta simulador de ROM lado de componentes

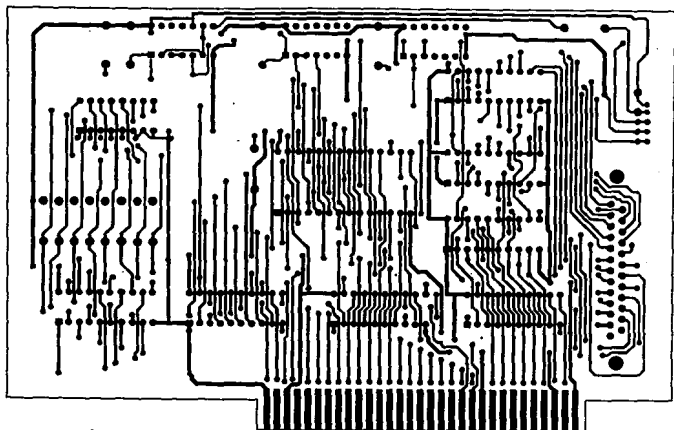


Fig. 14 Circuito impreso de la tarjeta simulador de ROM lado de soldadura

3.2 CONSTRUCCION DEL CIRCUITO IMPRESO

El trazo fotográfico de las pistas ó venas para la tarjeta se realizó utilizando una máquina Repromester modelo 310 marca Agfa, cuyo uso principal es obtener los negativos del circuito impreso para la construcción de la tarjeta. A partir de los dibujos originales, los cuales se elaboraron en el laboratorio de electrónica del Instituto Nacional de Investigaciones Nucleares.

Una vez obtenido el negativo de la película del circuito impreso, esta se adhiere a la tarjeta fibra de vidrio con dos caras de cobre, se utiliza el líquido revelador foto-resist para sensibilizar e imprimir las pistas ó venas del circuito, se enjuaga con agua limpia y posteriormente se utiliza el líquido tricotileno para quitar las impurezas, quedando sólo el diagrama del circuito impreso, se enjuaga nuevamente con ácido cloruro férrico, se estaña, se limpia nuevamente y por último se perfora los orificios donde se van a soldar los componentes de la tarjeta.

Una vez realizado el circuito impreso se llevó a cabo el ensamble de los componentes observando en él, la mejor disposición de los componentes, cuidando la facilidad de montaje y su estética, llegando a la disposición de componentes que se muestra en la fig. 15.

La especificación de los componentes corresponden a los indicados en esta figura con los valores que están en la lista de partes de la Table 11 del apéndice A.

Todos los circuitos están montados sobre portaintegrados para facilitar la prueba y mantenimiento de la tarjeta.

La tarjeta que se adiciona a la terminal de expansión de la PC está dentro de las dimensiones estándar ISA (Industrial Standard Architecture) para tarjetas adicionales a la PC (que es 33.02 cm de largo por 1.2 de ancho y 12.70 cm de altura).

Las dimensiones de la tarjeta simulador de ROM es de 16.4 cm de largo por 10.6 cm de alto y 1.2 cm de ancho. (ver fig. 16).

La tabla 9 muestra las dimensiones máximas para tarjetas admitidas en el conector de extensión de la PC XT y AT, estandarizada por IBM, el cual se compara con las dimensiones de la tarjeta simulador de ROM.

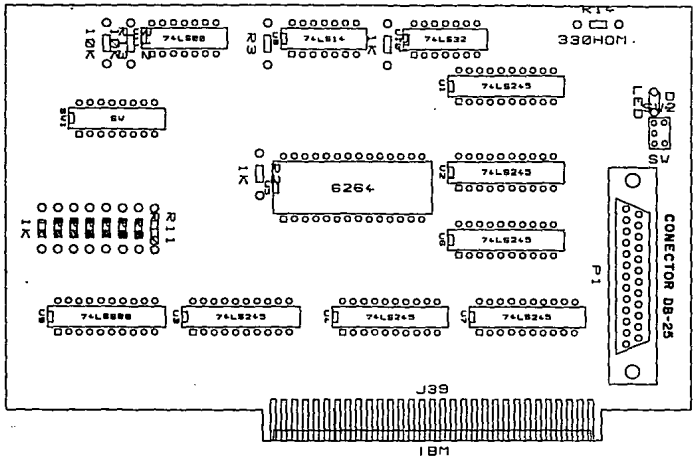


Fig. 15 Distribución de componentes de la tarjeta simulador de ROM

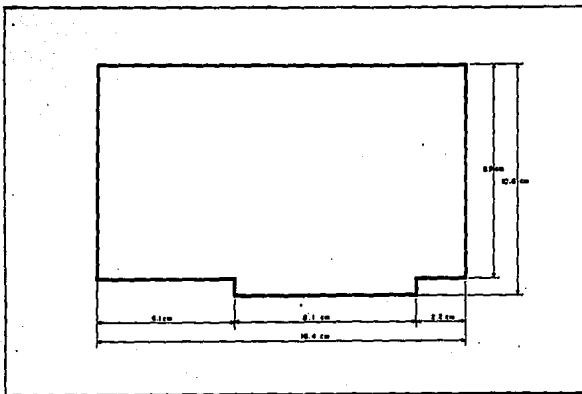


Fig. 16 Dimensiones de la tarjeta

MAXIMAS DIMENSIONES PERMITIDAS PARA LA TARJETA DE EXPANSION DE LA PC				
	PC XT	PC AT	TARJETA SIMULADOR DE ROM	UNIDAD
ALTO	10.66	12.19	10.6	cm
LARGO	33.52	33.52	16.4	cm
ANCHO	1.27	1.27	1.1	cm

Tabla 9 Comparación de las dimensiones de la tarjeta simulador de ROM con la estandarizada por ISA.

La caja de aluminio de forma rectangular tiene las siguientes dimensiones: 5 cm de ancho, 10 cm de largo por 4.5 cm de altura. Esta caja nos sirve para contener al interruptor y al LED (diodo emisor de luz). Las señales del LED y del interruptor originalmente se encuentran en la tarjeta simulador de ROM, pero para mayor comodidad, visualización y control son trasladados a la caja de aluminio. Por la caja de aluminio pasa el cable de interfaz que une la tarjeta simulador de ROM con el sistema minipc bajo prueba como se observa en la fig. 17.

3.3 INTERRUPTOR Y CABLE PLANO

El interruptor que controla la dirección de la transferencia de la información del programa (de la PC al simulador o del simulador al circuito bajo prueba) se encuentra en una caja rectangular de aluminio, también se encuentra un "LED" que indica la dirección de transferencia y además el cable plano que hace la transferencia de información de la tarjeta simulador de ROM al circuito bajo prueba el cual pasa por medio de la caja de aluminio donde se encuentran todos estos elementos como se observa en la fig. 17.

El interruptor direcciona la transferencia de información de la tarjeta al sistema bajo prueba; en la caja de aluminio se encuentra un "LED" indicador que enciende al cargar el programa a simular en la tarjeta simulador de ROM y se apaga cuando se transmite esta información al sistema bajo prueba.

La salida para la transferencia de información del circuito simulador de ROM se realiza a través de un conector estándar tipo DB-25, y la señal del LED indicador así como la señal del interruptor son transferidos de la tarjeta simulador de ROM a la caja de aluminio para mayor comodidad, utilizando para esto un conector EIS (sistema económico de interconexión). Estas salidas parten de la superior derecha de la tarjeta. El cable de interfaz que une a la tarjeta simulador de ROM con el sistema bajo prueba, así como la computadora y la caja de aluminio se muestra en el fig. 17.

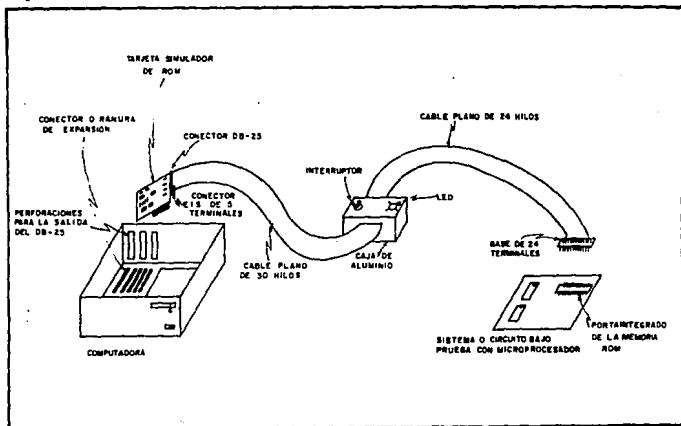


Fig. 17 Conexión de la tarjeta con la computadora, la caja de aluminio y el circuito bajo prueba.

El Corector EIS se encuentra en la tarjeta simulador de ROM y se utiliza para transferir las señales del LED y del interruptor a la caja de aluminio. En la posición "1" del interruptor la transferencia de información es de la PC a la tarjeta simulador de ROM y el LED indicador esta encendido, en la posición "2" la transferencia de información es de la tarjeta al sistema bajo prueba y el LED indicador esta apagado. En la fig. 18 se muestra la configuración de las señales del LED indicador y del interruptor, el cual se encuentran en la construcción interna de la caja de aluminio.

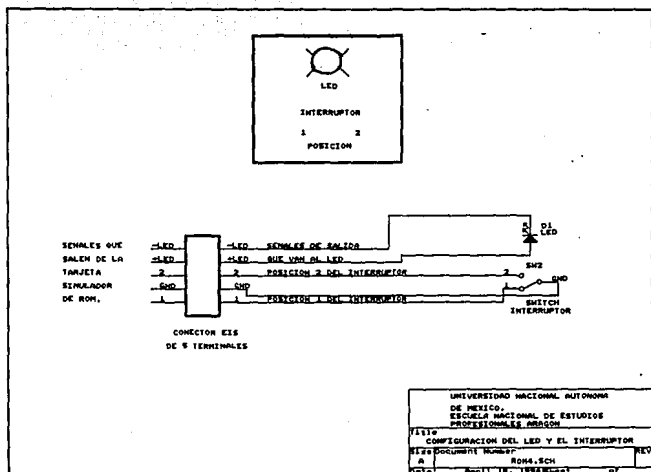


Fig. 18 Configuración del LED y del Interruptor

3.4 CONECTOR DB-25 PARA LA TARJETA SIMULADOR DE ROM

Para la conexión entre la tarjeta simulador de ROM y el circuito bajo prueba, se utiliza un conector DB-25 de 25 terminales o entradas. Aquellos que tienen terminales son conocidos como conectores machos, los que contienen las entradas son conocidos como conectores hembras. Cada terminal o entrada tiene un número de identificación, el cual está generalmente impresa en el cuerpo del conector.

La configuración de las terminales del conector DB-25 y las terminales de la base de 24 terminales se muestra en la fig. 19.

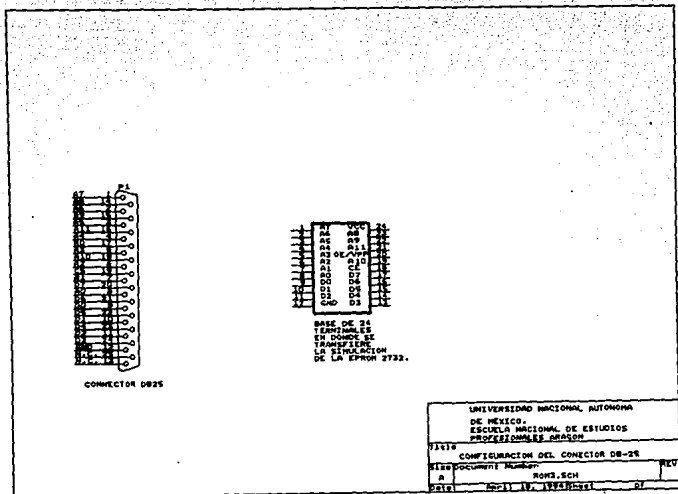


Fig. 19 Configuración del conector DB-25 y la base de 24 terminales para el simulador de ROM.

Las señales del A0 al A11 son líneas de dirección, las señales D0 al D7 son líneas que constituyen al "bus" de datos, RD es la señal que habilita para la lectura, CS es la señal del Chip Selección, GND es la señal de alimentación (Tierra) y N.C. indica no conducción.

Todas las señales anteriores son las que nos permiten llevar a cabo la comunicación entre la tarjeta Simulador de ROM y el sistema bajo prueba.

El conector DB-25 junto con el cable plano transfieren la información que contiene la tarjeta simulador de ROM a una base de 24 terminales tipo coid donde contendrá la ROM del circuito bajo prueba y del cual se ha simulado su programación para un sistema con microprocesadores.

En la tabla 10 se encuentra la especificación de las señales del conector D-25 y de la base de 24 terminales los cuales transfieren la información entre la tarjeta simulador de ROM y el sistema mínimo bajo prueba.

CONECTOR D-25 (TARJETA)	BASE DE 24 TERMINALES
1 A7	1 A7
2 A6	2 A6
3 A5	3 A5
4 A4	4 A4
5 A3	5 A3
6 A2	6 A2
7 A1	7 A1
8 A0	8 A0
9 D0	9 D0
10 D1	10 D1
11 D2	11 D2
12 GND	12 GND
13 N.C.	13 D3
14 A8	14 D4
15 A9	15 D5
16 A11	16 D6
17 A0	17 D7
18 A10	18 CE
19 CS	19 A10
20 D7	20 DE/VFP
21 D6	21 A11
22 D5	22 A9
23 D4	23 A8
24 D3	24 VCC
25 N.C.	

Tabla 10 Señales del conector D-25 y la base de 24 terminales

CAPITULO 4

MONTAJE Y PRUEBAS

En este capítulo se describe el procedimiento para el montaje y pruebas del circuito simulador de RQM.

Para realizar las pruebas se debe contar con una computadora PC XT o AT, verificando por secciones la operación adecuada del circuito y realizando las pruebas de carácter general que determinan las especificaciones de operación.

4.1 MONTAJE

Para realizar el montaje de las componentes en la tarjeta, en la computadora personal y en la caja de aluminio se realizan los siguientes pasos :

a) Se verifica que las líneas del circuito impreso estén libres de grasa o suciedad, así como asegurarse que no haya uniones entre líneas contiguas o rupturas de rutas y que todos los nodos con componentes estén perforados.

b) Se soldan todos los portaintegrados para los circuitos integrados, verificando la continuidad entre la pista y la terminal de la base, así como las terminales de cada elemento. Estos portaintegrados ("bases") nos permiten cambiar fácilmente al CI en caso de que se quemé o no sirva el integrado, haciéndolo más versátil para cambiar los elementos de la tarjeta.

- c) Se realiza una verificación exhaustiva de la conexión de las líneas del diseño del circuito impreso, especialmente donde se localizan los puntos de unión entre las dos caras de la tarjeta, así como la búsqueda de posibles corto circuitos entre las líneas de tierra y + 5 V.
- d) Se montan y se soldan todos los componentes que integran a la tarjeta simulador de ROM, así como el LED (diodo emisor de luz) y del interruptor que se encuentran en la caja de aluminio, verificando la posición correcta de cada elemento.
- e) Se conecta el cable plano de 30 hilos a los conectores DB-25 y al conector EIS de 5 terminales verificando que las señales tengan las posiciones correctas.

4.2 PRUEBAS

Las pruebas que se realizan en el circuito y en la caja de aluminio consiste en instalar la tarjeta simulador de ROM en la PC y verificar su adecuado funcionamiento, para esto se puede utilizar una computadora XT ó AT con las siguientes especificaciones:

- 1).- La computadora personal puede ser XT ó AT compatible con IBM con memoria RAM libre mayor a 8 Kbytes y con un manejador de disco flexible de 3 1/2 ó 5 1/4 .
- 2).- Se requiere que la PC tenga un conector de expansión de 8 bits libre para instalar la tarjeta simulador de ROM.

Las pruebas realizadas fueron las siguientes:

- a) Primeramente se tomó en consideración los pasos de montaje del circuito dadas anteriormente y el manual de operación del apéndice C.

b) Una vez instalado los componentes se le aplica un voltaje de + 5 V para medir la corriente de consumo de la tarjeta la cual no debe ser mayor a 400 mA. Esta prueba permite verificar que la tarjeta no tenga un corto circuito, ya que si no se hace esta primera prueba y se instala directamente al conector de expansión de la PC puede causar daños irreversibles a la computadora.

c) Posteriormente se instala la tarjeta en la PC, en caso de haber un conflicto al encender la PC, retirar la tarjeta y realizar los primeros 3 pasos de la sección de montaje para verificar que no haya uniones entre líneas contiguas o corto circuito en la tarjeta.

d) Se verifica que no haya calentamiento en ninguno de los componentes.

e) Se instalan los paquetes de programación a utilizar en el disco duro de la PC como se indica en el instructivo de operación del apéndice C.

f) Una vez cargado en el disco duro de la PC el programa ejecutable SIMROM .EXE (Programa que contiene las instrucciones para que el simulador funcione adecuadamente), se corre el programa. Este punto es el más importante del simulador, si no hay error significa que los circuitos asociados y la programación funcionan adecuadamente. Esto indica la adecuada operación de la simulación del circuito y la perfecta comunicación con la PC.

Para realizar las pruebas en la tarjeta simulador de ROM se debe considerar las instrucciones del manual de operación, el cual se indica en el apéndice C.

Las pruebas se realizan en el simulador con el fin de interpretar y depositar la información del archivo tipo ASCII a la localización de memoria del Simulador de ROM: se traslada la información al sistema bajo prueba y se verifica si realmente el simulador funciona adecuadamente. además se utilizó un programador universal tipo Multiprogramador marca Bytek 135H, el cual nos permite leer las instrucciones del simulador y verificar su correcto funcionamiento.

La secuencia en la realización de las pruebas en la tarjeta simulador de ROM fue la siguiente:

Una vez montada la tarjeta se carga en la PC el programa SIMROM.EXE. del circuito simulador de ROM, previamente el interruptor del circuito que se encuentra en la caja de aluminio debe estar en la posición 1 de "encendido" para que la información del programa se cargue en la tarjeta. En este momento aparece en la pantalla FILE NAME y se carga en la PC el programa a simular escribiendo el nombre del archivo en prueba, este archivo debe tener un nombre no mayor de ocho caracteres y con extensión HEX, en caso contrario aparecerá en la pantalla un mensaje de error como el siguiente:

Error en el nombre del archivo

Si no hay error aparece fin de archivo, esto indica que se ha cargado el programa en la tarjeta simulador de ROM, se cambia a la posición 2 del interruptor (se apaga el "LED") y se transfiere la información del programa a simular al circuito bajo prueba. Se verifica el funcionamiento del programa físicamente, es decir se prueba la programación de las memorias ROM o EPROM.

Se hicieron 4 pruebas de transferencia de programas, utilizando los siguientes:

- 1) Programa que genera un voltaje de 5 volts en las terminales de salida (21 al 28) del microprocesador 8031. Tiene la finalidad de verificar un programa que permita obtener un voltaje de 5 volts dc de amplitud en las terminales 21 al 28 del microprocesador 8031.
- 2) Programa que genera un tren de pulsos en las terminales de salida (21 al 28) del microprocesador 8031.

3) Programa que controla un sistema de detección de radiación, el cual verifica la presencia de radiación en la entrada principal del Instituto Nacional de Investigaciones Nucleares. Este programa cuenta el fondo radioactivo de radiación gamma y verifica la presencia de radiación.

4) Programa que genera un corrimiento de bits.

Todos estos programas se simularon en la tarjeta y se verificó su información con el multiprogramador universal tipo Multiprogrammer marca ByteK 135H.

Los resultados de estas pruebas se encuentran en el siguiente capítulo.

C A P I T U L O 5

RESULTADOS Y CONCLUSIONES

5.1 RESULTADOS

Los resultados de las pruebas son satisfactorios ya que la tarjeta efectivamente simula los programas para el sistema mínimo bajo prueba. Así también se verificó el contenido del programa a simular, esto gracias al multiprogramador universal marca ByteK modelo 135, el cual para los ejemplos de prueba muestran la siguiente información:

Los programas simulados se verificarón en el multiprogramador ByteK modelo 135, los programas simulados fuerón archivos de tipo ASCII los cuales contienen instrucciones para programar a una memoria ROM, EPROM y estos a su vez programan a los microprocesadores para los sistemas de aplicación específica.

Los resultados de los archivos que se simularón son de tipo "ASCII" y tienen la siguiente forma:

C:\ BASCULA.HEX

```
:03000000020100FA
:0C002300309903C29932E59909C2983265
:2001000075500075E170759920758DCC7588407587807598707580FF75A890
D2AFB48AF957
:1E0120007F1E07AFF7E930FAC2AF7920C299E7F590F5993099FDC29909B930
F280DD0D
:00000001FF
```

En este archivo está contenida la información de las direcciones y los códigos que deben ser depositados en la memoria EPROM del sistema bajo prueba.

La información está separada por líneas y en la primera existen los siguientes datos:

```
: 03 0000 00 02 01 00 FA
```

DONDE:

- [:] Marcan el inicio de una línea.
- [03] Es el número de instrucciones que tiene esa línea en hexadecimal y en decimal equivale a 3.
- [0000] Es la dirección a partir de la cual se deben depositar las instrucciones.
- [00] Es el código que identifica el tipo de línea.
- [020100] Son los tres datos que deben quedar en memoria y contienen las instrucciones para programar al microprocesador.
- [FA] Es la suma de la línea para verificación.

La última línea del programa tiene siempre la forma :

```
" : 00 0000 01 FF "
```

Cuando aparece ésta línea se da por terminado el archivo.

La información de los datos en la primera línea del ejemplo queda direccionada en la memoria de la siguiente forma:

DIRECCION	DATO
0X0000	02
0X0001	01
0X0002	00

Los resultados de las pruebas realizadas en la tarjeta simulador de ROM fueron las siguientes:

A:\ Resultados del archivo PRUEBA1.HEX

Este programa genera una amplitud de 5 volts dc. en las terminales de salida (21 al 26) del microprocesador 5031, y el resultado que muestra en el programador y en la pantalla de la computadora es la siguiente:

```
:03000000020100FA
:0E0100007590001201507590FF12015080F280
:0D0150007F0F:F7E0F1EBE00FCBF00F&22B9
:00000001FF
```

El programa en lenguaje ensamblador es la que se presenta en la siguiente hoja.

```
DEFSEG SER, START=0
SEG SER
JMP PRIN
```

```

;PROGRAMA DE FRIJERA

```

```
DEFSEG PRI, START=100H
SEG PRI
```

```
PRIN: MOV P1, #00H
      CALL RETAR
      MOV P1, #0FFH
      CALL RETAR
      JMP PRIN
```

```

;RUTINA DE RETARDO

```

```
DEFSEG RET, START=150H
SEG RET
```

```
RETAR: MOV R7, #0FH           ;MUEVE A R7, FF
RE:    DEC R7                 ;DEC R7
      MOV R6, #0FFH         ;MUEVE R6, FF
SIG:   DEC R6                 ;DEC R6
      CJNE R6, #00H, SIG     ;SI R6 NO IGUAL 00H BRINCA A SIG
      CJNE R7, #00H, RE      ;SI R7 NO IGUAL 00H BRINCA RET
      RET
      END;
```

El resultado de este programa fue satisfactorio, ya que efectivamente la tarjeta simuló la programación de las instrucciones mostradas anteriormente en microprocesador 8031. En las terminales del 21 al 28 del microprocesador 8031 se encuentra el "bus" de puertos bidireccional y se observa en ellas un voltaje de 5 volts de amplitud, para verificar esta prueba se utilizó un osciloscopio marca Tektronix 2225 y la gráfica de la señal se muestra en la fig. 20.

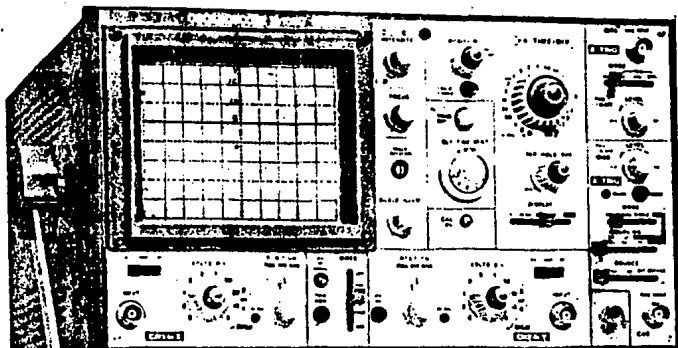


Fig.20 Gráfica de la señal mostrada en el Osciloscopio, amplitud de 5 volts para la prueba 1.

A:\ Resultados de la PRUEBA2.HEX

Programa que genera un tren de pulsos en las terminales del microprocesador 8031, estas terminales se encuentran del 21 al 28, el resultado que muestra en la pantalla de la PC y en el programador universal Bytek es la siguiente:

:020000000120DD

:1B00200074FFF590112B1460120122FF7EFF1E7DFF1DED8400FBEEB400F42261

:00000001FF

El programa en lenguaje ensamblador es la siguiente:

```
DEFSEG AAA,START=00H
SEG AAA
AJMP INICIO
DEFSEG BBB,START=20H
SEG BBB
INICIO:  MOV A,#0FFH
SIGUE:  MOV P1,A
        ACALL DELAY
        DEC A
        JZ FIN
        AJMP SIGUE

DELAY:  MOV R7,A
        MOV R6,#0FFH
TIME1:  DEC R6
        MOV R5,#0FFH
TIME2:  DEC R5
        MOV A,R5
        CJNE A,#00H,TIME2
        MOV A,R6
        CJNE A,#00H,TIME1
        RET
FIN:    END
```

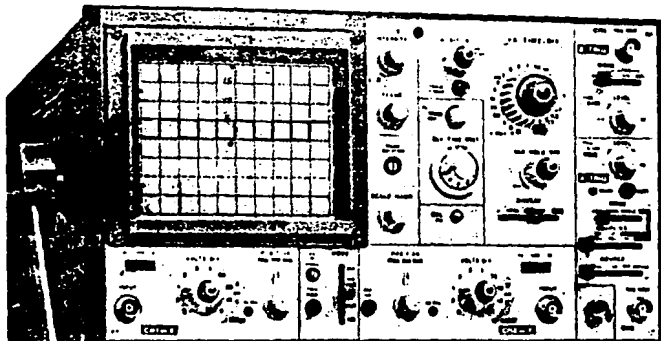



Fig.21 Gráfica de la señal mostrada en el osciloscopio
 tren de pulsos de 5 volts.

A:\ Resultados de la FRUEBAS.HEX

Programa que controla un sistema de detección de radiación,
 este programa verifica la presencia de radiación en la entrada
 principal del Instituto Nacional de Investigaciones Nucleares.

```

:03000000020200F9
:1D0200000075D00C7530007531001200DAE530B49B03020250B49D03020300B0E3EE
:200250000075F00F758800758C00758A00740075320075340075891575A888758D3C758B6
:20027000AF309239758850E53220E00280F9A88AA98C758A00758C001204367530007531E
:20029000001200DAE530B4FCF21203D97530007531001200DAE530B4FD031203D97531002
:0902B000753000753200020200F5
:200300000075880075F0007938753500758C00758A0075320075340075891575A888758D1
:200320003C758BAF74FF30920D14B400F9753100753000020200758B507400E532209205F
:2003400020E00280F609EAF709EBF7053575320020920280E6C28C28E75360075383975B
:200360003756A938E7FB09E7FC0915358938A935B9FF0280D0EBF8ECF9120436A937ECF76
:2003800009EBF709EAF70905368937AD358D00D275375674007531001200DAE530B4FCF35
:2003A000753100A937E7FC09E7FB09E7FA09153689371203D97530007531001200DAE530B
  
```

: 2003C000B4FC031203D9AD36BD0009753100753000020200B0ED020200C2AFD2B5742012A5
: 2003E000042A742012042AEC54F0C4443012042AEC540F443012042AEB54F0C444301204D3
: 200400002AEB540F443012042AEA54F0C4443012042AEA540F443012042A742012042A74C1
: 200420000A12042A00C2B5C2982200C299F599003099FCC2992200C37D107A007B007C0093
: 200440007E00C3EB33FB8E933F9EA33FAE533FBEC33FCED14700302047BFDEA12046BFAEBA6
: 1D04600012046BFREC12046BFC80D7FE240330E301FEEE243030E701FEEE220022B2
: 03001B00020040AG
: 0E004000F533C0DOE5F0B40F0774000200C025
: 200050007400758D3C758BAF20920705347414B53418C28CC28EAA8AAB8C758A00758C00AB
: 0F007000753400D28CD28E753201E533D000328B
: 2000C00000758D3C758BAF05347414B53407C28CC28E753201E533D000320075D00075890F
: 1D00E00022758DC75885075878075987075A890C2B5E531E40F030200FC80DE224F
: 1B002300309903C29932E599F530B4000675310075300075310FC2B5C2983209
: 00000001FF

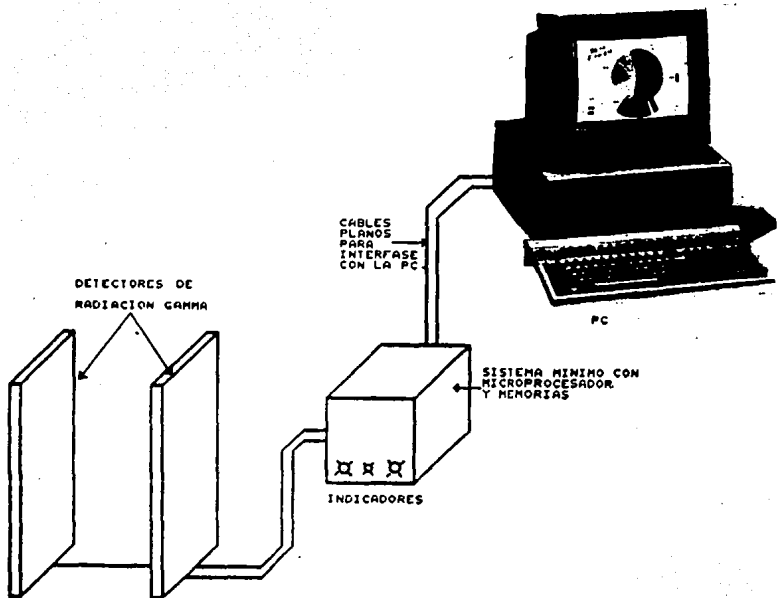


Fig. 22 Esquema de deteccion de radiación del programa anterior.

En estos archivos está contenida la información o programación el cual se simula en la tarjeta y realiza un trabajo en específico, además contiene la dirección y los códigos que deben ser depositados en la memoria EPROM del sistema bajo prueba, por lo que quedó comprobado el funcionamiento del simulador de ROM; ya que los resultados fueron satisfactorios, comprobándose que la información del programa a simular era la misma tanto del programa visualizado en la PC, como las instrucciones del programador universal. Además se observó su correcto funcionamiento físicamente a través de los sistemas bajo prueba como lo muestran las figuras respectivas de cada programa.

5.2 ALCANCES

La tarjeta simulador de ROM alcanza su objetivo ya que efectivamente simula la programación de la memoria EPROM 2732. El proyecto de esta tarjeta fue solicitada por los usuarios del área de electrónica en el Instituto Nacional de Investigaciones Nucleares, para agilizar la programación de microprocesadores y microcontroladores en los diseños y proyectos específicos que solicitan otras áreas dentro del Instituto.

La tarjeta simulador de ROM se ha utilizado para formar parte de otros proyectos más amplios agilizando los diseños y minimizando el tiempo en el desarrollo de estos. Como se puede observar en la fig. 22 la tarjeta simulador se utilizó junto con otro proyecto (Detección de radiación gamma), el cual confirmó el funcionamiento de los dos proyectos.

Los alcances de esta tarjeta es muy amplia y variada, se puede utilizar independientemente para sistemas mínimos pequeños o en conjunto con otros sistemas más amplios en el cual utilicen memorias EPROM 2732 y microprocesadores.

Se puede utilizar en alguna etapa dentro de un sistema muy amplio como por ejemplo en algún proyecto de un sistema de control en el cual utilicen programación de memorias EPROM 2732 y microprocesadores.

El alcance de esta tarjeta es de acuerdo a la necesidad del usuario para realizar la simulación de la programación de este tipo de memoria, aplicados a microprocesadores dentro de un sistema mínimo pequeño e independiente o amplio y en conjunto con otros proyectos o sistemas.

5.3 CONCLUSIONES

El desarrollo de este trabajo ha tenido la finalidad de llevar a cabo un programa específico para el área de electrónica del Instituto Nacional de Investigaciones Nucleares (ININ), y construir una tarjeta simulador de ROM o EPROM que sirva de herramienta para la simulación de estas memorias, los cuales permiten el ahorro de tiempo y versatilidad en la programación de sistemas con microprocesadores.

Además la aportación fundamental de este trabajo contribuye a la formación de recursos humanos en esta área de la ingeniería porque brinda la experiencia e introduce al campo del trabajo profesional.

En la tarjeta se ha observado su correcto funcionamiento, por lo que se considera que es una herramienta de gran utilidad, versátil y de fácil manejo para sistemas mínimos, en los cuales se utilizan memorias ROM o EPROM y en cualquier tipo de microprocesadores tales como el 8031 o en cualquier otro tipo.

La tarjeta agiliza la programación en el diseño de sistemas con microprocesadores.

Una característica relevante del circuito es la de simular o emular la programación, ya que interpreta la programación de tipo ASCII a binario para que sea entendible para los CI.

En el simulador se realizan las pruebas pertinentes a un programa o se realizan las correcciones adecuadas para que funcione correctamente en el sistema mínimo; por tal motivo es una herramienta muy dinámica y versátil que permite el ahorro de tiempo en los diseños de proyectos con microprocesadores.

Se considera un circuito electrónico con componentes de fácil adquisición, manejo, mantenimiento, y sobre todo de bajo costo.

La parte inteligente es la programación que se realiza en lenguaje "C", la cual permite la interfase entre el circuito simulador de ROM y la PC.

Esta tarjeta se debe reproducir en serie ya que los microprocesadores se están utilizando ampliamente en la actualidad y el simulador se puede ocupar en cualquier diseño donde se necesite simular la programación de las memorias ROM o EPROM.

La tarjeta realiza las funciones de un emulador ROM con la ventaja de observar físicamente la funcionalidad del programa en un sistema mínimo ya que la información simulada es aplicada a un sistema que se visualiza físicamente, es decir que se observa el circuito con todas las funciones que realiza la programación de la memoria ROM o EPROM.

Con este diseño se tiene una tarjeta que proporciona los mismos servicios de una tarjeta comercial Simulador de ROM con capacidad de 8 kbytes, pero con un costo sumamente bajo de N\$ 200.00 en componentes electrónicos. Esta tarjeta se diseñó considerando las especificaciones que requiere un usuario cuando va a trabajar con memorias ROM o EPROM, por lo que la tarjeta simula cualquier tipo de memoria EPROM 2732 con capacidad de 8 Kbytes máximo.

Si se requiere un simulador de ROM o EPROM con mayor capacidad se deberá modificar su "Hardware" y su "Software", pero los procedimientos para llevarlo a cabo son similares a los que se realizaron con la tarjeta del presente trabajo.

Para realizar dicho trabajo se utilizó una computadora personal marca Televideo, además para comprobar la funcionalidad de la tarjeta se utilizó una PC AT 386 marca Televideo.

Se concluye que el proyecto cumplió plenamente su objetivo.

B I B L I O G R A F I A

- 1.- Mueller Scott, "UPGRADING AND REPAIRING PC'S", Que Corporation, 1988, pp.279-280.
- 2.- Tokheim Roger L., "FUNDAMENTOS DE LOS MICROPROCESADORES", Mc Graw Hill, 1992, pp.67-71.
- 3.- C. Brener Robert, "IBM PERSONAL COMPUTER", Howard W. Sams & Company, 1991, pp.20-21.
- 4.- C. Eggebrecht Lewis, "INTERFACING TO THE IBM PERSONAL COMPUTER", Howard W. Sams & Company, 1987, pp. 77.
- 5.- C. Eggebrecht Lewis, "INTERFACING TO THE IBM PERSONAL COMPUTER", Howard W. Sams & Company, 1987, pp. 52-59.
- 6.- C. Eggebrecht Lewis, "INTERFACING TO THE IBM PERSONAL COMPUTER", Howard W. Sams & Company, 1987, pp. 125-143.
- 7.- Intel the Microcomputer Company, "MICROCOMPUTER BOARD AND SYSTEMS HANDBOOK", Intel, 1990, pp.17-25.
- 8.- Mueller Scott, "UPGRADING AND REPAIRING PC'S", Que Corporation, 1988, pp.279-280.
- 9.- Motorola, "FAST AND LS TTL DATA", Motorola, 1989, pp. 5.271.
- 10.- Motorola, "FAST AND LS TTL DATA", Motorola, 1989, pp.5.203. 5.203.
- 11.- Mueller Scott, "UPGRADING AND REPAIRING PC'S", Que Corporation, 1988, pp.279-280.
- 12.- Murray, H. William; Fappas, H Chris, "80386/80286 PROGRAMACION EN LENGUAJE ENSAMBLADOR", Osborne/Mc Graw-Hill, 1990, pp.3-5.

A P E N D I C E S

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

DESIGNACION	DESCRIPCION	FABRICANTE
U1, U2, U3, U4, U6 Y U7	C. I. 74LS245 BUS DE TRANSFERENCIA	MOTOROLA
UB	C. I. 74LS14 COMPUERTA INVERSOR	MOTOROLA
U9	C. I. 74LS688 COMPARADOR	MOTOROLA
U11	C. I. 74LS00 COMPUERTA NAND	MOTOROLA
U10	C. I. 74LS32 COMPUERTA OR	MOTOROLA
U5	C. I. MCM 6264 MEMORIA RAM	MOTOROLA
SW1	DIP SWITCH-8 POSICIONES	GRAY HILL
P1	CONECTOR DB-25M	PRODEL
R1 A R14	RESISTENCIAS DE 1K OHMS , 1/2 W, 5 % TOL.	PEGASO SILITRONIC
D1	DIODO EMISOR DE LUZ (LED) ROJO	MONSANTO
U1, U2, U3, U4, U6, U7, Y U9.	BASES DE 20 TERMINALES	WE
U12	BASES DE 16 TERMINALES	WE
UB, U10, Y U11	BASES DE 14 TERMINALES	WE
IT1	INTERRUPTOR 1 POLO 2 TIROS MINIATURA	ALCOSWITCH
U5	BASE DE 24 TERMINALES	WE
-----	1 m DE CABLE PLANO DE 29 HILOS CALIBRE 28 AWG.	BELDEN
J37	TARJETA DE CIRUITO IMPRESO CON TERMINAL DE EXPANSION IEM.	---

Tabla 11 Lista de componentes del Circuito Simulador de ROM

A P E N D I C E "B"

HOJA DE DATOS DE FABRICANTES DE CI

HOJA DE DATOS DE FABRICANTES DE CI.

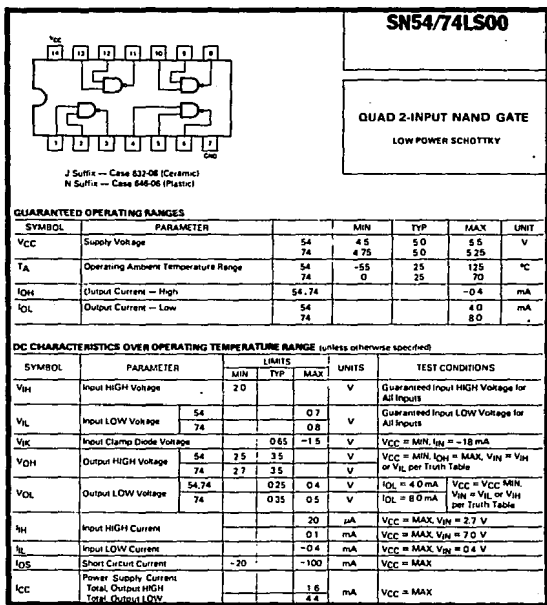


Tabla 12 Configuración de terminales y tabla de verdad del

74LS00.



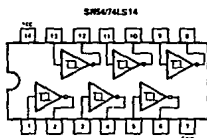
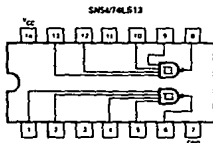
SN54/74LS13 SN54/74LS14

DESCRIPTION — The SN54LS/74LS13 and SN54LS/74LS14 contain logic gates/inverters which accept standard TTL input signals and provide standard TTL output levels. They are capable of transforming slowly changing input signals into sharply defined, jitter-free output signals. Additionally, they have greater noise margin than conventional inverters.

Each circuit contains a Schmitt trigger followed by a Darlington level shifter and a phase splitter driving a TTL totem pole output. The Schmitt trigger uses positive feedback to effectively speed-up slow input transitions, and provide different input threshold voltages for positive and negative-going transitions. This hysteresis between the positive-going and negative-going input thresholds (typically 800 mV) is determined internally by resistor ratios and is essentially insensitive to temperature and supply voltage variations.

**SCHMITT TRIGGERS
DUAL GATE/HEX INVERTER
LOW POWER SCHOTTKY**

LOGIC AND CONNECTION DIAGRAMS

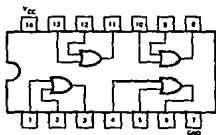


J Suffix — Case 632-06 (Ceramic)
N Suffix — Case 646-06 (Plastic)

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
VCC	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
TA	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
IOH	Output Current — High	54, 74			-0.4	mA
IOL	Output Current — Low	54			4.0	mA
		74			8.0	

Tabla 13 Configuración de terminales y tabla de verdad del 74LS14



J Suffix — Case 632-08 (Ceramic)
N Suffix — Case 646-06 (Plastic)

SN54/74LS32

QUAD 2-INPUT OR GATE

LOW POWER SCHOTTKY

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V _{CC}	Supply Voltage		54	4.5	5.0	5.5
			74	4.75	5.0	5.25
T _A	Operating Ambient Temperature Range		54	-55	25	125
				0	25	70
I _{OH}	Output Current — High		54, 74			-0.4
I _{OL}	Output Current — Low		54			4.0
			74			8.0

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74		0.8		
V _{IK}	Input Clamp Diode Voltage		-0.65	1.5	V	V _{CC} = MIN, I _{IN} = -18 mA
V _{OH}	Output HIGH Voltage	54	2.5	3.5	V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74	2.7	3.5	V	
V _{OL}	Output LOW Voltage	54, 74	0.25	0.4	V	I _{OL} = 4.0 mA, V _{CC} = V _{CC} MIN, V _{IN} = V _{IH} or V _{IL} per Truth Table
			74	0.35	0.5	
I _H	Input HIGH Current			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V
				0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V
I _L	Input LOW Current			-0.4	mA	V _{CC} = MAX, V _{IN} = 0.4 V
I _{OS}	Short Circuit Current	-20		-100	mA	V _{CC} = MAX
I _{CC}	Power Supply Current Total Output HIGH Total Output LOW			6.2	mA	V _{CC} = MAX
				9.8		

AC CHARACTERISTICS: T_A = 25°C

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
t _{PLH}	Turn On Delay, Input to Output		14	22	ns	V _{CC} = 5.0 V
t _{PHL}	Turn On Delay, Input to Output		14	22	ns	C _L = 15 pF

Tabla 14 Configuración de terminales y tabla de verdad del 74LS32



MOTOROLA

DESCRIPTION — The SN54/74LS245 is an Octal Bus Transceiver / Receiver designed for 8-bus microprocessors 2-wire data communication between data buses. Direction Input (DIR) controls transmission of Data from bus A to bus B or bus B to bus A depending upon its logic level. The Enable Input (E) can be used to disable the device.

- HYSTERESIS INPUTS TO IMPROVE NOISE IMMUNITY
- 2-ANALOG ASYNCHRONOUS DATA BUS COMMUNICATION
- INPUT DIODES LIMIT HIGH-SPEED TERMINATION EFFECTS

TRUTH TABLE

INPUTS		OUTPUT
E	DIR	
L	L	Bus B Data to Bus A
L	H	Bus A Data to Bus B
H	H	High-Z

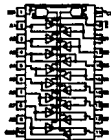
H = HIGH Voltage Level
 L = LOW Voltage Level
 Z = Impedance

SN54/74LS245

OCTAL BUS TRANSCEIVER

LOW POWER SCHOTTKY

**LEADS AND CONNECTION DIAGRAM
SEP (TOP VIEW)**



J Suffix — Case 752-01 (Cermet)
 H Suffix — Case 752-02 (Plastic)

Tabla 15 Configuración de terminales y tabla de verdad del 74LS245



**SN54/74LS682
SN54/74LS684
SN54/74LS688**

DESCRIPTION — The SN54LS/74LS682, 684, 688 are 8-bit magnitude comparators. These device types are designed to perform comparisons between two 8-bit binary or BCD words. All device types provide F=Q outputs and the LS682 and LS684 have P>Q output also.

The LS682, LS684 and LS688 are bus compatible devices. The LS682 has a 20 kΩ pullup resistor on the Q inputs for setting or output data.

**8-BIT MAGNITUDE
COMPARATORS**

LOW POWER BIPOLAR

FUNCTION TABLE

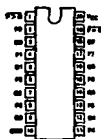
DATA P, Q	INPUTS		OUTPUTS	
	L, H*	H, L*	F=Q	P>Q
P=Q	L	L	L	H
P>Q	L	H	H	H
P<Q	H	L	H	H
X	X	X	X	X

*H=High-Level, L=Low-Level, X=Unknown

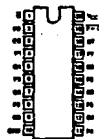
TYPE	F=Q	P>Q	INPUT SIGNALS	OUTPUT CONFIGURATION	PULLUP
LS682	yes	yes	no	open-drain	yes
LS684	yes	yes	no	open-drain	yes
LS688	yes	yes	no	open-drain	no
LS682	yes	yes	no	open-drain	no
LS684	yes	yes	yes	open-drain	no
LS688	yes	yes	yes	open-drain	no
LS682	yes	yes	yes	open-drain	no
LS684	yes	yes	yes	open-drain	no
LS688	yes	yes	yes	open-drain	no

**CONNECTION DIAGRAMS
(TOP VIEW)**

SN54LS/74LS682/684



SN54LS/74LS688



J Sells — Can 70-40 Circuit
S Sells — Can 70-40 Printer

Tabla 16 Configuración de terminales y tabla de verdad del 74LS688

**MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA**

MCM6264

**Product Preview
8K x 8 Bit Fast Static RAM**

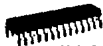
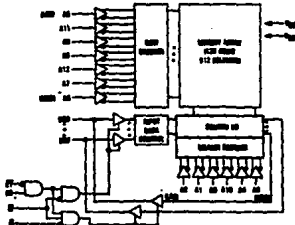
The MCM6264 is a 64Kbit static random access memory organized as 8192 words of 8 bits. Fabricated using Motorola's advanced-generation high-performance silicon-gate CMOS DPMOS cell technology. Specific design advantages are noted for external clocks or timing signals, with CMOS circuitry reducing power consumption when powerlines become inactive.

The chip enables pins \overline{ST} and \overline{CS} to not connect. Many pins, when connected, allow the part to enter a low power standby mode. The part cell consists of standard metal and bulk pins and standard bus applications. The availability of active high and active low chip enables pins provides more system design flexibility than static chip enable devices.

The MCM6264 is available in a 288 pin, 28 pin plastic dual-in-line package and a 488 pin, 28 pin plastic SOJ package. Both packages feature the JSDC standard pinout.

- Single 5 V Supply, $\geq 10\%$
- 8K x 8 Organization
- Fully Static—No Clock or Timing Sequence Necessary
- Fast Access Time—25, 45 ns (Maximum)
- Low Power Operation—110/100 nA (Maximum, Active)
- Three State Outputs
- All Inputs and Outputs are TTL Compatible
- Output Enable \overline{OE} Feature for Increased System Flexibility and to Minimize Bus Connection Problems

BLOCK DIAGRAM



**P PACKAGE
288 PIN PLASTIC**



**J PACKAGE
28 PIN PLASTIC
SOJ CASE 883**

Pin ASSIGNMENT

Pin 1	10	NC
Pin 2	17	CS
Pin 3	26	D0
Pin 4	35	D1
Pin 5	34	D2
Pin 6	23	A11
Pin 7	30	D3
Pin 8	39	D4
Pin 9	21	A14
Pin 10	38	D5
Pin 11	19	D6
Pin 12	17	D7
Pin 13	10	D8
Pin 14	18	D9

Pin Function

AD-A12	Address
CS	Chip Select
D0-D7	Data Bus
OE- \overline{OE}	Output Enable
WE- \overline{WE}	Write Enable
VCC	5 V Power Supply
VSS	Ground
NC	No Connection

Tabla 17 Configuración de terminales del MCM6264

ST	CS	D0	D7	Signal	Output Current	I/O Pin
H	H	H	H	Not Connected	—	High Z
H	L	H	H	Not Connected	—	High Z
L	H	H	H	Output Enabled	100 μ A	High Z
L	H	L	H	Not	—	Out
L	H	H	L	Write	100 μ A	Out

The driver pinout enables in product use. When output drivers are not used, the high state voltage is output buffer leakage. It is advised that output drivers be used to avoid degradation of any circuitry that the structure will be used in high-temperature areas.

Tabla 18 Tabla de verdad del MCM6264

NMC27C32 32,768-Bit (4096 x 8) UV Erasable CMOS PROM

General Description

The NMC27C32 is a high speed 3BL UV erasable and electrically reprogrammable CMOS EPROM, ideally suited for applications where fast turnaround, pattern optimization and low power consumption are important requirements.

The NMC27C32 is packaged in a 34-pin dual-in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern.

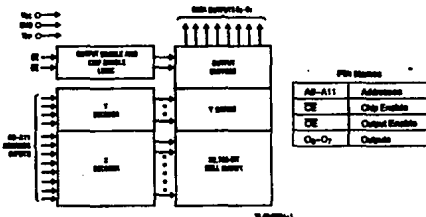
A new pattern can then be written into the device by following the programming procedure.

This EPROM is fabricated with the reliable, high volume, state proven, p⁺CMOS™ silicon gate technology.

Features

- Access time down to 300 ns
- Low CMOS power consumption
 - Active power: 25.25 mW max
 - Standby power: 0.50 mW max (80% average)
- Extended temperature range available (NMC27C32E-40 and NMC27C32E-45, -40°C to +85°C, 400 ns 1.5% power supply
- 10 ms programming available (NMC27C32E, at 60% bus voltage)
- Pin compatible to NMC2728 and higher density EPROMs
- Status-to status required
- TTL compatible inputs/ outputs
- Tri-State control
- TRI-STATE[®] output

Block Diagram



NMC27C32
Dual-In-Line Package

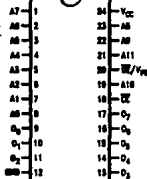


Tabla 19 Configuración de terminales del NMC27C32

A P E N D I C E "C"
M A N U A L D E O P E R A C I O N

MANUAL DE OPERACION

1. INTRODUCCION

El Circuito Simulador de ROM es una herramienta que se diseñó especialmente para simular memorias de sólo lectura ROM y EPROM, útil para sistemas mínimos que trabajan con microprocesadores.

Este manual presenta la información necesaria para el uso de la tarjeta simulador de ROM, los requerimientos para su manejo y el procedimiento a seguir para que la tarjeta funcione adecuadamente sin que se dañe.

2. EQUIPO QUE SE REQUIERE PARA MANEJAR A LA TARJETA DEL SIMULADOR DE MEMORIA ROM

El equipo requerido para que funcione la tarjeta es:

- 2.1) Una computadora Personal XT o AT compatible con IBM con 8 Kbytes o más de memoria RAM y con un manejador de disco flexible.
- 2.2) Se requiere que la PC tenga un conector de expansión de 8 bit libre para instalar la tarjeta Simulador de ROM.

2.3) Tener la tarjeta simulador de ROM con el "Dip Switch" direccionado en la dirección AB H (Hexadecimal), ver fig. 23. Estas posiciones son útiles para direccionar el intervalo de memoria del microprocesador de la FC que se ocupa precisamente para guardar la información del programa a simular y no encimar información en localidades ocupadas. Se observa que los interruptores 1, 3 y 5 deben estar en la posición ON, mientras que los restantes están en OFF.

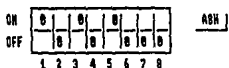


Fig. 23 Direccionamiento del "Dip Switch"

2.4) La caja de aluminio en donde encontramos al interruptor y el "LED" indicador. Además del cable plano en donde encontramos el conector D-25 y el conector EIS de 5 terminales por un extremo, y por el otro extremo del cable encontramos la base de 24 terminales, el cual une al sistema bajo prueba. Esta base de 24 terminales es por donde se transfiere la información del programa en prueba hacia el sistema mínimo. Todo el conjunto de estos elementos sirven para la transferencia de información entre la tarjeta Simulador de ROM y el circuito bajo prueba.

2.5) Disco flexible que contiene el programa SIMROM.EXE de la tarjeta simulador de ROM y el programa a simular con extensión HEX, o cargar en el disco duro de la PC el programa SIMROM y en el disco flexible el programa a simular que puede ser de baja o alta capacidad.

2.6) Se recomienda que la toma de corriente sea de 127 V AC para la PC, con una derivación hacia tierra física para protección de la máquina y del circuito simulador de ROM.

3. ESPECIFICACIONES

- 3.1) Temperatura de operación de 0 a 40 grados centígrados.
- 3.2) Suministro de energía de + 5 V dc proveniente de la fuente de la PC.
- 3.3) Corriente y potencia consumida por la tarjeta de 400 mA y de 2 W respectivamente.
- 3.4) Dimensiones de la tarjeta 16.4 cm de largo, 1.2 cm de ancho por 10.6 cm de altura para que se pueda instalar dentro de la PC.

4. PROCEDIMIENTO A SEGUIR PARA MANEJAR LA TARJETA SIMULADOR DE ROM

1.- Instalación de la tarjeta

Para instalar la tarjeta Simulador de ROM en la computadora se sigue la secuencia siguiente:

- a) Asegurarse que la computadora este apagada, que sea XT ó AT y que tenga las especificaciones dadas en el punto 3 de este manual.
- b) Quitar la cubierta de la computadora para poder tener acceso a sus terminales o conectores de expansión.
- c) Instalar la tarjeta en alguno de los conectores de expansión disponible, cuidando que la posición del conector DB-25 quede en la parte de atrás de la PC para que forme parte de los conectores externos de dicha PC.
- d) Cerrar la computadora.

2.- Conectar el cable de interfase de la tarjeta simulador de ROM a la caja de aluminio y de esta al circuito bajo prueba.

3.- Encender la computadora.

4.- El interruptor que se encuentra en la caja de aluminio debe estar en la posición 1 (encendido el "LED") para que el programa a simular se almacene primero en la tarjeta simuladora.

5.- Se carga en la PC el programa SIMROM.EXE de la siguiente forma:

a).- En el drive "a" de la PC se instala el disco que tiene este programa y se tecléa SIMROM.EXE

b).- Debe aparecer en la pantalla: FILE NAME .

c).- Escribir el nombre del archivo donde se tiene el programa para simular la memoria EPROM ("nombre con extensión HEX"), donde el nombre del archivo debe ser no mayor de 8 caracteres y la extensión debe ser HEX.

d).- Se oprime ENTER y empieza la simulación del programa ó la PC envía un mensaje de error como el siguiente: "Error en el nombre del archivo"

e).- Debe aparecer: Fin de archivo, con lo cual se indica que se ha cargado el programa en la tarjeta simulador de ROM.

6.- El programa que se carga debe ser un archivo de tipo ASCII, el cual previamente se realiza en lenguaje ensamblador utilizando un procesador de palabras. Como ejemplo se muestra el siguiente programa en ASCII y su respectivo programa en lenguaje ensamblador.

Programa en ASCII

```
:020000000120DD
```

```
:1B00200074FFF590112B1460120122FF7EFF1E7DFF1DEDB400FBEEB400F42261
```

```
:00000001FF
```


Programa prueba en ensamblador del programa anterior

	1	DEFSEG AAA,START=00H
	2	SEG AAA
0000& 01 20	3	AJMP INICIO
	4	DEFSEG BBB,START=20H
	5	SEG BBB
0000& 74 FF	6	INICIO: MOV A,#0FFH
0002& F5 90	7	SIGUE: MOV P1,A
0004& 11 2B	8	ACALL DELAY
0006& 14	9	DEC A
0007& 60 12	10	JZ FIN
0009& 01 22	11	AJMP SIGUE
	12	
000B& FF	13	DELAY: MOV R7,A
000C& 7E FF	14	MOV R6,#0FFH
000E& 1E	15	TIME1: DEC R6
000F& 7D FF	16	MOV R5,#0FFH
0011& 1D	17	TIME2: DEC R5
0012& ED	18	MOV A,R5
0013& B4 00 FB	19	CJNE A,#00H,TIME2
0016& EE	20	MOV A,R6
0017& B4 00 F4	21	CJNE A,#00H,TIME1
001A& 22	22	RET
	23	FIN: END

7.- Se cambia el interruptor a la posición 2 para transferir la información del simulador de ROM al circuito bajo prueba y se verifica si la información del programa funciona de acuerdo a lo esperado.

5. PRECAUCIONES QUE SE DEBEN DE SEGUIR

1.- Nunca instalar o retirar la tarjeta Simulador de ROM del conector de extensión de la PC cuando la computadora esta encendida porque puede causar un corto circuito ó algún daño irreversible a la computadora.

2.- Asegurarse que el conectores DB-25 y el conector EIS de 5 terminales estén bien colocados para que no tengan falsos contactos.

3.- Nunca cambiar las direcciones del "Dip Switch", ya que altera las direcciones del mapa de memoria y puede causar daños irreversibles a la PC.

4.- El interruptor debe estar en la Posición "1" ya que si no es así no hay comunicación entre la PC y la tarjeta Simulador de ROM.

6. GUIA PARA LA SOLUCION DE PROBLEMAS

A continuación se presenta una tabla con algunos problemas que se pueden presentar en la tarjeta simulador de ROM, su causa y su posible solución.

PROBLEMA	CAUSA	POSIBLE SOLUCION
No se establece la comunicación de la PC ó la transferencia de datos es errónea.	Mal direccionamiento del "Dip switch"	Corregir los interruptores del "Dip switch", es decir colocar en las posiciones correctas.
No hay transferencia de datos.	El cable de conexión esta dañado o mal configurado.	Verifique la continuidad en los conductores ó cambie el cable.
No acepta el programa	El nombre del archivo tiene más de 8 caracteres o no tiene extensión HEX. Existe daño en algún elemento o circuito integrado de la tarjeta.	Verificar que el nombre del archivo a simular no tenga más de 8 caracteres y que tenga extensión HEX. Cambiar el elemento del circuito por uno nuevo.

Tabla 20 Soluciones a posibles problemas que puede presentar la tarjeta.

A P E N D I C E "D"

LINEAMIENTOS PARA ESCRIBIR UN PROGRAMA EN LENGUAJE ENSAMBLADOR

1. LINEAMIENTOS PARA ESCRIBIR UN PROGRAMA EN LENGUAJE ENSAMBLADOR

Para desarrollar el programa fuente de un archivo se utiliza un editor de textos y se deben tomar en cuenta los siguientes lineamientos para escribir un programa en lenguaje ensamblador:

1.1 El programa fuente, salvo una excepción trabaja por delimitadores de campo, lo que significa que no hay necesidad de seguir un formato estricto al desarrollar el archivo fuente.

1.2 La excepción a la regla se presenta en la definición de etiquetas, que invariablemente deberán empezar en la primera columna.

Etiquetas, operaciones y operandos deben estar con letras mayúsculas para poder ensamblar correctamente el programa.

En general se tienen 4 campos en el archivo fuente:

- a) campo de etiquetas
- b) campo de operaciones
- c) campo de operandos
- d) campo de comentarios

Etiquetas: Estas deben empezar en la primera columna. Acepta caracteres alfanuméricos únicamente y su longitud máxima es de 6 caracteres, sus delimitadores pueden ser ":", espacio en blanco o tabulador vertical.

Operaciones: Aquí se tienen los mnemónicos (instrucciones), así como algunas directivas del ensamblador. Los delimitadores que acepta son espacio y tabulador vertical

Operandos: En este campo tenemos los elementos sobre los que se va a ejecutar la operación. Dependiendo de ésta podemos tener; 0, 1, 2 o 3 operandos. Cuando se tiene más de 2 operandos, éstos se separan por ", ". El campo se termina con fin de línea oprimiendo la tecla Control y las letras L y F de la PC simultáneamente ó con " ; ".

2. DIRECTIVAS DEL ENSAMBLADOR

Se conoce como directivas del ensamblador a las instrucciones que dirigen o determinan una asignación de una función especial para realizar un programa en lenguaje ensamblador. Las directivas más importantes son las que se muestran a continuación.

ORG: Marca un valor absoluto a partir del cual se asigna valores reales a los símbolos y etiquetas.

Sintaxis: ORG operando.
donde operando puede ser una etiqueta o una dirección.

EQU: Sintaxis : Etiqueta EQU operando.
Asigna a etiqueta el valor de operación. Operando puede ser otra etiqueta si ésta se definió con anterioridad.

DB: Sintaxis: (Etiqueta) DB operando.
Reserva espacio en memoria asignando valores a las localidades reservadas. Cuando dichos valores son caracteres, éstos pueden ir entre comillas , o puede reservarse el espacio de toda una cadena colocando la cadena completa entre comillas.

END: Todo programa fuente debe terminar con esta directiva. Lo que haya después de "END", no se tomará en cuenta en el ensamblado. por lo que debe ser la última instrucción en el programa.

A P E N D I C E "E"
ESPECIFICACION ISA PARA LAS TARJETAS IMPRESAS

ESPECIFICACION ISA PARA LAS TARJETAS IMPRESAS

Las tarjetas que se adiciona a la terminal de expansión de la PC deben de estar dentro de las dimensiones estandarizadas por ISA (Industrial Standard Architecture), la cual especifica las dimensiones dadas en la tabla 21.

MAXIMAS DIMENSIONES PERMITIDAS PARA LA TARJETA DE EXPANSION DE LA PC.			
	PC XT	PC AT	UNIDAD
ALTO	10.66	12.19	cm
LARGO	33.52	33.52	cm
ANCHO	1.27	1.27	cm

Tabla 21 Dimensiones Estandarizadas por ISA.

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

GLOSARIO DE TERMINOS

ALGORITMO : Conjunto de reglas bien definidas para resolver un problema en un número finito de operaciones ó procedimiento cuyos pasos especifican el orden de un programa.

ARCHIVO : Conjunto de registros relacionados que guarda un programa; agrupamiento organizado de registros relacionados entre si para guardar información.

ARCHIVO FUENTE : Archivo donde residen los mnemónicos de las instrucciones de un programa. Contiene instrucciones de un programa.

ARQUITECTURA : Organización de interconexión de los componentes de un sistema de cómputo.

ASINCRONO : Transmisión de una señal que no tiene sincronía con otra señal en su operación.

BINARIO : Sistema de numeración de base dos que se usa ampliamente en circuitos digitales. Utiliza los dígitos 0 y 1.

BIT : Unidad básica de la memoria del computador; es una abreviatura del dígito binario (Contracción en inglés de binary unit) y puede valer 0 ó 1.

BUFFER : Circuito integrado con salida reforzada; proporciona mayor capacidad de manejo de corriente, sin ejercer ninguna función lógica sobre la misma; dispositivo de almacenamiento que se usa para compensar la velocidad del flujo de datos de un dispositivo a otro, por ejemplo de un dispositivo de entrada / salida a la CPU.

BUS : Conjunto de líneas conectoras entre varios componentes; circuito que provee un camino de comunicación entre dos o más dispositivos como una CPU y un periférico.

BYTE : Grupo de 8 bits que suele considerarse como unidad.

CHARACTER (CHAR) : Constante que corresponde a una letra o número.

CIRCUITO INTEGRADO (CI) : Grupo de componentes que forman un circuito electrónico completo miniaturizado. El circuito tiene varios transistores y circuitos asociados que se fabrican juntos en una sola pieza de material semiconductor.

CODIGO : Sistema de símbolos y reglas para representar, transmitir y almacenar información.

CODIGO BINARIO : Sistema numérico que utiliza sólo dos dígitos 0 y 1 para expresar todos los números.

CODIGO OBJETO : Representación de información de un programa en forma hexadecimal.

CODIGO FUENTE : El código fuente consta de una serie de mnemotécnicos (instrucciones) y operandos seleccionados por el programador para realizar ciertas acciones, normalmente se utiliza un editor de textos.

CODIGO MAQUINA : Representación de información en forma binaria, utiliza sólo ceros y unos.

CODIGO MNEMOTECNICO : Sistema de abreviaturas diseñado para reemplazar términos complejos que se utilizan en la preparación de programas en lenguaje ensamblador.

COMPILADOR : Programa que traduce las sentencias de un lenguaje de alto nivel a lenguaje de máquina. Esto acelera la ejecución del programa.

COMPUTADORA PERSONAL (PC) : Es una máquina electrónica digital constituida por la unidad de entrada, de salida, aritmética, de control, y la unidad de memoria

COMPUTADORA AT : (Tecnología Avanzada) Computadora Personal más poderosa y más eficaz que la de tipo XT, basados en la arquitectura de los microprocesador 80286, 80386, y 80486.

COMPUTADORA XT : (Tecnología extendida) Computadora Personal basado en la arquitectura del microprocesador 8088.

CONECTOR DE EXPANSION : Conexión especial en el interior de la computadora que se utiliza para añadir tarjetas periféricas y realizar interfaces con la CPU de la computadora.

CONTROLADOR : Dispositivo que puede acoplarse al computador o a otro dispositivo para controlar una señal, generalmente es un circuito integrado (CI).

COPROCESADOR MATEMATICO : Circuito integrado (CI) por ejemplo el 8087; este procesador está dedicado a realizar cálculos matemáticos de alta precisión.

DECODIFICADOR : Circuito combinacional que permite un código decimal a partir de una entrada binaria.

DEPURACION : Detección, localización y eliminación de errores en los programas y/o en las deficiencias de funcionamiento de los equipos.

DIAGRAMA DE FLUJO: Diagrama que utiliza símbolos y líneas interconectadas para mostrar:

a) Un sistema de procedimientos para lograr objetivos (diagrama de sistema).

b) La lógica y la secuencia de operaciones específicas de un programa. (diagrama del programa).

DISCO DURO : Dispositivo giratorio de almacenamiento masivo que utiliza un disco rígido, tiene una capacidad de almacenamiento de información superior que la del disco flexible.

DISCO FLEXIBLE : Dispositivo giratorio, disco sencillo que almacena información para las computadoras personales.

DISPOSITIVO PERIFERICO: Cualquier pieza o elemento que está fuera de la computadora personal (PC), pero que se relaciona o se conecta con ella.

EMULADOR : Dispositivo de lógica almacenada o programa que permite a una computadora ejecutar instrucciones de lenguaje de máquina de otro computador o sistema.

ENSAMBLADOR : Programa que traduce el código fuente al código objeto que contiene ya los códigos numéricos reales para la máquina.

ERROR DE PARIDAD : Error en la verificación de la transmisión de una señal digital, el dispositivo que envía ó agrega un bit de paridad el cual es calculado según el contenido de los bits de datos, el dispositivo receptor verifica que el bit de paridad guarde la relación correcta para los otros bits, si no es así algún error ocurrió durante la transmisión.

HARDWARE : Componentes físicos de un sistema de computación; tal como los dispositivos electrónicos, teclado, monitor, etc.

INTERFASE: Se conoce como interfase o interconexión al dispositivo mediante el cual existe un diálogo entre la CPU y una unidad externa.

INTERFASE PARALELA : En este tipo de interfase los bits de información viajan simultáneamente hacia el periférico.

INTERFASE SERIAL : En este tipo de interfase los bits de información viajan uno después del otro por la misma línea.

INSTRUCCIONES MNEMONICOS : Conjunto de instrucciones para escribir un programa en lenguaje ensamblador.

INDUSTRIAL STANDARD ARCHITECTURE (ISA): Asociación que especificó las dimensiones de las tarjetas de expansión y otras características eléctricas en una computadora personal (PC).

LATCHES : Dispositivo que retiene los datos recibidos para ser utilizada posteriormente.

LENGUAJE : Es la facultad de expresarse por medio de sonidos, escritura o señales para comunicarse.

LENGUAJE "C" : Lenguaje de programación de empleo general, caracterizado por su flexibilidad, por poseer un moderno control de flujo y estructuras de datos, así como un conjunto de operadores muy amplio.

LENGUAJE ENSAMBLADOR : Lenguaje de programación de bajo nivel que utiliza mnemotécnicos para escribir instrucciones.

MEMORIA DE ACCESO ALEATORIO (RAM) : Memoria de lectura-escritura volátil. Pierde su información al quitarle la fuente de alimentación.

MEMORIA DINAMICA : Memoria RAM en la que su contenido se pierde después de un tiempo de 5 a 10 ms. y para poder conservar su contenido es necesario reescribir la información periódicamente cada 2 ms. El proceso de reescritura consiste en leer el contenido y volverlo a grabar en las mismas posiciones; ofrece mayor densidad de integración y menor tiempo de acceso a una célula elemental.

MEMORIA DE SOLO LECTURA (ROM): Memoria que contiene información almacenada permanentemente y no se pierde al quitarle la fuente de alimentación, el contenido de esta memoria se fija durante la fabricación.

MEMORIA DE SOLO LECTURA BORRABLE (EPROM): Es una ROM programada y borrable con luz ultravioleta.

MICROCONTROLADOR: (Circuito integrado de muy alta escala de integración), es un controlador programado que tiene una memoria dentro de su arquitectura y un contador de programa que permite direccionar instrucciones a ejecutar.

MICROPROCESADORES: (Circuito de muy alta escala de Integración, VLSI) : realiza las tareas de la unidad central de procesamiento de una microcomputadora u otro sistema de control automático.

MULTITAREA EN TIEMPO REAL : Realización de tareas simultáneas en el mismo tiempo dentro de una computadora.

PROCESADOR: Dispositivo que interpreta las instrucciones que tiene en memoria; programa de computador que permite manipular instrucciones e información.

PROGRAMA : Grupo organizado de instrucciones que le dicen al computador lo que tiene que hacer.

PROGRAMA FUENTE : Programa donde se originó y reside la información de un trabajo específico.

PROGRAMACION: Realización de un programa para un fin determinado.

PUERTO: Dispositivo que posibilita el diálogo entre dos unidades periféricas, a estos dispositivos se les llama puertos de entrada y salida, (input/output port). Es el lugar donde se conectan los dispositivos de entrada y salida.

PUERTO PARALELO : Puerto de entrada o salida donde los bits que contienen la información viajan simultáneamente.

PUERTO SERIE : Puerto de entrada o salida donde los bits que contienen la información viajan uno después del otro por la misma línea.

PUNTO FLOTANTE : Valor con signo de la forma exponencial.

REGISTRO : Bloque organizado de datos.

RESET : Señal usada para restablecer o inicializar el sistema lógico en el encendido del dispositivo.

SIMULADOR : Representación de un elemento o dispositivo que simula o imita al original.

SINCRONO : Modalidad de la señal que opera en sincronía o en combinación con otra señal.

SISTEMA MINIMO : Sistema constituido por circuitos integrados (CI) principalmente por microprocesador, memorias y dispositivos de entrada y salida (buffers).

SISTEMA OPERATIVO: Programa que controla la operación básica del computador.

TARJETA DE EXPANSION : Tarjeta de circuito impreso con componentes electrónicos para realizar una interfase con la CPU de la computadora.

TARJETA MADRE : Es el sistema donde se encuentra la unidad de procesamiento central del sistema (CPU), que realiza las funciones inteligentes de control, las funciones aritméticas y lógicas de una computadora personal.

UNIDAD DE PROCESAMIENTO CENTRAL (CPU) : Es la parte inteligente de un microprocesador que controla a una computadora.