

18
2ej



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

MODULO DE EVALUACION DE PROPOSITO
GENERAL BASADO EN UN
MICROCONTROLADOR

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :
DANIEL CHACON MORENO

Director de Tesis: Ing. Alberto Templos Carbajal



MEXICO, D. F.

TESIS CON
FALLA DE ORIGEN

1981



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE

INTRODUCCION	i
CAPITULO I	
<i>SISTEMATIZACION DEL DISEÑO Y CONSTRUCCION DE UN PROTOTIPO ELECTRONICO: EL MEV</i>	1
I.1 Fases del desarrollo de un prototipo electrónico	1
I.2 Fase de planificación de un prototipo	2
I.3 Especificación del sistema	4
I.4 Análisis del sistema	7
I.5 Desarrollo del hardware	10
I.6 Desarrollo de software	10
I.7 Integración del sistema	11
CAPITULO II	
<i>JUSTIFICACION DEL USO DE UN MICROCONTROLADOR</i>	13
II.1 Valoración del uso actual de los microcontroladores	14
II.2 Elección del microcontrolador a utilizar	17
CAPITULO III	
<i>DISEÑO DEL SISTEMA GENERAL DE DESARROLLO (MEV)</i>	21
III.1 Arquitectura del microcontrolador	23
III.1.1 Organización de la memoria	26

III.1.2 Registros de Funciones Especiales (SFR's)	36
III.1.3 Circuito oscilador	39
III.1.4 Ciclos de la unidad central de procesamiento	41
III.1.5 Estructura y operación de los puertos	42
III.1.6 Sistema de interrupciones	45
III.1.7 Funcionamiento y características de los Contadores/temporizadores	49
III.1.8 El puerto serial de comunicación	52
III.1.9 Modo de operación Paso-A-Paso	56
III.1.10 Condiciones de reinicialización	57
III.1.11 Auto-inicialización	58
III.1.12 Registro de control de consumo de potencia(PCON)	58
III.2 Diseño electrónico del sistema MEV	61
III.2.1 Organización general del MEV	62
III.2.2 Diseño del módulo 1: El Microcontrolador	65
III.2.3 Mapa de memoria del MEV	73
III.2.4 Diseño de los módulos de Entrada/Salida	77
III.3 Implantación del MEV	90
III.3.1 Conformación Mecánica	90
III.3.2 Rentabilidad de la implantación	95
III.4 Pruebas de funcionamiento	95
III.4.1 Medios de pruebas	95
III.4.2 Principio general de procedimiento	96
III.4.3 Casos de fallas	97
III.4.4 Autodiagnóstico	99

CAPITULO IV

<i>DISEÑO DEL PROGRAMA MONITOR DEL SISTEMA DE DESARROLLO: MEV</i>	101
IV.1 Conjunto de instrucciones del Microcontrolador	102
IV.1.1 Modos de direccionamiento	106
IV.2 Diseño del programa monitor	109
IV.2.1 Sistema monitor general	110
IV.2.2 Diseño del sistema de inicialización del MEV	110
IV.2.3 Diseño del sistema de autodiagnóstico	113
IV.2.4 Diseño del sistema de comandos de línea	114
IV.3 Programación del sistema monitor del MEV	122
IV.3.1 Programación del sistema de inicialización	123

IV.3.2 Programación del sistema de autodiagnóstico	124
IV.3.3 Programación del sistema de comandos de línea	125
IV.3.4 Ensamblado de los programas del monitor	125
IV.4 Pruebas de funcionamiento	127
IV.4.1 Plan de pruebas del software	128
IV.4.2 Estrategias de las pruebas	130

CAPITULO V

APLICACIONES Y CONCLUSIONES 133

V.1 Simulación del movimiento oscilatorio 134

V.1.1 Características de un péndulo 134

V.1.2 Programa de simulación del movimiento de un péndulo 135

V.2 Adquisición de datos vía RS-232C de una computadora personal 137

V.2.1 La comunicación serial 137

V.2.2 Características de la interface serial de una computadora personal 139

V.2.3 Hardware adicional para el standard RS-232C 139

V.2.4 Programa de adquisición de datos vía RS-232C 140

V.3 Monitoreo de una variable física 144

V.3.1 Características de la conversión A/D 145

V.3.2 Frecuencia máxima de muestreo 145

V.3.4 Principio general de conversión analógica/digital 146

V.3.5 Conexión del ADC803 al MEV 147

V.3.6 Hardware adicional para el monitoreo de la variable física 148

V.3.7 Programación para el procesamiento de la variable física 148

V.4 Conclusiones 150

APENDICE A

DESCRIPCION DE PINES DEL MICROCONTROLADOR 8031 153

APENDICE B <i>SIMBOLOS CARACTERISTICOS DEL CONJUNTO DE INSTRUCCIONES DE LA FAM. MCS-51</i>	157
APENDICE C <i>INTERFACE ESTANDAR RS-232C (CCITT V.24/ISO 2110)</i>	159
BIBLIOGRAFIA	161

INTRODUCCION

El objetivo primordial de este trabajo de tesis consiste en elaborar un módulo de evaluación de propósito general teniendo a un microcontrolador como circuito central de control del sistema. El beneficio principal a obtener de un microcontrolador, en este caso, es adherir fácilmente nuevas características al prototipo sobre los "kit's" tradicionales, y poder darle una nueva personalidad haciendo pequeños cambios, sin afectar las conexiones del circuito, lo que es imposible tratar de hacer en alambrados con circuitos de control de tecnología TTL.

Introducir un microcontrolador nos permitirá tener un módulo diseñado para funciones de control industrial, reemplazando gran parte de la lógica digital, en consecuencia se gastará la mayor parte de tiempo en el diseño electrónico de dispositivos de Entrada y Salida (E/S), en lugar de perderlo en las etapas básicas como la sincronización.

Sin lugar a dudas, las perspectivas del mercado de dispositivos electrónicos como son microprocesadores, microcontroladores, interfaces y otros, marcan un futuro promisorio, donde la integración de cada vez más funciones dentro de un espacio reducido marca la pauta, así como la relación costo/eficiencia para el consumidor.

El campo de aplicación de los microcontroladores es cada día más variado y accesible, los podemos encontrar en teclados,

controladores de discos duros, en instrumentación médica, en control industrial, robótica, telecomunicaciones, procesamiento de señales, imágenes y voz y en el área militar (navegación, control de misiles teledirigidos), etc. Los campos de mayor alcance son los mostrados en la figura 1.

El contenido, ya propiamente dicho, consta de cinco capítulos, donde tenemos que:

En el primer capítulo se establecen las bases y los objetivos específicos a alcanzar, se desarrolla en forma sistemática las fases de desarrollo de un prototipo electrónico, desde su planeación -conceptualización- hasta las etapas de integración de la parte electrónica -hardware- y la parte de programación -software-, como bases para lograr un éxito en el diseño, construcción y lanzamiento de un producto.

Dentro del segundo capítulo se asientan las diferencias entre los microprocesadores, los microcontroladores y las microcomputadoras, así como se justifica el uso del microcontrolador como núcleo del sistema, las ventajas de ello y las razones de costo/efectividad a obtener. Finalmente se elige el microcontrolador de la gran variedad de controladores existentes en el mercado, basados en el soporte tanto de programación como de dispositivos electrónicos.

El tercer capítulo presenta características particulares del microcontrolador y su familia, además de los requerimientos necesarios para formar el primer diseño, después se llevan a la realidad y se termina haciéndole pruebas utilizando equipo alterno de apoyo como son los osciloscopios, el analizador de estados lógicos y otros.

Con el capítulo cuatro se muestran las características de programación del microcontrolador, su conjunto de instrucciones, sus modos de direccionamiento y el uso de simuladores y ensambladores para probar y verificar el código de máquina obtenido, útil para sistemas de este nivel; se concluye con el diseño de un programa que controla e inicializa al sistema, el cual se implementa y se hacen las pruebas necesarias.

DSP DE PROPOSITOS GENERALES	GRAFICOS / IMAGENES	INSTRUMENTACION
Filtrado digital Convolución Correlación Transformada de Hilbert Transformada rápida de Fourier Filtros adaptativos Generación de formas de onda Creación y manejo de ventanas	Rotación en 3a. Dimensión Visión de robots Transmisión de imágenes Reconocimiento de patrones Mapeo de animación digital Estaciones de trabajo Mejoramiento de imágenes Comprensión de imágenes	Análisis de espectros Generación de funciones Comparación y búsqueda de patrones Procesamiento sísmico Análisis transitorio Filtros digitales
PROCESAMIENTO DE VOZ	CONTROL	MILITAR
Correo por voz Reconocimiento de voz Verificación oral Mejoramiento de voz Síntesis de voz Texto-a-voz	Controladores de discos Servocontroladores Controladores de robots Control de impresoras láser Ingeniería de control Controladores de motores	Comunicaciones secretas Procesamiento de señales de radares y sonares Navegación Guía de misiles Modems de radiofrecuencia
TELECOMUNICACIONES		AUTOMOVILISTICA
Video conferencias Codificadores ADPCM Múltiplexado de canales Modems de 1200 a 19200 bps	Ecuilibradores adaptativos Encriptación de datos Fax y telefonía celular Telefonía parlante	Ing. de control Análisis de vibraciones Comandos por voz Posicionamiento global
CONSUMO POPULAR	INDUSTRIA	MEDICINA
Juguetes Educativos Sintetizadores de música Audio / TV digitales	Robótica industrial Control numérico Accesos de seguridad	Monitoreo de síntomas Diagnosticadores Monitoreo de fetos

FIGURA 1: Campos típicos de aplicaciones de los MICROCONTROLADORES
(DSP = Procesamiento digital de señales).

En el último capítulo, el cinco, se exponen un conjunto de aplicaciones y las conclusiones finales. Damos a conocer las metas alcanzadas, los obstáculos hallados, los resultados obtenidos y las perspectivas, después de un contacto verdadero con un microcontrolador.

Podemos decir que se han contemplado ambos campos de la ingeniería de computación, como son el "software", el "hardware" y las aplicaciones en el mundo real, de una manera muy concreta.

CAPITULO I

SISTEMATIZACION DEL DISEÑO Y CONSTRUCCION DE UN PROTOTIPO ELECTRONICO: EL MEV

Un sistema basado en un microprocesador comprende, en muchos casos, un proceso técnico para automatizar un proceso, controlar operaciones variadas, producir o transformar energía, pero casi siempre sin que exista la intervención humana. Cuando un sistema es controlado por una computadora se tienen dos subsistemas: El objeto a controlar y el subsistema de control, que se conectan mediante un canal de comunicaciones, por donde corren datos y órdenes en ambos sentidos.

I.1 FASES DEL DESARROLLO DE UN PROTOTIPO ELECTRONICO

Para diseñar y construir un sistema basado en un microprocesador se requiere de desarrollo de "hardware" y de "software"; dentro de la primera parte, una división funcional es: unidad central de proceso (CPU), memoria e interfaces de los

periféricos, cuyo tamaño depende del uso y los requerimientos del sistema. Existen unidades estandarizadas y se les denomina subsistema de microcomputación. Los grupos modulares naturalmente no están diseñados de manera óptima para todos los usos y pueden contener funciones redundantes, que en un diseño especial se evitan.

La experiencia ha demostrado que la necesidad de una sistematización en el desarrollo de productos, basados en la modularización por funciones, repercutirá en una mayor organización, documentación y eficiencia durante la vida útil del sistema.

Las fases de desarrollo que apoyan en la sistematización de la elaboración de un prototipo se reseña en la figura I-1. En la fase de planificación, primero se reúnen ideas que se fijan a las especificaciones del sistema. El análisis del sistema que se efectúa después, investiga la relación sistema-objeto y, con los conocimientos obtenidos, se diseña funcionalmente el sistema de control. A continuación se hace el desarrollo físico del prototipo -hardware- y de su programación -software-, que puede ser en forma paralelo. Se sigue con la integración de ambas partes -unión paso a paso del hardware y del software-. Se autoriza la producción del prototipo resultante después de pasar por minuciosos ensayos.

I.2 FASE DE PLANIFICACION DE UN PROTOTIPO

*"A todo proceso de desarrollo
antecede la idea del nuevo producto."*

La planificación se inicia con la conceptualización del objetivo o función del producto a realizar, al reunir ideas, se tiene la convicción o el deseo de llevar un diseño ideal al mundo real. Existen elementos gráficos que apoyan eficazmente todas las fases de desarrollo, éstos medios auxiliares tienen el propósito de ayudar a hacer más comprensible una estructura complicada; de una visualización global se va llegando a los detalles por medio de un afinamiento. En el diseño por diagramas en bloques se toma en cuenta en primer lugar, de qué bloques funcionales está compuesto el sistema y como están relacionados entre sí. Muy a

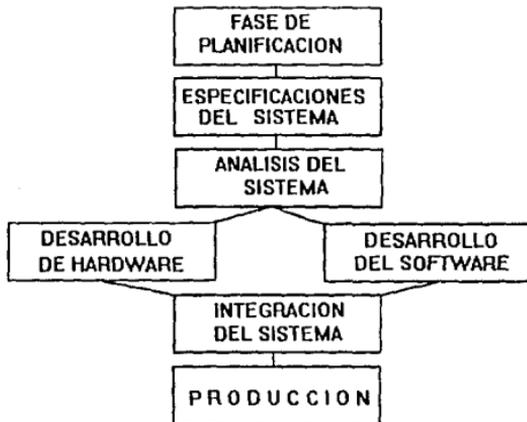


FIGURA 1-1: Fases de Desarrollo de la construcción de un Sistema basado en un Microprocesador

menudo, la primera noción del nuevo sistema es imprecisa, con funciones que aún deben aclararse, todo logrado después de varias consultas de aporte de ideas, que logran cristalizar el perfil de trabajo sobre la concepción ideal. La factibilidad de un lanzamiento exitoso de un producto al mercado depende de:

- **Análisis de mercado:**
 - Brecha existente en el mercado
 - Reducir costos
 - Incrementar la participación en el mercado
- **Plan económico:**
 - Rentabilidad
 - Costo/utilidad
 - Personal y capacitación
- **Análisis técnico:**
 - Requerimientos de Hw (Hardware) y Sw (Software)
 - Volumen de Hw y Sw

El estudio nos determinará si en realidad hay demanda para un producto de este tipo.

Construir un **Módulo de Evaluación** de propósito general -nos referiremos a él como MEV-, consiste en elaborar una microcomputadora, en su nivel básico, donde el usuario podrá manejar completamente un procesador específico para realizar diferentes tareas (comunicación, procesos y control), utilizará las ventajas de tener a disposición un sistema completo dedicado únicamente a una labor principal y toda la potencialidad de un **MICROCONTROLADOR**.

1.3 ESPECIFICACION DEL SISTEMA.

"Debe ser completa, pero dejará suficiente libertad para la posterior realización."

Una vez ya concretada la idea del prototipo, se pasa ahora, a las especificaciones del sistema y consisten en una descripción detallada del producto que se desea realizar. El documento resultante son las especificaciones del producto, donde se menciona desde las funciones generales, hasta las normas acreditadas a usar como las interfaces, los estándares, el tipo de diálogo usuario-máquina y otros. Los elementos de las especificaciones del producto se dan en los puntos siguientes.

1.3.1 DESCRIPCION FUNCIONAL DEL SISTEMA

El **MODULO DE EVALUACION** de propósito general es un computador monoplaca (single-board computer) o una computadora compacta, con todo lo necesario para requerimientos y medios de capacidad de memoria e interfaces para periféricos. Las características de organización que se tendrán en cuenta son:

- Todos los conductores importantes deben ser accesibles, especialmente las líneas de datos, direcciones y control -buses-.
- Se proveerán lugares extras para memorias -zócalos-.
- Los módulos de E/S deberán ser programables.
- Elaborar programas de verificación, autoejecutables durante la inicialización del sistema.

- Preveer una reserva de alimentación para casos de fallas.
- Se utilizarán exclusivamente componentes comunes, para los cuales existan también proveedores alternativos (segunda fuente).

En cuanto a los alcances del prototipo se tiene que:

- Contiene una CPU de palabra de 8 bits
- Funcionará a una frecuencia óptima para comunicación con una computadora personal IBM compatible (9600Khz).
- Con un zócalo para EPROM de 2 ó 4 kbytes.
- Con una RAM alimentada por una tensión auxiliar para casos de fallas.

Ahora bien, para probar o evaluar un programa en forma rentable, se dispondrá de un programa monitor (de 2 kbytes), almacenado en una EPROM, que solamente requiere enchufarse sobre la computadora compacta -el MEV-. A través de los pines del puerto serial y con solamente 2 convertidores de nivel externo se podrá conectar el MEV a una computadora personal IBM compatible. Para efectuar un diálogo entre el usuario y la microcomputadora (programa monitor) se utiliza un teclado de funciones y una forma visual de desplegado de datos.

Como principales áreas de aplicación se ha previsto: medición, control y regulación; bajo la denominación de programa MONITOR se entiende a un programa relativamente pequeño -del orden de los 2 kbytes-, que debe apoyar en la inicialización, prueba y comunicación con el usuario, el "software" y el "hardware". Sus funciones más importantes son el desarrollo o ejecución paso-a-paso de un programa prueba, así como la lectura y descripción de posiciones de memoria y registros internos del procesador. Finalmente, para el control de un proceso real se contemplan convertidores A/D, como comunicación entre un fenómeno continuo y su manipulación mediante un sistema de microcomputadora.

1.3.2 DESCRIPCION DE LAS INTERFACES

La determinación tanto de las funciones de cada interfaz, como la forma de los datos (información de E/S y las señales de control),

es importante para la comunicación hombre-máquina, necesaria en el manejo del equipo.

En la comunicación visual se usarán 8 despliegues, enumerados de izquierda a derecha, como V0 hasta V7, la información desplegada es: el dato en V4 a V5 y la dirección que contiene ese dato está en V0 a V3; si el dato es de 16 bits (por ejemplo el dato del Contador de Programa), se ocupan los despliegues V6 y V7; si la información es de un registro, en los primeros cuatro despliegues se mostrará el nombre del registro. En la entrada de información se usa un teclado multifunciones, dividido en dos sectores: las teclas de control y el teclado hexadecimal.

Para la comunicación con la PC IBM compatible, se utilizan las especificaciones de la norma RS-232C, para simplificar los formatos y acrecentar las posibilidades de interconexión con sistemas mayores. Para casos de desviación de la norma y para solucionar algún problema específico, se exigen las especificaciones verificables, complementadas con diagramas de tiempo.

1.3.3 DESCRIPCION DEL DISPOSITIVO DE PRUEBAS

Contiene métodos de verificación y las correspondientes entradas y salidas de pruebas en caso de fallas. Un dispositivo de autodiagnóstico (selftest) para el MEV, simplificará y acelerará el mantenimiento y la reparación del equipo. Mediante una autoprueba que se encuentra dentro del programa monitor se podrán verificar algunos dispositivos como la RAM, los despliegues, el teclado, el programa monitor y las demás interfaces. El método para verificación de la RAM consiste en:

- *Barrer toda la RAM*
- *Escribir en todas las localidades, solamente ceros*
- *Leer en todas las localidades.*
- *Si los datos leídos son diferentes a cero avisar del error de lectura.*

Para sistemas basados en procesadores es relativamente económico, incorporar un dispositivo de diagnóstico. Dentro de la implementación y después, existen procesos de pruebas de funcionamiento (mencionados en el capítulo III, inciso 4), donde

antes de poner en marcha por primera vez al prototipo y de integrarle el programa monitor final, son probados todos los elementos electrónicos -decodificadores, latch's, teclas, etc.-, para evitar fallas y daños a circuitos mayores -procesador, memorias, etc-.

1.3.4 DESCRIPCION DE LAS CONDICIONES DE TRABAJO Y AMBIENTALES

Una serie de recomendaciones, para el perfecto funcionamiento del MEV, son dadas a continuación y estan basadas en las normas más comunes:

- Evitar la transmisión de cargas estáticas al sistema, que dañarían a los circuitos MOS como las memorias y el procesador.
- Aunque el MEV contendrá un dispositivo de protección, es necesaria una alimentación regulada.
- Los rangos de temperatura y humedad son los mismos que para los circuitos integrados constitutivos: de 0 a 70°C.

I.4 ANALISIS DEL SISTEMA

"Proceder según el principio de la refinación paso a paso."

Los productos controlados por microprocesadores se pueden dividir en sistema-objeto y sistema control. En el análisis del sistema y después de la primera subdivisión se van vislumbrando los detalles. El MEV está constituido por: Un microprocesador, las memorias y las interfaces de E/S (figura 1.2).

Una vez modularizado por funciones, podemos obtener las organizaciones arbóreas que se muestran en las figuras 1-3, 4 y 5, comunicadas a través de los canales de comunicación.

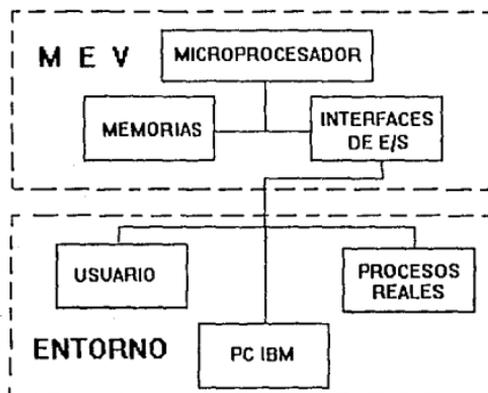


FIGURA I-2: Entorno de un sistema basado en un microprocesador.

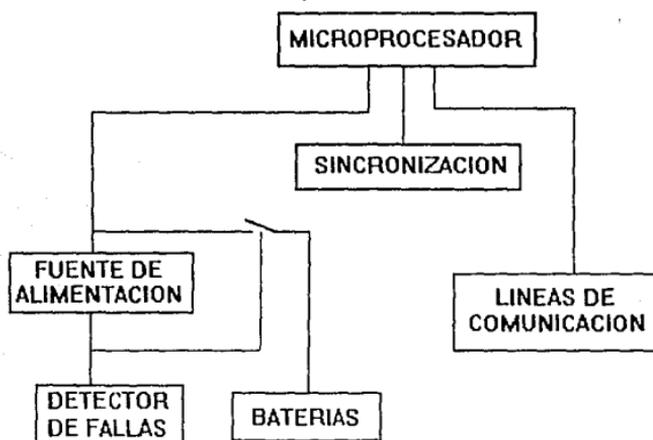


FIGURA I-3: Organización funcional del Subsistema de Microprocesador.

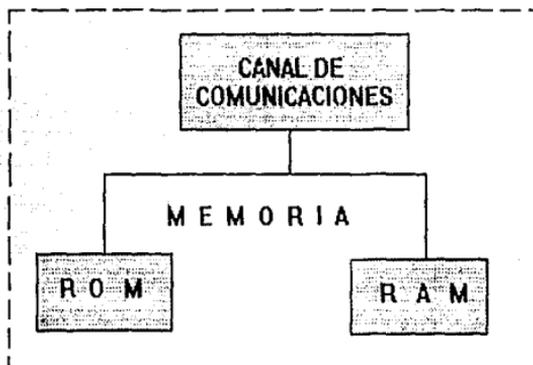


FIGURA I-4: Organización funcional del Subsistema de Memoria.

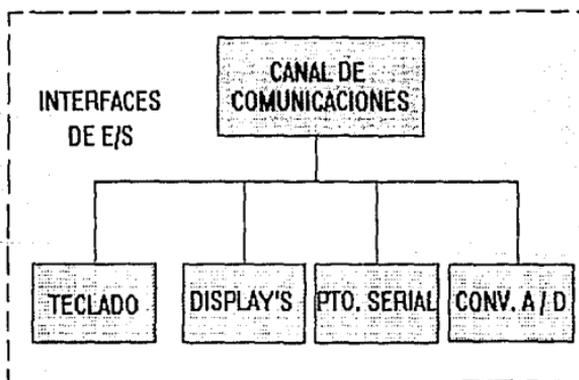


FIGURA I-5: Organización funcional del Subsistema de E/S.

I.5 DESARROLLO DEL HARDWARE

"La estructura técnica será modular y deberán contener funciones completas."

Después de la conceptualización, las funciones analizadas se deben llevar a la realidad. Se utiliza la modularización, comenzando con los requerimientos de E/S, se analiza la formación de los subsistemas y se llevan a la implementación. En el capítulo III se hace el desarrollo físico detallado del prototipo hasta finalizar con las pruebas.

I.6 DESARROLLO DE SOFTWARE

"La calidad del software se determina por su aceptación y su facilidad de ampliación."

El software del MEV consiste de dos partes:

- El programa monitor.
- El conjunto de programas de aplicación.

El sistema debe inicializarse y crearse el ambiente interno óptimo para estar en condiciones de establecer una comunicación -prompt- y relacionarse con el usuario, todo esto lo hace el programa monitor, que tiene características determinantes:

- No se pierde al desconectarse o fallar la alimentación.
- Hace que el MEV se comporte de una manera preestablecida.
- Tiene rasgos de aceptación por los usuarios -amigable-.
- Es fiable, los datos desplegados son realmente los que se encuentran en el sistema -memorias y registros internos-.
- Es fácil de utilizar.

Dentro del conjunto de programas de aplicación se tiene una biblioteca de ejemplos para solucionar determinados problemas y comunicar al MEV con un sistema real, basados en la necesidad de demostrar la potencialidad del prototipo resultante. El diseño, implementación y pruebas del sistema de programación es llevado a cabo en el capítulo IV, con todo detalle.

I.7 INTEGRACION DEL SISTEMA

"El desarrollo del hardware y del software no son caminos separados, en la integración se encuentran"

La continua cooperación entre las partes electrónicas y la programación, permite apoyo mutuo; así se deben saber los requerimientos de memoria -por el lado del hardware- y las direcciones definitivas de los canales de E/S -por el otro, el software-. La prueba de integración (figuras I-6 y I-7) se lleva a cabo cuando se examinan en conjunto los módulos, ya terminados, físicos y de programación.

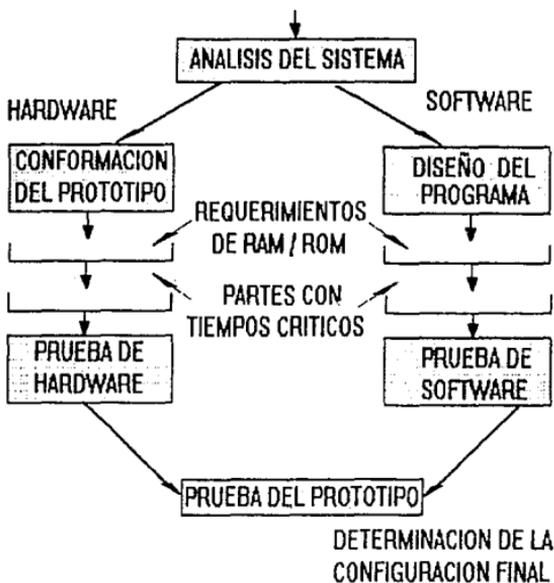


FIGURA I-6: Cooperación y apoyo entre el hardware y el software.

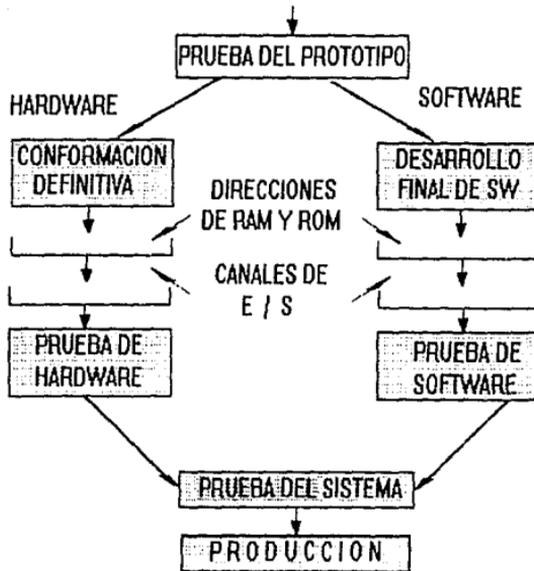


FIGURA I-7: Cooperación y apoyo entre el hardware y el software. (cont').

CAPITULO II

JUSTIFICACION DEL USO DE UN MICROCONTROLADOR

La integración de funciones más complejas, en menores espacios físicos para diversas aplicaciones es una de las ideas vanguardistas que muchas compañías fabricantes de dispositivos electrónicos han estado llevando a cabo, lo que repercute en un mercado ampliamente dotado de innovaciones como en los últimos años lo han sido los microcontroladores.

Realmente han quedado atrás aquellos días donde un proyecto complejo requería de grandes cantidades de circuitos TTL y muchas tabletas, como resultado de la aparición de los *microprocesadores*.

Así los microprocesadores se convirtieron en el núcleo de los sistemas ya que controlaban a todos los CI's (circuitos integrados), procesaban los datos, habilitaban al periférico adecuado y se comunicaban al exterior proporcionándonos los resultados.

Y una vez más, en el intento por integrar un número mayor de funciones de un sistema han surgido una gran variedad de dispositivos que se han popularizado comercialmente por su precio, funcionalidad y eficiencia, los llamados MICROCONTROLADORES.

II.1 VALORACION DEL USO ACTUAL DE LOS MICROCONTROLADORES

Es muy importante tener bien claro las diferencias entre una microcomputadora, un microprocesador y un microcontrolador, ya que si por su origen y evolución provienen de la misma familia, las funciones y los componentes que los integran son diferentes, como sus alcances y perspectivas, así lograremos tener una valoración de uno con respecto a los otros.

Un microprocesador contiene una unidad aritmética-lógica (ALU) y una unidad de control, contenida en un sólo circuito integrado, es decir, es solamente la unidad central de procesamiento (CPU) de una computadora, sin memoria, ni unidades de entrada/salida, ni periféricos necesarios para tener un sistema completo. Lo que nos indica que una CPU no puede comunicarse al exterior por sí solo. Aparecieron a partir del año de 1970, como un producto de la avanzada tecnología en semiconductores que permitan su fabricación en un sólo circuito integrado. En cuanto a ejemplos tenemos muchos como el Z80, el 8088, 80486 y 68000, que son de 8 y 16 bits, comúnmente llamados "micros", cuya organización puede observarse en la figura II-1.

Una microcomputadora es la combinación de un microprocesador con unidades de E/S y de memoria, además puede manejar otros periféricos para comunicarse con el mundo exterior, en la figura II-2 se observa la organización de una microcomputadora. También nos referimos a ellos con el prefijo "micro", pero la diferencia es tremenda.

Muchas compañías han sumado las funciones de los periféricos en el mismo estrato que la CPU tiene, para hacer una

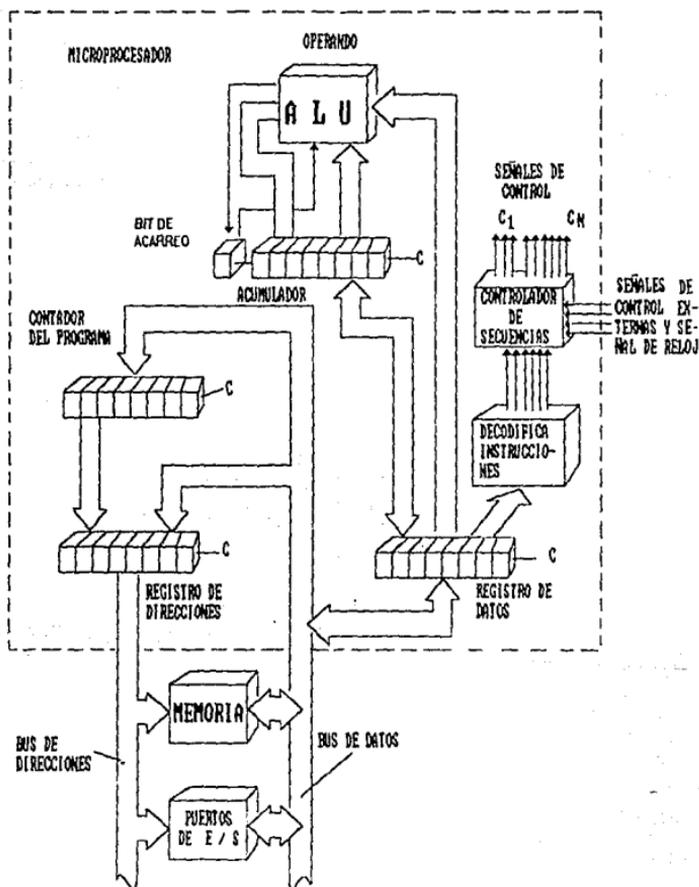


FIGURA II-8: Organización interna de un MICROPROCESADOR.

microcomputadora completa, son llamadas *microcomputadoras* en un sólo "chip" (encapsulado), para diferenciarlas de las microcomputadoras basadas en un microprocesador.

Las microcomputadoras en un sólo "chip" son diseñadas para dispositivos pequeños, donde no se necesitan todas las funciones de un sistema de computación completo como en aplicaciones específicas de control, donde aún con pocos CI's es necesario el soporte de una CPU como el Z80, que sería demasiado.

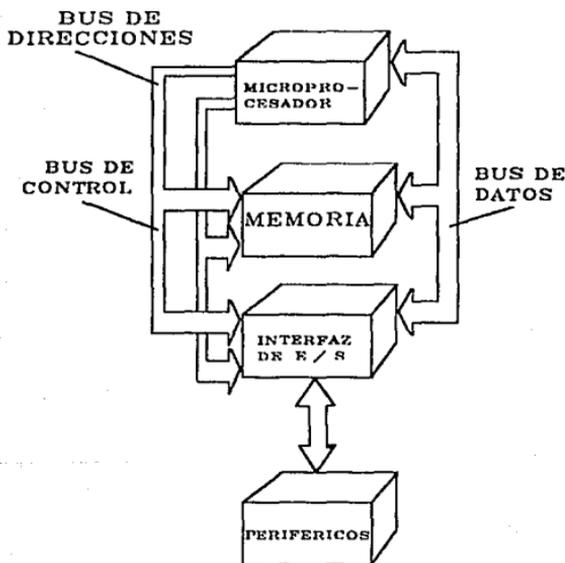


FIGURA II-9: Diagrama de bloques funcionales de una MICROCOMPUTADORA.

En vez de ello, se puede usar una microcomputadora de un sólo "chip" para el manejo de ciertas actividades. Cuando una microcomputadora de un sólo "chip" es usada en sistemas de

control industrial se le llama microcontrolador. Pero básicamente no existe diferencia entre una microcomputadora y un microcontrolador.

Actualmente, los microcontroladores son usados en todos los teclados existentes, en interfaces de discos, en fin, en dispositivos donde se requieren de circuitos de bajo nivel o la atención de una CPU cada vez que es usado el subsistema.

En el mercado podemos hallar microcontroladores que contienen en un sólo circuito varios puertos de comunicación exterior (seriales o paralelos), RAM interna, ROM (o alguna variedad de ese tipo de circuitos), circuitos osciladores, uno o más contadores/temporizadores, etc., todo esto que es necesario en sistemas complejos y que ahora solamente se le agregarían las unidades de entrada/salida como un teclado y una forma de despliegado de información.

II.2 ELECCION DEL MICROCONTROLADOR A UTILIZAR

Los microcontroladores pueden ser de 8, 16 ó 32 bits, por ejemplo el 8031 de Intel, el 68040 de Motorola y el TMS320C17 de Texas Instruments.

El **8031 de Intel** es un microcontrolador clásico perteneciente a la familia MCS-51 que tiene muchos miembros, cada uno adaptado a un tipo específico de sistema. Esta generación es más avanzada que la 8742 y proviene de la 8048 de 4 bits. El **8051** es el microcontrolador representativo de esa familia y tiene dos parientes muy cercanos el **8751** y el **8031**, y un primo, el **8052** (figura II-3). Todos tienen la misma CPU, la misma arquitectura de RAM interna, los mismos temporizadores/contadores, puertos paralelos y un puerto serial iguales. Las diferencias son en cuanto a cantidad de RAM y en la existencia y tipo de ROM.

El **8051** tiene 4k de ROM, mientras que en el **8751** la ROM es reemplazada por una EPROM. El **8031** usa memoria de sólo

lectura externa (como todos los procesadores), ya que no contiene ROM, tiene 4 puertos paralelos y usa dos de ellos para hacer acceso al programa del exterior.

Aún cuando un microcontrolador muy sencillo como el 8031 puede funcionar como un microprocesador de un circuito expandido, el número de circuitos eventualmente necesarios para expandir la memoria de E/S es todavía considerablemente menor que usando un microprocesador y CI periféricos.

Además las compañías de dispositivos electrónicos han liberado otros circuitos con tecnologías de bajo consumo de potencia, más memoria interna, más líneas de E/S, como el 80C31 que es la versión en tecnología CMOS del 8031.

Para grandes volúmenes, es preferible usar un 8051, ya que aunque el grabado de la ROM es de fábrica con un costo de \$3000 con pedidos de 1000 chips, lo hace poco práctico en sistemas de prototipos de bajo volumen. Se tiene el 8751 con las mismas características que el 8051, excepto que contiene una EPROM reemplazando a la ROM, grabar un programa así como borrarlo es más sencillo y puede usarse como producto final, sin embargo, el costo de un 8051 es de \$4 a \$5 por pieza, mientras que del 8751 es de \$25 a \$40 por pieza.

El 8031 no tiene almacenamiento interno de programa, por lo que debemos incluir una EPROM externa y un "latch" para demultiplexar direcciones y datos. Considerando los bajos precios de las EPROM's, el alto precio del 8751 y el bajo volumen del proyecto que impide usar un 8051, el 8031 es una alternativa muy viable a pesar de los dos circuitos adicionales necesarios.

Para sistemas pequeños, la combinación 8031-EPROM tiene un costo efectivo menor que un 8051, por lo tanto es de este tipo el sistema que se desarrollará, que basado en un microcontrolador de un sólo chip, sus circuitos internos reemplazan toda la lógica digital que normalmente colocamos para el control, sincronización, etc., solamente necesitaremos el teclado, el desplegado de datos y resultados, relés, interruptores y en fin el producto final de E/S para usos específicos.

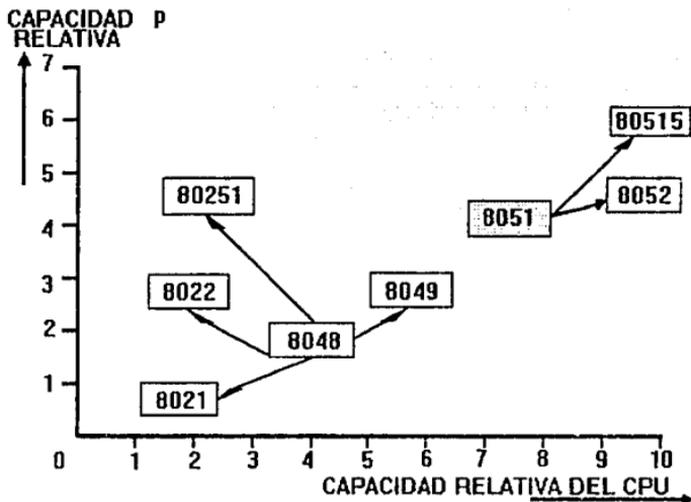


FIGURA II-10: Comparación de los microcontroladores de un sólo chip según su capacidad P (funciones periféricas integradas) y su capacidad de CPU (velocidades etc.).

CAPITULO III

DISEÑO DEL SISTEMA GENERAL DE DESARROLLO (MEV)

Los *microprocesadores* son dispositivos que necesitan de otros circuitos llamados interfaces para poder comunicarse con el exterior, la mayor parte de ellos requieren un sistema de sincronización y una circuitería de reloj extra. Actualmente se han hecho muy populares los microcontroladores, como el 8031, que pertenece a la familia de microcontroladores de 8 bits MCS-51, diseñados y producidos por Intel, así vemos que el 8031 es la versión sin ROM del 8051, cuyo alto "performance" en un solo circuito y su característica de "stand-alone", ha permitido que sea utilizado en aplicaciones más complejas superando a su antecesor, el 8048, como en el manejo de periféricos inteligentes, instrumentación y control industrial. Está provisto de un "hardware" de arquitectura mejorada y un set de instrucciones fácilmente manipulable que lo hacen uno de los controladores de un costo efectivo y poderoso, muy popular sobre todo en las versiones 80C31 y 80C51 de tecnología CMOS, que ofrecen un consumo más bajo de potencia y mayor rapidez.

Los microcontroladores 8031 y 8051 fueron diseñados para aplicaciones de control secuencial con 8 bits, con algunas similitudes de la familia MCS-48, pero son de 2 a 5 veces más rápidos e incluyen más periféricos de comunicación exterior.

Además, la familia MCS-51 de Intel es el standard industrial para los microcontroladores de 8 bits de alto "performance", cuya arquitectura está optimizada para aplicaciones de control en tiempo real, como en instrumentación médica y control de frenos para automóviles.

En cuanto al soporte de software existente se puede contar con el ASM-51 y PL/M-51 en versiones para computadoras personales y con ventajas como la relocalización y ligado; el ICE-5100 es un emulador para diseño de sistemas en tiempo real con 16 Mhz, con lenguaje de alto nivel e instalable en computadoras personales AT.

Su procesamiento de bytes y sus operaciones numéricas sobre estructuras de datos sencillas son facilitadas por los diversos modos de direccionamiento para acceso a RAM. El conjunto de instrucciones incluyen operaciones aritméticas de 8 bits como la multiplicación y la división. La característica de procesador booleano permite la manipulación directa bit-a-bit, útiles en pruebas para sistemas de control. La eficiencia del microcontrolador se basa en la tecnología usada y en un set de instrucciones que consiste en un 44 % de instrucciones de un byte, un 41 % de 2 bytes y un 15 % de 3 bytes.

De manera típica, con un cristal de 12 Mhz, el 58 % de las instrucciones son ejecutadas en 1 microsegundo, el 40 % en 2 microsegundos y la división y la multiplicación en 4 microsegundos solamente.

La compañía AMD (siglas de Advanced Micro-Devices) ha usado al 8051 como base para ir lanzando en el mercado nuevos dispositivos, anexándoles nuevas funciones, incrementando el "performance", el costo/rendimiento y la integración de sus productos derivados. Así el 80C521 tiene el doble de memoria interna tanto de RAM como ROM y contiene un convertidor A/D (analógico/digital) de 8 bits. También existe una versión ideal para aplicaciones de control de motores, el 80514 con su

convertidor A/D y un temporizador-guardián, que evita anomalías de programación, eventos externos inesperados y condiciones de entrada inválidas.

III.1 ARQUITECTURA DEL MICROCONTROLADOR

El microcontrolador a utilizar pertenece a la familia de dispositivos electrónicos INTEL MCS-51 mostrada en la tabla III.1.

NUMERO DEL CIRCUITO	VERSION SIN ROM	VERSION CON EPROM	CANT. DE ROM (BYTES)	CANT. DE RAM (BYTES)	CONTADORES 16 BITS	TECNOLOGIA DE FABRICA
8051	8031	(8751)	4K	128	2	HMOS
8051AH	8031AH	8751H	4K	128	2	HMOS
8052AH	8032AH	8752BH	8K	256	3	HMOS
80C51BH	80C31BH	87C51	4K	128	2	CHMOS
83C51FA	80C51FA	87C51FA	8K	256	4	CHMOS
83C152	80C152	87C152	8K	256	2	CHMOS

TABLA III-1: Microcontroladores de la familia MCS-51 de INTEL.

Los microcontroladores que forman a la familia MCS-51 son dispositivos de 16 líneas para direccionamiento de memoria, con un bus de datos de 8 bits, que en sus inicios se construían basados en la tecnología HMOS I, sin embargo, actualmente las versiones que se hallan en el mercado son de tipo HMOS II, cuyas ventajas se encuentran en el ahorro de consumo de energía y la velocidad de operación.

Ahora bien, con la justificación dada anteriormente, el diseño del sistema de monitoreo está basado en las características del 8031, sin embargo, el término MCS-51 y 8051 son usados indiscriminadamente para referirnos a la familia en general, cuya descripción es básicamente la misma que para el 8031. En forma

general la arquitectura en forma de bloques es posible verla en la figura III-1.

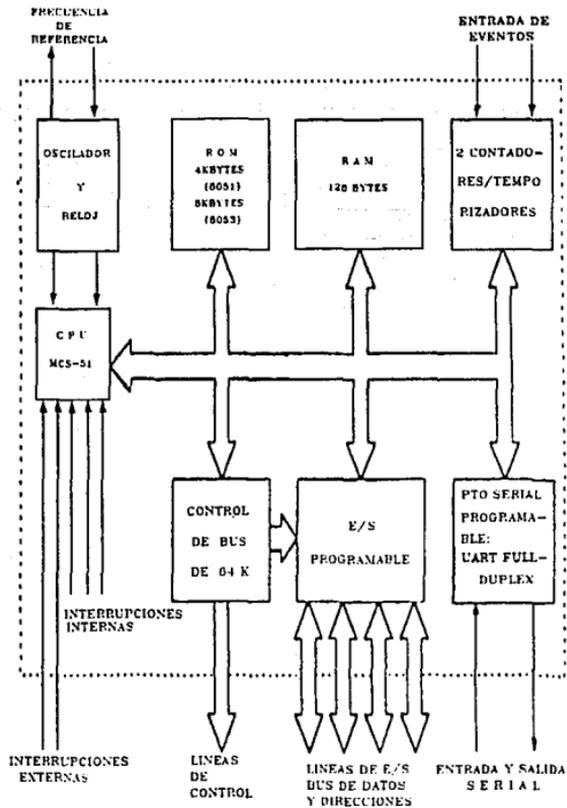


FIGURA III-1: Diagrama de bloques de la familia MCS-51 de Intel.

Se pueden observar cuatro puertos de 8 bits cada uno, por lo tanto tenemos 32 líneas de entrada/salida disponibles, con sus manejadores y registros independientes, esta es la base de comunicación de la MCU (unidad de microcontrolador) con el exterior; además contiene una RAM interna y una ROM (en algunos casos) ó algunas de sus modalidades PROM, EPROM ó EEPROM, cuyos tamaños dependen de la versión del dispositivo como se observa en la primera tabla. Vemos que los puertos tienen más de una sola función, como el puerto P0 que puede contener el byte bajo del Registro de Programa o contener las líneas de datos.

La Unidad Aritmética/lógica (ALU) maneja dos registros TMP1 y TMP2, que se comunican con acumulador (ACC) o registro principal, también existe un registro auxiliar B; el registro PSW es afectado por la ALU como resultado de las operaciones llevadas a cabo como son sobreflujos, acarreos, signos, etc.

En la parte inferior existe un circuito oscilador y dos líneas para conectar el circuito exterior de oscilación, el cual es muy sencillo y puede consistir en un cristal y dos capacitores, por lo que el circuito de sincronización, de reloj y control son internos; la señal de reloj es llevada hacia la circuitería de control que manipula señales como PSEN (habilitación de lectura de memoria externa de programas), ALE (define si el programa es externo o interno), RST (puesta de marcha), etc.

Finalmente, en la parte central existen los registros de control y datos de los contadores/temporizadores, de la comunicación serial y sus modos de operación. A manera de bloques se ha podido ver las ventajas del microcontrolador, todo ello será explicado a detalle en las secciones siguientes. Las características principales de la familia son:

- Una unidad central de procesamiento de 8 bits, para aplicaciones de control.
- Circuito de reloj contenida en el mismo dispositivo.
- 32 líneas de entrada/salida, direccionables de manera independiente.
- 64 kilobytes de espacio direccionable para memoria externa de datos.

- 64 kilobytes de espacio direccionable para memoria externa de programación.
- Dos contadores/temporizadores de 16 bits cada uno.
- Una estructura de interrupciones de 5 fuentes con 2 niveles de prioridades.
- Puerto serial "duplex", programable.
- UART interna.
- Procesador booleano, es decir, capacidad del procesador de ejecutar cálculos bit por bit o en un sólo bit lógico.
- Modalidad de espera con ahorro de energía en dispositivos CHMOS.
- RAM interna, cuyo tamaño varía según la versión, siendo lo mínimo de 128 bytes.
- Todas las versiones son compatibles pin-a-pin.

III.1.1 ORGANIZACION DE LA MEMORIA

El microcontrolador puede manejar operandos, datos y códigos de operación obtenidos de diferentes partes, por lo que existen tres espacios de memoria definidos como:

- Memoria de programas (ROM):
Puede ser interna y/o externa.
- Memoria externa de datos (RAM).
- Memoria interna de datos:
RAM de datos interna
Bloque de Registros de Funciones Especiales

El microcontrolador 8031 tiene espacios de memoria externos e internos definidos separadamente tanto para programas como para datos, como se muestra en la figura III-2, sin embargo, esto lo podemos hacer variar como se indica más adelante.

La ventaja de ésta separación es que permite que la memoria de datos sea accesada por direcciones de 8 bits, lo cual nos ayuda a una manipulación y almacenamiento más rápido utilizando una unidad de procesamiento de 8 bits, si es necesario una dirección de 16 bits se genera a través del Registro apuntador de Datos (DPTR).

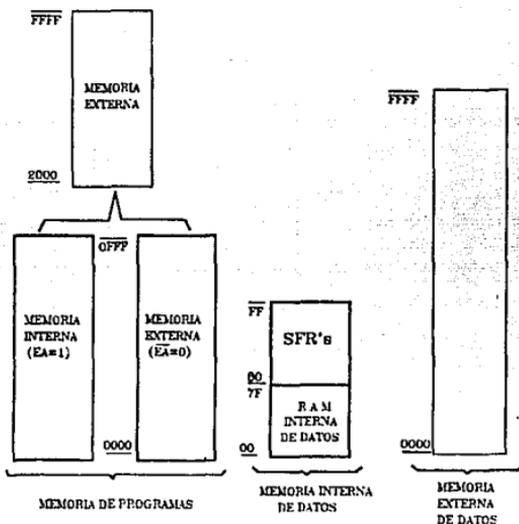


FIGURA III-2: Organización de la memoria interna del microcontrolador 8051.

Hemos visto, que la familia de microcontroladores MCS-51 tiene espacios de memoria direccionables separados, tanto para datos como para programas. Del espacio asociado a los programas tenemos que los 4 kilobytes más bajos pueden estar residentes dentro del circuito mismo, sin embargo, esto no existe dentro del 8031. Para el espacio asociado a datos, que puede consistir de 64 kilobytes de RAM exterior, además de los 128 bytes de RAM interna contenida en el dispositivo, más los registros de funciones especiales ("SFR's" por sus siglas en inglés), cuyas direcciones se muestran en la figura III-3.

En la memoria de programa está prohibido escribir, solamente es posible leer, lógicamente se encuentra en ROM o en alguna de sus variedades, y es posible tener hasta 64 kbytes ya que existen sólo 16 líneas dedicadas al direccionamiento. Es en las versiones 8051 donde los 4 kbytes más bajos se encuentran en el

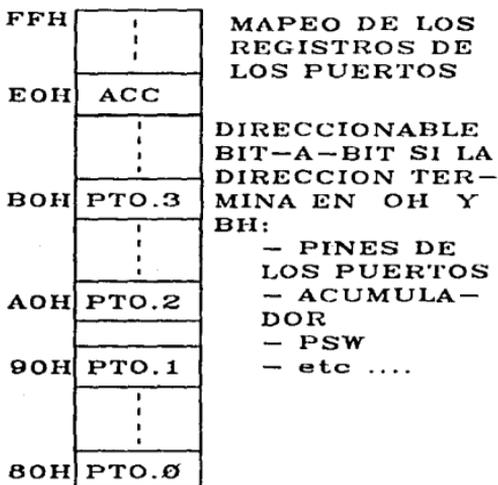


FIGURA III-3: Espacio de memoria de los registros especiales (SFR).

mismo circuito. En las versiones sin ROM interna como es el caso del 8031, el área de memoria de programa se halla fuera del circuito, que para leerla se habilita la señal PSEN (siglas de Program Store Enable).

Para la memoria de datos, que ocupa un espacio separado de la memoria de programas, se pueden direccionar también hasta 64 kbytes debido a que las líneas de direcciones son las mismas 16, el acceso a éstos datos se logra con las señales RD y WR para una lectura y una escritura, respectivamente.

Ambos espacios de memoria pueden ser combinados y formar uno solo mediante una operación lógica AND entre las señales RD y PSEN, el resultado de ésta compuerta ser una señal de habilitación de lectura del espacio de memoria externo de programa/datos.

III.1.1.1 CARACTERISTICAS DEL ESPACIO DE MEMORIA DE PROGRAMAS

El mapa de la parte más baja de la memoria de programa, separada de la memoria de datos se encuentra en la figura III-4.

Si aplicamos la señal de RST ("RESET"), que reinicializa al

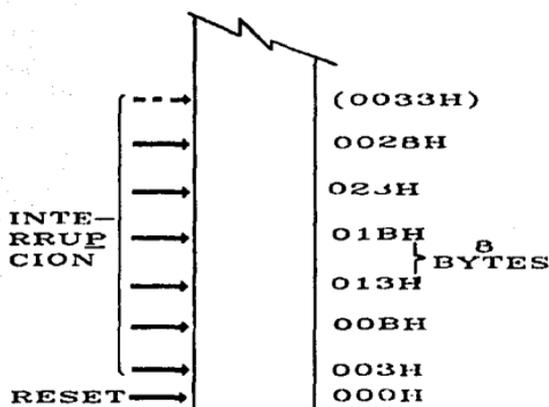


FIGURA III-4: Espacio de memoria de programas.

CPU, hace que la unidad de procesamiento empiece la ejecución desde la localidad 0000H del espacio de memoria de programa. La primera parte del área de memoria de programa se tiene apartada para la atención a las interrupciones, cada una de ellas tiene asignado ciertas localidades definidas. Si la interrupción es generada, la CPU "responde", si le es permitido, saltando a la localidad asignada e inicia la ejecución de la rutina de atención o servicio.

Si una interrupción no va a ser utilizada, su espacio correspondiente puede ser ocupado por el programa de aplicación.

De la figura anterior, se observa que la interrupción 0 está definida en la localidad 0003H, así el servicio para esta interrupción debe empezar en dicha localidad. Las localidades están espaciadas en intervalos de 8 bytes:

● Interrupción externa 0	0003H
● Timer 0	000BH
● Interrupción externa 1	0013H
● Timer 1	001BH
● etc.

Si la rutina de atención para cualquier tipo de interrupción es pequeña y cabe en los 8 bytes definidos, es posible escribirla en ese espacio, como sucede en la mayoría de rutinas de control. Pero si no es el caso, puede usarse una instrucción JUMP para saltar las localidades posteriores hasta el inicio de la rutina de atención.

Para que el microcontrolador sepa si la ROM interna o externa es la que contiene el programa de inicialización, existe la señal EA (External Access) que puede estar conectada a +5V o a 0V, en el primer caso, el programa se halla en ROM interna (EA = +5V ó Vcc), las direcciones serán desde 0000H hasta 0FFFH (4kbytes para el 8051 y derivados), para direccionar ROM externa sería desde la dirección 1000H hasta FFFFH (60 kbytes de memoria ROM externa). En la serie 8052 la memoria ROM interna ocupa desde la dirección 0000H hasta la 1FFFH (8 kbytes) y la ROM externa desde la 2000H hasta la FFFFH (56 kbytes). En el segundo caso EA = 0V o Vss, indica que todo el área de memoria de programa es externo, y es usado siempre en dispositivos 8031 y derivados como el 8032.

La señal $\overline{\text{PSEN}}$ es activada para acceder memoria externa de programa, por lo tanto no se activa para el acceso de memoria interna.

III.1.1.2 HARDWARE MÍNIMO PARA LA EJECUCIÓN DE UN PROGRAMA

La circuitería mínima para ejecutar un programa desde memoria externa es el mostrado en la figura III-5.

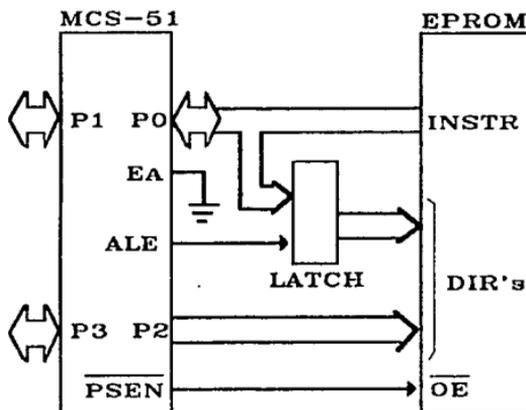


FIGURA III-5: Sistema básico del microcontrolador con memoria externa de programas (EPROM).

Observamos que las 16 líneas que forman a los puertos P0 y P2 son dedicados a direcciones, es decir, contienen el bus de direcciones, el puerto P0 multiplexa direcciones y datos, proporcionando la parte baja de una dirección y todo el bus de datos. Así el puerto P0 propaga el valor del byte bajo del contador de programa (PCL) como parte de la dirección de memoria ROM externa, se activa la señal ALE y se almacena en el latch, después de un ciclo de espera para obtener la respuesta de la memoria, y entonces la parte baja del bus de direcciones se convierte en el bus de datos, así el byte del código de operación es leído por el microcontrolador.

III.1.1.3 CARACTERISTICAS DEL ESPACIO DE MEMORIA EXTERNA DE DATOS

Existe una memoria de datos direccionable de un tamaño máximo de 64 kbytes para el usuario del microcontrolador. Una dirección del espacio de memoria externa de datos puede ser de 1 ó 2 bytes. Las direcciones de un sólo byte son usados junto con más líneas del puerto P2 para paginar la RAM, cuando se usan 2 bytes, el más bajo lo proporciona el puerto P0 y la otra parte se adquiere a través del puerto P2.

III.1.1.4 HARDWARE MINIMO PARA EL USO DE RAM EXTERNA

Existe, en todos los dispositivos MCS-51 una RAM interna, pero además es posible ampliarla mediante una RAM externa al servicio del usuario, dentro de lo que hemos llamado *espacio de memoria externa de datos*. La configuración mínima para el acceso a RAM usando un microcontrolador de la familia MCS-51 es la que se puede observar en la figura III-6.

El programa de inicialización se encuentra en una ROM interna del microcontrolador, el puerto P0 contiene de manera multiplexada la parte baja de la dirección a acceder y los datos; por ello un latch es necesario para permitir la diferenciación de datos y direcciones, mediante la señal \overline{ALE} es habilitado y se almacena la dirección dentro del latch, hasta que se tienen los datos en las líneas correspondientes. El puerto P2 permite paginar la RAM, ya que nos proporciona la parte alta de la dirección a acceder; el microcontrolador envía dos señales más: RD y WR que permiten una lectura o una escritura en la RAM. Solamente el puerto P1 es totalmente dedicado a propósitos generales.

III.1.2 CARACTERISTICAS DEL ESPACIO DE MEMORIA INTERNO DE DATOS

El microcontrolador 8031 contiene 128 bytes de RAM interna y 20 registros de funciones especiales, luego entonces se tiene una memoria interna de datos dividida en dos secciones:

RAM interna de datos:

De la dirección 00H a la 7FH

Bloque de Registros de Funciones Especiales:

De la dirección 80H a la FFH

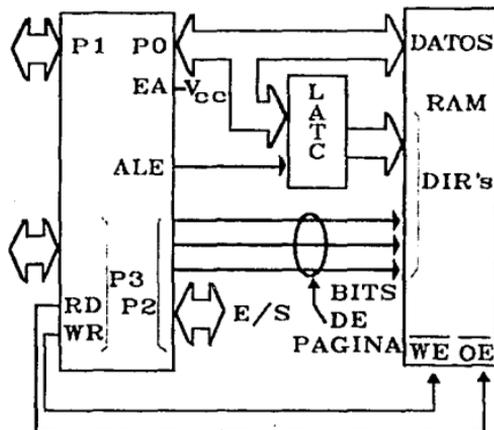


FIGURA III-6: Sistema básico del microcontrolador con memoria externa de datos (RAM).

Por lo tanto la profundidad de la pila está dada por la RAM interna de datos con 128 localidades disponibles. Un mapa general del rea de memoria interna de datos es el mostrado en la figura III-7.

El espacio se encuentra dividido en tres bloques, que son referidos como los 128 más bajos, los 128 más altos y el espacio SFR (Registros de Funciones Especiales). Los bytes más altos y los SFR's ocupan el mismo bloque desde la dirección 80H hasta la FFH con características físicas que los separan. Las direcciones de la memoria interna de datos son de un sólo byte, por lo que podemos direccionar hasta 256 localidades, pero con las diferentes modalidades de direccionamiento es posible acceder

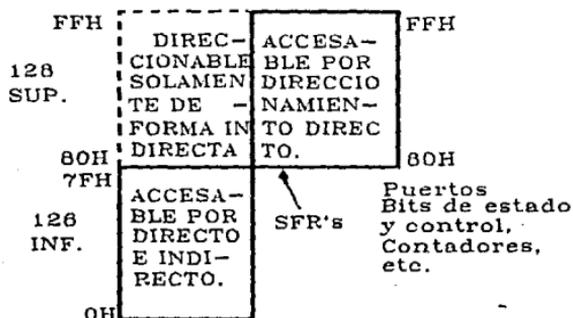


FIGURA III-7: Espacio de Memoria interna de datos.

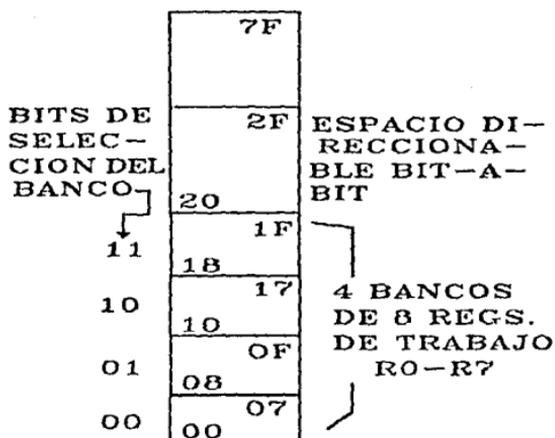


FIGURA III-8: Organización de la RAM interna del Microcontrolador.

hasta 384 bytes. El direccionamiento directo abajo de 7FH sólo es un espacio de memoria libre con algunos registros R0,...,R7; mientras que el espacio superior es diferente y se accesa con direcciones mayores a 7FH, como se observa en la figura III-8.

Los primeros 32 bytes de la parte baja de memoria RAM interna es agrupada en 4 bancos de 8 registros, desde la dirección 00H hasta 1FH. Nos referiremos a esos registros llamándolos R0, R1, R2, R3, R4, R5, R6 y R7; para saber cual de los cuatro bancos está activo se utilizan dos bits de la Palabra de Estado del Programa (PSW). La agrupación por bancos nos permite una mayor eficiencia en el uso de espacio de código, lográndose unas instrucciones de registro más cortas que las instrucciones que utilizan el direccionamiento directo. Los siguientes 16 bytes al banco de registros forman un bloque de espacio direccionable bit-a-bit. Dentro del set de instrucciones se incluye una sección de operaciones con un sólo bit. El área referida se encuentra entre las direcciones 00H y 7FH, que lo podemos ver en la figura III-9.

Los primeros 128 bytes de RAM pueden ser accedados por ambos modos: directo e indirecto; más allá de los 128 bytes solamente es posible utilizar el modo indirecto, en las versiones

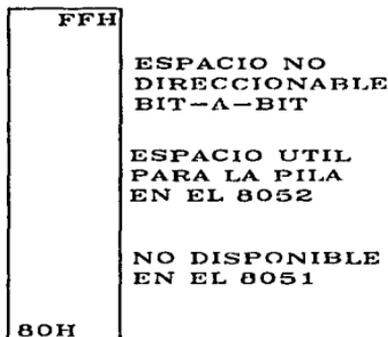


FIGURA III-9: Espacio de memoria interna y los tipos de direccionamientos válidos en ella.

que contienen más memoria RAM interna (8052, 8351, etc), también existe el espacio de los Registros de Funciones Especiales, que incluye los latches de los puertos, de los temporizadores, etc., donde únicamente podemos utilizar direccionamiento directo, con ambos modos: byte o bit-a-bit.

III.1.2 REGISTROS DE FUNCIONES ESPECIALES

Existe un área de memoria interna de datos dedicada al almacenamiento de datos de los registros internos del procesador, es llamado espacio de los registros de funciones especiales (SFR), cuyo mapa de memoria dentro del microcontrolador es el mostrado en la figura III-10

Es de notarse que las localidades ocupadas no son continuas, aquellas que no son usadas es porque no están implementadas en el circuito. Una lectura a dichas direcciones nos dará datos aleatorios y una escritura no tendrá efecto. Mediante la programación es posible escribir en ellas, sin embargo, son localidades reservadas a posibles expansiones. Después de una reinicialización (RST) es posible tener 0's en esas localidades.

Todos los registros, con excepción de los registros R0, R1, R2, R3, R4, R5, R6 y R7, se encuentran en el espacio de los SFR's, con direcciones desde la 80H hasta la FFH, como los registros aritméticos, apuntadores, puertos de Entrada/Salida, registros del sistema de interrupciones, contadores/temporizadores y el área temporal del puerto serial. Las funciones de los "SFRs" son descritas a continuación:

ACUMULADOR:

El registro de Acumulador se designa como ACC, sin embargo es simplemente referido como A, en general. Su dirección física es E0H.

REGISTRO B:

Es usado para las operaciones de multiplicación y división o como registro temporal de datos; también puede ser usado como "colchón" en otras instrucciones (registro de resguardo). Su dirección física es F0H.

8 BYTES

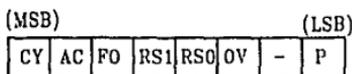
F8								FF
F0	B							F7
E8								EF
E0	ACC							E7
D8								DF
D0	PSW							D7
C8	(T2CON)	(RCAP2L)	(RCAP2H)	(TL2)	(TH2)			CF
C0								C7
B8	IP							BF
B0	P3							B7
A8	IE							AF
A0	P2							A7
98	SCON	SBUF						9F
90	P1							97
88	TCON	TMOD	TLO	TL1	TH0	TH1		8F
80	PO	SP	DPL	DPH			PCON	87

() Regs. que solo existen en el 8052.

FIGURA III-10: Mapa de memoria interna de los registros de funciones especiales (SFR').

PALABRA DE ESTADO DE PROGRAMA:

Referido como PSW (por sus siglas en inglés) contiene la información sobre el estado de procesador, organizado como se ve en la figura III-11. Su dirección física es D0H.



SIMBOLO	POSICION	NOMBRE	SIMBOLO	POSICION	NOMBRE Y FUNCION
CY	PSW.7	BIT DE ACARREO	OV	PSW.2	BIT DE SOBRE-FLUJO
AC	PSW.6	BIT AUXILIAR DE ACARREO	-	PSW.1	BIT DEFINIDA POR EL USUARIO
FO	PSW.5	BANDERA 0 DE PROPOSITO GRAL	P	PSW.0	BIT DE PARIDAD
RS1	PSW.4	BIT 1 Y BIT 0 PA-			
RS1	PSW.3	RA SELECCIONAR EL BANCO ACTIVO DE REGISTROS DE TRABAJO.			
RS1	RS0	BANCO ACTIVO	DIRECCIONES		
0	0	0	00H-07H		
0	1	1	08H-0FH		
1	0	2	10H-17H		
1	1	3	18H-1FH		

FIGURA III-11: Formato de la palabra de estado del microcontrolador.

PUNTADOR DE LA PILA:

Llamado SP (por sus siglas en inglés) es un registro de 8 bits, que es incrementado antes de almacenar un dato durante la operación de agregar a la pila y en las llamadas a subrutinas. Mientras que la pila puede residir en cualquier parte de la RAM interna, el apuntador es inicializado a 07H después de aplicar un "reset". Lo que causa que la pila se inicialice en la localidad 08H. Su dirección física es 81H.

APUNTADOR DE DATOS:

Designado como DPTR (por sus siglas en inglés) contiene en dos bytes: un byte bajo (DPL, por sus siglas en inglés) y un byte alto (DPH, por sus siglas en inglés), debido a que contendrá una dirección de 16 bits para manejar datos de 8 bits, además puede ser manipulado como dos registros independientes de 8 bits.

PUERTOS 0, 1, 2 Y 3:

Los puertos son los medios de comunicación del circuito con el mundo exterior. Contiene manejadores y registros que son

referidos como P0, P1, P2 y P3, respectivamente. Y sus direcciones físicas son 80H, 90H, A0H y B0H.

AREA DE ALMACENAMIENTO TEMPORAL DE DATOS SERIALES:

Llamado SBUF (del término en inglés "Serial Data Buffer") contiene dos registros separados: el receptor y el transmisor. Cuando el dato es llevado al SBUF, puede ser transmitido mediante comunicación serial; cuando un dato es traído desde el SBUF, el dato puede ser recibido desde el área de almacenamiento. Su dirección física es 99H.

REGISTROS TEMPORIZADORES:

Son registros pares de 16 bits que contienen los registros de los contadores/temporizadores 0, 1 y 2, respectivamente designados como TH0, TL0, TH1, TL1 y TH2, TL2. Con las direcciones 8CH, 8AH, 8DH, 8BH, CDH y CCH.

REGISTROS DE CAPTURA:

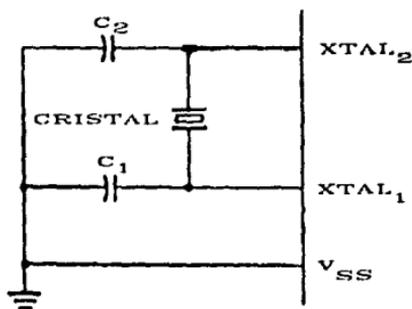
Es un registro par, referido como RCAP2H y RCAP2L, para el temporizador 2 en modo de captura. Con un modo de autocarga. Su dirección física es DH y CAH.

REGISTROS DE CONTROL:

Los registros de funciones especiales son IP, IE, TMOD, TCON, T2CON, SCON y PCON que contienen los bits de estado y control para el sistema de interrupciones, los contadores y el puerto serie. Más adelante serán descritos a detalle. Sus direcciones físicas, en el orden mencionado son: B8H, A8H, 89H y 88H.

III.1.3 CIRCUITO OSCILADOR

Una de las ventajas del microcontrolador 8031 es la de contener, dentro del mismo dispositivo su circuito de reloj, contiene dos pines llamados XTAL1 y XTAL2, de entrada y salida, que puede ser configurado con pocos componentes exteriores, así utilizando un oscilador y dos capacitores es posible hacer un circuito oscilador propio del microcontrolador, físicamente conectados como se puede ver en la figura III-12.



$C_1, C_2 = 30\text{pF} \pm 10\text{pF}$ (Cristal)
 $= 40\text{pF} \pm 10\text{pF}$ (resonantes de cerámica)

FIGURA III-12: Configuración externa para el circuito oscilador del microcontrolador.

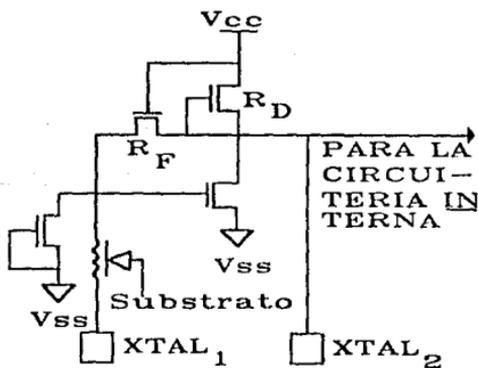


FIGURA III-13: Amplificador interno del circuito de oscilación del microcontrolador.

referidos como P0, P1, P2 y P3, respectivamente. Y sus direcciones físicas son 80H, 90H, A0H y B0H.

AREA DE ALMACENAMIENTO TEMPORAL DE DATOS SERIALES:

Llamado SBUF (del término en inglés "Serial Data Buffer") contiene dos registros separados: el receptor y el transmisor. Cuando el dato es llevado al SBUF, puede ser transmitido mediante comunicación serial; cuando un dato es traído desde el SBUF, el dato puede ser recibido desde el área de almacenamiento. Su dirección física es 99H.

REGISTROS TEMPORIZADORES:

Son registros pares de 16 bits que contienen los registros de los contadores/temporizadores 0, 1 y 2, respectivamente designados como TH0, TL0, TH1, TL1 y TH2, TL2. Con las direcciones 8CH, 8AH, 8DH, 8BH, CDH y CCH.

REGISTROS DE CAPTURA:

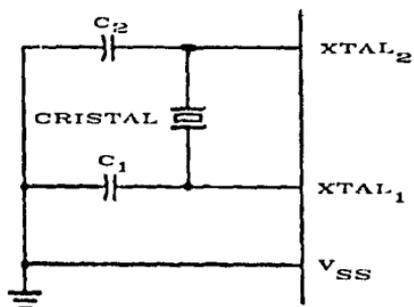
Es un registro par, referido como RCAP2H y RCAP2L, para el temporizador 2 en modo de captura. Con un modo de autocarga. Su dirección física es DH y CAH.

REGISTROS DE CONTROL:

Los registros de funciones especiales son IP, IE, TMOD, TCON, T2CON, SCON y PCON que contienen los bits de estado y control para el sistema de interrupciones, los contadores y el puerto serial. Más adelante serán descritos a detalle. Sus direcciones físicas, en el orden mencionado son: B8H, A8H, 89H y 88H.

III.1.3 CIRCUITO OSCILADOR

Una de las ventajas del microcontrolador 8031 es la de contener, dentro del mismo dispositivo su circuito de reloj, contiene dos pines llamados XTAL1 y XTAL2, de entrada y salida, que puede ser configurado con pocos componentes exteriores, así utilizando un oscilador y dos capacitores es posible hacer un circuito oscilador propio del microcontrolador, físicamente conectados como se puede ver en la figura III-12.



$C_1, C_2 = 30\text{pF} \pm 10\text{pF}$ (Cristal)
 $= 40\text{pF} \pm 10\text{pF}$ (resonantes de cerámica)

FIGURA III-12: Configuración externa para el circuito oscilador del microcontrolador.

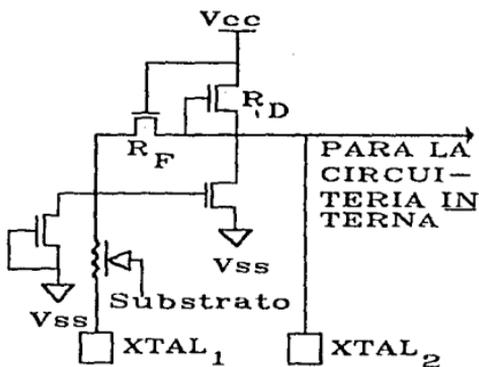


FIGURA III-13: Amplificador interno del circuito de oscilación del microcontrolador.

El oscilador maneja al generador interno de reloj, que provee al procesador, de una frecuencia media con respecto al oscilador y nos define las fases internas, los estados y los ciclos de máquina. Los pines de XTAL1 y XTAL2 se comunican a un amplificador-inversor con el exterior, que se deberá configurar para controlar el circuito oscilador interno, como se muestra en la figura III-13.

III.1.4 CICLOS DE LA UNIDAD CENTRAL DE PROCESAMIENTO

Un ciclo de máquina consiste en 6 estados, es decir, en 12 períodos de oscilación. Cada estado está dividido en dos fases ambas con señal de reloj activa. Los seis estados del ciclo de máquina se enumeran como S1P1 (estado 1, fase 1), S1P2, ... S6P2. Cada fase tiene una duración de un período de oscilación y cada estado de dos períodos. Entonces, por ejemplo, si utilizamos un cristal de 12 Mhz un ciclo de máquina con 12 oscilaciones tardaría un microsegundo en efectuarse.

Normalmente, una operación lógica y aritmética es ejecutada durante la fase uno y la transferencia de registro-a-registro durante la fase dos.

La señal $\overline{\text{ALE}}$ (por siglas en inglés "Address Latch Enable") es activada dos veces durante cada ciclo de máquina: la primera durante S1P2 y S2P1 y nuevamente durante S4P2 y S5P1.

La ejecución de una instrucción es llevada a cabo durante el S1P2:

- El código de operación es almacenado en el registro de Instrucción (IR)
- Si es una instrucción de dos bytes, el segundo byte es leído durante el cuarto estado del mismo ciclo de máquina.
- Si es una instrucción de un sólo byte, se hace un ciclo "fetch" en el estado cuatro, pero el byte leído es ignorado (porque podría ser el código de operación de la siguiente instrucción), y el PC (contador de programa) no es incrementado.
- La ejecución es terminada en la S6P5.

- La mayor parte de las instrucciones son ejecutadas en un sólo ciclo de máquina, solamente las instrucciones MUL (multiplicación) y DIV (división) necesitan de más de dos ciclos de reloj completos, es decir, requieren de cuatro ciclos de máquina.

Casi siempre, son extraídos dos bytes de código desde la memoria de programa, durante cada ciclo. La única excepción es cuando la instrucción MOVX es ejecutada, ya que es de un byte y dos ciclos para acceder la memoria externa de datos.

Las secuencias de traer el código de operación y ejecutar una instrucción son las mismas, sin importar el tipo de memoria de programa y el tiempo de ejecución tampoco depende de ello. Si ocurre un acceso a memoria para leer un dato, PSEN se habilita 2 veces para permitir demultiplexar al bus de datos/direcciones. Cuando el microcontrolador ocupa memoria interna de programa, PSEN no se activa jamás, y el bus de direcciones se halla siempre disponible para otros propósitos, aunque ALE si es activada dos veces por cada ciclo de máquina y puede servir como una salida de reloj, pero varía en instrucciones que hacen accesos a memoria RAM externa.

III.1.5 ESTRUCTURA Y OPERACION DE LOS PUERTOS

El microcontrolador tiene cuatro puertos bidireccionales, que consisten en un "latch", referidos como registros de funciones especiales de P0 hasta P3, de un manejador de salida "driver" y de un área temporal de datos de entrada "buffer".

Además, los puertos tienen funciones alternas, los puertos P0 y P2 contienen los bytes bajos y altos del bus de direcciones de la memoria externa a acceder; el puerto P0 es multiplexado al tiempo de hacer una escritura o una lectura, si la dirección no es de 16 bits, el puerto P2 mantiene la información del contenido del SFR. El P0 también contiene las líneas de datos. Todo el puerto P3 es multifunciones, como se describe a continuación:

PIN	Función Alternas
P3.0	RXD (Receptor del Puerto Serie)
P3.1	TXD (Transmisor del Puerto Serie)
P3.2	INT0 (Interrupción externa)

- P3.3 INT1 (Interrupción externa)
- P3.4 T0 (entrada externa del contador/temporizador1)
- P3.5 T1 (entrada externa del contador/temporizador2)
- P3.6 WR (Habilitador de escritura de memoria de datos externa)
- P3.7 RD (Habilitador de lectura de memoria externa de datos)

Las funciones alternas de los puertos pueden estar activadas, solamente si el bit correspondientes al latch del SFR del puerto contiene un 1.

III.1.5.1 CONFIGURACION PARA ENTRADA/SALIDA DE LOS PUERTOS

Los puertos contienen "latches" para su comunicación con el exterior, cada uno con una estructura electrónica diferente. Uno de los bits del SFR está representado como un flip-flop tipo D, que muestra el valor interno del bus, como respuesta a la señal "escribe al latch" proveniente del procesador. El valor Q, que es la salida del flip-flop, es llevado al bus interno, también como respuesta pero a una lectura.

Existen algunas diferencias entre las configuraciones físicas de los puertos, los manejadores de los puertos P0 y P2 son elegidos en base a una dirección ADDR y a unas líneas del ADDR/DATA mediante señales de control para usarse en memoria externa. Durante un acceso de memoria externa, el SFR del Puerto P2 permanece inalterado, sin embargo, el SFR del puerto P0 contiene 1's. Mientras que el puerto P3 funciona de otra manera, si el bit del latch contiene un 1, el nivel de salida es controlada por una señal referida como "función alterna de salida". Por lo que cualquier pin del puerto P3, está disponible para una función alterna de entrada.

Los tres puertos anteriormente mencionados, tienen resistencias internas para levantar el voltaje ("pull-ups"). El puerto P0 tiene salidas de colector abierto. Cada línea de entrada y/o salida puede ser utilizada de manera independiente, ya sea para entrada o salida de datos. Sin embargo, es importante recordar que los puertos P0 y P2 no pueden ser siempre usados

para cualquier objetivo ya que contienen al bus de datos y de direcciones. También hay que notar que el bit del latch debe contener un 1, para poder usar el puerto como entrada, ya que de esta manera es apagado el FET manejador de salida.

En el puerto P0 no existen las resistencias internas de activación. El FET de activación en el manejador de salidas del puerto P0 es utilizado solo cuando el puerto esta mandando 1's durante los accesos de memoria externa. De otro manera el FET se halla apagado. Por lo tanto, las líneas del puerto P0 que son para bits de salida son de drenaje abierto. Si mandamos un 1 al bit del latch, se hace que ambas salidas de los Fets estén apagadas, como si el pin estuviera sin conectarse, lo que nos permite tener una condición de alta impedancia a la entrada.

Se puede decir que los puertos P1, P2 y P3 son "pseudo-bidireccionales", y el puerto P0 realmente es bidireccional, porque no tiene resistencia de activación interna. Si mandamos una señal de "RESET", y se escriben 0's al latch del puerto, cualquiera que sea, es posible reconfigurar el puerto si mandamos 1's a dicho latch.

III.1.5.2 PROCESO DE ESCRITURA EN UN PUERTO

Para poder escribir un dato en el latch de uno de los puertos, el dato es mandado durante el S6 (sexto estado) del ciclo de máquina, al final de la instrucción. Los niveles de entrada y salida de los manejadores de puertos son TTL 1S: es posible obtener salidas tipo colector abierto en versiones HMOS y CHMOS.

Existen instrucciones para leer el latch del puerto y otras para leer el pin del puerto. Las instrucciones que leen al latch son instrucciones de solo lectura de un valor, que lo pueden activar y reescribir en el latch, se les llama *instrucciones de lectura-modificación*. Por ejemplo, si tenemos conectado un pin de un puerto a la base de un transistor. Cuando se escribe un 1, el bit enciende al transistor, si la CPU lee ese mismo pin, lee un 0 que es el voltaje de la base del transistor, pero si se lee el latch se lee un 1.

Cuando el operando destino es un puerto o uno de sus bits, las instrucciones leen el latch en vez del pin, como en las siguientes instrucciones:

*ANDL, ORL, XRL, JBC, CPL, INC, DEC,
DJNZ, MOV PX.Y, CLR PX.Y, SETB PX.Y,*

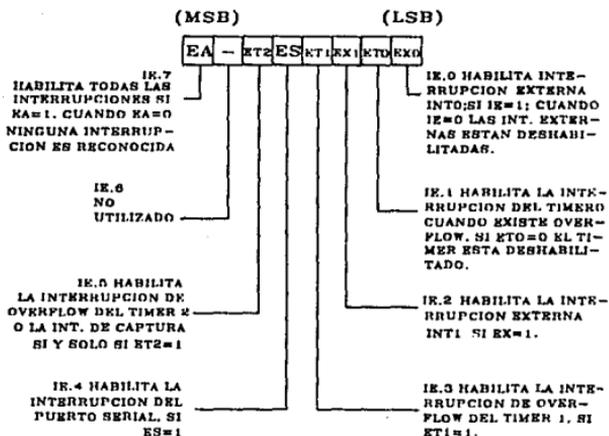


FIGURA III-14: Formato del registro IE (habilitación de interrupciones).

III.1.6 SISTEMA DE INTERRUPTACIONES

La familia de los microcontroladores MCS-51 tiene cinco fuentes que pueden generar interrupciones:

- Interrupciones externas (INT0 e INT1)
- Interrupciones de los temporizadores (T0 y T1)
- Interrupción del puerto serial

Cada una de las interrupciones puede o no estar habilitada, es decir, existe el registro IE (de las siglas Interrupt Enable) que permite que una interrupción pueden hacerse ó no, también contiene un bit que deshabilita todas las interrupciones, el formato de IE es el mostrado en la figura III-14.

También se pueden clasificar en niveles de interrupción asignándoles prioridades mediante el registro IP (Interrupt Priority), así cada fuente de interrupción puede ser programada en uno de los dos niveles al activar o no el bit correspondiente. El formato de IP se puede ver en la figura III-15.

Una prioridad baja puede ser interrumpida por una alta prioridad, pero jamás de modo inverso, ni por otra prioridad más baja. Además, una prioridad alta no puede ser interrumpida.

Dentro del arbitreo de dos prioridades de diferente nivel que requieren ser atendidas al mismo tiempo se tiene que el procesador atiende a la de mayor prioridad. Pero si tienen la misma prioridad, mediante un sistema de poleo interno se decide

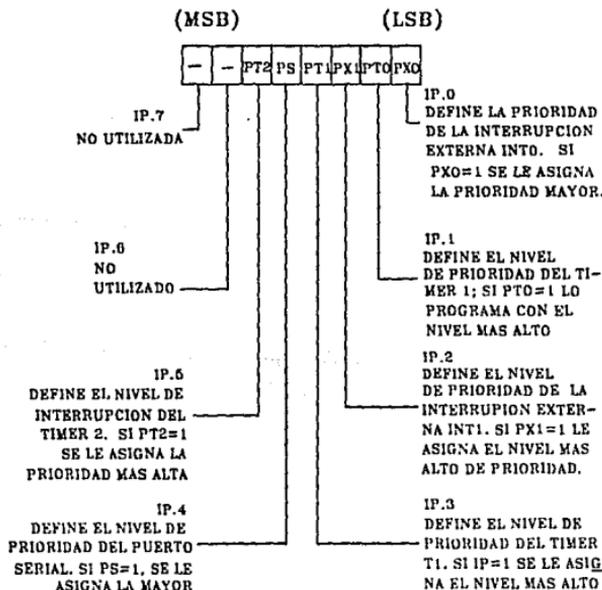


FIGURA III-15: Formato del registro de prioridades de las interrupciones (IP).

a cual atender primero. Por lo tanto para cada nivel de prioridades existe una segunda estructura determinado por la secuencia de poleo siguiente:

<i>Fuente</i>	<i>Prioridad</i>
1. IE0	<i>Muy alta</i>
2. TFO	...
3. IE1	...
4. TFI	...
5. RI + TI	<i>Muy baja</i>

El modo de operación del sistema de control de interrupciones es:

- Todas las banderas de interrupciones son guardadas en el sistema de control de interrupciones en el S5 del ciclo de máquina.
- Durante el siguiente ciclo de máquina se polea todas las entradas.
- Si una de las banderas está activa (1) se genera la interrupción y una instrucción LCALL es ejecutada hacia la rutina de atención.
- Al terminar la rutina de atención se reestablece PC con su valor inicial.

Al hacerse un LCALL, se introduce la dirección actual del registro PC dentro de la pila de datos y una nueva dirección le es asignada, la de la rutina de atención, definidas en la parte baja de la memoria de programas.

Cuando se introduce el registro PC en la pila de datos, se hace automáticamente, cosa que no se puede hacer con ningún otro registro; esto permite decidir cuanto tiempo se tomará salvar los otros registros, lo que mejora la respuesta real de una interrupción, dejándolo como responsabilidad del programador cualquier problema de espera infinita. El valor introducido en PC es llamado vector de interrupciones y son:

<i>Fuente</i>	<i>Vector</i>
IE0	0003H
TFO	00BH
IE1	0013H
TFI	001BH
RI + TI	0023H

El vector es la nueva dirección del registro PC, después de ejecutarse la instrucción LCALL y mediante RETI se recarga el valor de la instrucción donde fue hecha la interrupción.

En consecuencia, tenemos muchas funciones de interrupción que son típicas en aplicaciones de control, como recargar un temporizador o descargar el área de datos del puerto serial, que nos ahorra tiempo con respecto a otras arquitecturas.

Para casos donde se necesite un nivel más de prioridades, mediante el auxilio de la programación se puede simular, creando el mismo efecto, como tener un tercer nivel de prioridades. Por ejemplo podemos simularlo así:

```

PUSH IE
MOV IE, #MASK
CALL ETIQ
....
....
rutina de atención
....
....
POP IE
RET
ETIQ: RETI

```

- 1o. Las interrupciones con niveles mayor que uno se les asigna el nivel 1 mediante el registro IP.
 - 2o. Las rutinas de interrupción con prioridad 1 interrumpen.
 - 3o. Se redefine el registro IE, después de deshabilitar las interrupciones de prioridad 2.
 - 4o. Se ejecuta el CALL ETIQ y la instrucción RETI.
 - 5o. Entonces cualquier interrupción de prioridad 1 está deshabilitada, pero una mayor sí podrá interrumpir.
 - 6o. Se atiende en la rutina de servicio.
 - 7o. La instrucción POP reestablece las condiciones iniciales.
- Si tenemos un cristal de 12 Mhz, esta parte de programación requiere 10 microsegundos para atender a una interrupción.

Las interrupciones externas son las generadas a través de los pines INT0 e INT1 y pueden ser: activadas por flanco o activadas por nivel, dependiendo de los bits IT0 e IT1 del registro TCON. Si $ITx = 0$, la interrupción es dada por una señal baja en el pin INTx. Si $ITx = 1$ la interrupción se activa por flanco.

Las banderas de interrupción para esas fuentes de interrupciones son IE0 e IE1 del registro TCON. La bandera IEx es activa si en un ciclo hay un voltaje alto y al siguiente ciclo un bajo.

Para los temporizadores T0 y T1, los bits son llamados TF0 y TF1, que serán programados a través de sus registros respectivos.

En el puerto serial, una interrupción se genera con una OR lógica entre R1 y T1.

Una interrupción puede ser ignorada o bloqueada por alguna de las siguientes condiciones:

- 1.- Una interrupción de igual o mayor prioridad que está siendo atendida.
- 2.- El actual ciclo de poleo no está en la etapa final de la instrucción en ejecución.
- 3.- La instrucción que está ejecutándose es RETI u otra que haga alguna modificación sobre los registros IE ó IP.

El poleo se lleva a cabo en cada ciclo de máquina, y el resultado se presenta después de cada ciclo de instrucción. Las banderas son deshabilitadas, excepto las del puerto serial y del temporizador dos (para el 8052), donde el usuario tendrá que hacerlo por programación. Se tiene que limpiar las banderas si se usa activación por flanco.

El tiempo de respuesta a una interrupción es siempre mayor a 3 ciclos de máquina y menor a 9 ciclos.

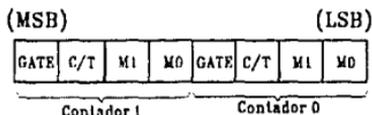
III.1.7 FUNCIONAMIENTO Y CARACTERÍSTICAS DE LOS CONTADORES/TEMPORIZADORES

Se tienen dos contadores/temporizadores programables dentro de un dispositivo 8051 llamados T0 y T1, son utilizados como temporizadores o como contadores de eventos.

Cuando funciona como temporizador, su registro se incrementa cada ciclo de máquina, por lo tanto su rango es 1/12 de la frecuencia de oscilación.

Cuando se usa como contador, el registro se incrementa si sucede un cambio de nivel 1 a 0 en el pin externo T0 o T1, lo que toma dos ciclos de máquina para ser reconocido y actualizar el registro. No existen restricciones en la señal de entrada, pero debe asegurarse una duración de al menos un ciclo de reloj.

Para programar a los contadores/temporizadores se utiliza el registro TMOD, con el formato de la figura III-16.



<p>GATE: Control de compuerta. Cuando el Contador o temporizador esta habilitado solamente mientras el pin INTx esta en nivel alto y el pin de control TRx=1. Cuando el Timer x esta borrado y habilitado cualquier TRx=1</p> <p>C/T: Selector y borrado del contador/temporizador Para el modo de temporizador la entrada en el reloj del sistema. Para el modo de contador, la entrada es el pin Tx.</p>	<table border="0"> <tr> <td style="text-align: center;">M1</td> <td style="text-align: center;">M0</td> <td style="text-align: center;">MODO DE OPERACION</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Timer TLx sirve como preescala de 5 bits</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>El valor THx de 8 bits es auto-recargado en TLx cada vez que ocurre un overflow.</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>TLO es un contador/temporizador de 8 bits controlado por la palabra de control del Timer 0. TH0 es controlado por el Timer 1</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>El Contador/Temporizador 1 se encuentra en alto total.</td> </tr> </table>	M1	M0	MODO DE OPERACION	0	0	Timer TLx sirve como preescala de 5 bits	0	1	El valor THx de 8 bits es auto-recargado en TLx cada vez que ocurre un overflow.	1	1	TLO es un contador/temporizador de 8 bits controlado por la palabra de control del Timer 0. TH0 es controlado por el Timer 1	1	1	El Contador/Temporizador 1 se encuentra en alto total.
M1	M0	MODO DE OPERACION														
0	0	Timer TLx sirve como preescala de 5 bits														
0	1	El valor THx de 8 bits es auto-recargado en TLx cada vez que ocurre un overflow.														
1	1	TLO es un contador/temporizador de 8 bits controlado por la palabra de control del Timer 0. TH0 es controlado por el Timer 1														
1	1	El Contador/Temporizador 1 se encuentra en alto total.														

FIGURA III-16: Formato del registro TMOD.

Los bits M1 y M0 permiten seleccionar uno de los cuatro modos disponibles, donde el modo 0, 1 y 2 son iguales en ambos contadores/temporizadores y el modo 3 es diferente.

III.1.7.1 MODOS DE OPERACION DE LOS CONTADORES/TEMPORIZADORES

MODO 0:

El contador/temporizador funciona como un contador de 8 bits dividido por una preescala de 32. El registro del contador está configurado como una localidad de 13 bits, formado de 8 bits de TH1 más los 5 bits más bajos de TL1 (los otros tres están indeterminados y deberán ser ignorados); el contador va de todos 1's a todos 0's y activa la bandera del registro TF1, lográndose un evento.

Ambos contadores/temporizadores pueden ser programados en modo 0.

MODO 1:

Básicamente es lo mismo que el modo 0, aunque el temporizador cuenta con los 16 bits completos.

MODO 2:

Consiste en la programación del temporizador como un contador (TL1) de 8 bits con autocarga. En el registro de control TCON, el octavo bit indica un sobreflujo (OV) detectado en el registro del contador TL1, lo que hace que se cargue el contenido de TH1 en TL1, donde TH1 fue inicializado por programación. Esta operación no afecta a TH1.

El modo 2 también puede usarse para programar al temporizador 0.

MODO 3:

La programación en modo 3 es diferente para el contador 0 y para el contador 1.

CONTADOR 0 EN MODO 3:

Los registros TL0 y TH0 funcionan como dos contadores diferentes. Donde el contador TL0 usa los bits de control del temporizador 0: C/T, GATE, TR0, INT0 y TF0. Y el contador TL1

es usado como contador de ciclos de máquina y toma a los bits TF1 y TR1 del temporizador 1 para control, por lo que el contador TH0 controla la interrupción del temporizador 1.

CONTADOR 1 EN MODO 3:

Simplemente el contador 1 almacena su cuenta, es parecido a poner el bit de control TR1 = 0. Las ventajas del modo 3 son que nos permite tener un contador mas; dentro del dispositivo en total serían 3 contadores disponibles; y sería por medio de la programación, activar y desactivar al contador 1 cuando el contador 0 se encuentra en modo 3.

III.1.8 EL PUERTO SERIAL DE COMUNICACION

El puerto serial es de tipo dúplex, lo que nos permite transmitir y recibir de manera simultánea, con un área de almacenamiento temporal del receptor, lo que implica que puede empezarse a recibir un segundo byte antes de que el byte anterior haya sido colocado en el registro del receptor. Sin embargo, puede perderse un byte si el primero todavía no ha sido leído y el segundo ya ha sido recibido. El área de almacenamiento temporal es llamado SBUF (siglas del término en inglés de "Serial Buffer"), cuando escribimos al registro SBUF significa transmitir y leer de SBUF es recibir.

El puerto serial es programable en 4 modos diferentes.

MODO 0:

La transmisión y la recepción se hace a través del pin RXD. Se reciben/transmiten 8 bits cada vez, primeramente el LSB (bit menos significativo), el baudaje es igual a un doceavo de la frecuencia de oscilación. El pin TXD tiene un reloj de corrimiento. La recepción se inicia si R1 = 0 y REN = 1. Para los otros modos basta con REN = 1 (figura III-17).

MODO 1:

El conjunto de datos es de 10 bits, compuesto de:

- *Un bit de inicio (= 0)*
- *8 bits de datos (de D0 a D7, primeramente el LSB)*
- *Y un bit de paro (= 1)*

La transmisión se hace a través de TXD y la recepción en RXD. El baudaje es variable y el bit de paro es asignado al bit RB8 del

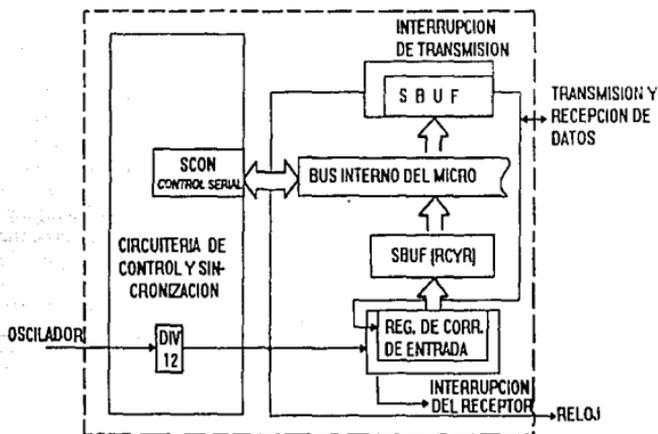


FIGURA III-17: Diagrama de bloques del puerto serial programado en Modo 0.

registro SCON (registro de control del puerto serial) al momento de la recepción.

MODO 2:

El conjunto de datos es de 11 bits, transmitidos a través de TXD y por RXD son recibidos. El dato está formado por:

- Un bit de inicio (= 0)
- 8 bits de datos
- Un bit extra
- Y un bit de paro (= 1)

El bit extra puede ser usado como paridad, puede tener un valor de 1 ó 0 asignado al bit TB8 del registro SCON y podemos cargarlo de registro PSW (Vgr. PSW.0, que es el bit de paridad) y moverlo al TB8. En la recepción, el bit de paro es ignorado y el bit extra se carga en el bit RB8 del registro SCON. El baudaje puede ser 1/32 ó 1/64 de la frecuencia de oscilación.

MODO 3:

Es simplemente lo mismo que el modo 2, solamente que con un baudaje más flexible.

Los modos de operación del puerto serial son elegidos a través del registro SCON. Cuando el registro SBUF es usado como destino de una instrucción, los modos programados en el registro SCON se ponen activos y se inicia la función del puerto serial.

Como puede verse es posible usar los modos 2 y 3 para poder comunicar muchos procesadores, donde después de recibir el 9o. bit, se recibe el bit de paro (= 1). Así el puerto serial puede ser programable para que la interrupción de puerto sea activado después de recibir el bit de paro sólo si el 9o. bit (asignado al bit RB8) esta activo (= 1), y usando el bit SM2 del registro SCON podemos diferenciar datos y direcciones en un procesador receptor. El bit SM2 no es afectado en el modo 0, y en el modo 1 puede ser usado como un bit de validación del bit de paro, por ejemplo si SM2 = 1, la interrupción de recepción no será activada hasta que el bit de paro es recibido. Para programar el puerto serial se utiliza un registro llamado SCON que contiene los bits de control y el registro de estado del puerto, como se observa en la figura III-18. El registro SCON contiene el modo de operación del puerto que es elegido de acuerdo a los 2 bits más significativos, almacena el 9o. bit del conjunto de datos transmitidos o recibidos en TB8 y RB8 y los bits de interrupción del puerto serial (TI y RI).

III.1.8.1 BAUDAJES DEL PUERTO SERIAL

El puerto serial puede transmitir/recibir a diferentes velocidades dependiendo de la forma de programación o de un agente externo como un temporizador.

En modo 0, la velocidad de transmisión esta dada por:

$$VT = \frac{\text{Frecuencia de oscilación}}{12}$$

En el modo 1 y 3 es utilizado el temporizador 1 como generador de tasa de baudaje, basado en el rango de sobreflujo (OV) del

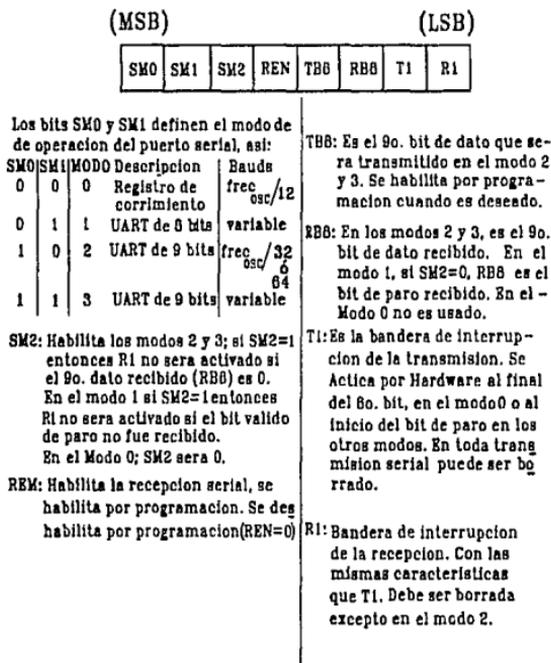


FIGURA III-18: Formato del registro SCON (Control del puerto serial).

contador y el valor del bit SMOD del registro PCON (1 ó 0) dado por la fórmula:

$$VT = \frac{2 \cdot \text{SMOD} \cdot (\text{Rango de QV})}{32}$$

La interrupción del temporizador 1 no se encuentra disponible, puede estar programado en cualquier modo y como contador o principalmente como temporizador en el modo de autocarga (TMOD=0010 ****) y la velocidad de transmisión estaría dada por:

$$VT = \frac{2 \cdot \text{SMOD} \cdot \text{Frec. Oscilación}}{32 \cdot (12X | 256 - (1111) |)}$$

Es posible obtener rangos bajos de baudajes usando el temporizador 1 teniendo disponible su interrupción y configurándolo como temporizador de 16 bits. (TMOD=0001****) y utilizando la interrupción del temporizador para hacer la carga de 16 bits por programación.

Finalmente, en el modo 2, la velocidad de transmisión depende del valor del bit SMOD del registro PCON, según lo siguiente:

$$VT = 2SMOD \frac{\text{Frec. de oscilación}}{64}$$

Donde tenemos que si SMOD = 0 (valor de reinicialización), la

$$VT = \frac{\text{Frec. Oscilación}}{64}$$

pero si SMOD = 1 entonces

$$VT = \frac{\text{Frec. Oscilación}}{32}$$

III.1.9 MODO DE OPERACION PASO A PASO

La estructura interna del sistema de interrupciones permite la ejecución paso a paso con muy pocas líneas de programación. Recordemos que una interrupción no será atendida mientras otra de igual o mayor prioridad está siendo atendida, si la instrucción es un RETI (retorno de una rutina de atención), será hasta que al menos otra instrucción sea ejecutada, cuando la interrupción sea atendida. Una forma de sacar provecho a ésta característica es para programar ejecuciones *paso a paso*, programando una interrupción externa (Vgr. INT0) activada por nivel ó si la rutina de atención terminará con el código siguiente:

```
JNB P3.2, X      ; espera la interrupción INT0 (= 1)
JB P3.2, X       ; espera el cambio de nivel INT0 (= 0)
RETI             ; regresa a ejecutar otra instrucción
```

Si en el pin INT0 tenemos normalmente una señal de 0 voltios, el procesador irá directamente a la rutina de interrupción externa 0 y se quedará allí hasta que halla un pulso en INT0 (de bajo a alto y de alto a bajo). Luego ejecutará la instrucción RETI,

regresará al programa llevando a cabo otra instrucción e inmediatamente volverá a la rutina de atención y esperará otro pulso en INT0, consecuentemente una instrucción (un paso) es ejecutado en cada pulso en el pin P3.2.

III.1.10 CONDICIONES DE REINICIALIZACION

El pin correspondiente a una señal de reinicialización para el microcontrolador es llamado RST, que contiene una entrada a un circuito tipo "schmitt-trigger"; la señal RST deber de durar por lo menos 2 ciclos de máquina (24 períodos de oscilación) para ser detectada y que el microcontrolador genere su señal interna de reinicialización.

La señal RST externa es asíncrona, y se verifica en el estado 3 fase 2 de cada ciclo de máquina. Después de habilitarse, el microcontrolador sigue trabajando durante 19 hasta 31 períodos

REGISTRO	CONTENIDO
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0-P3	0FFH
IP	***00000B
IE	0**00000B
TMOD	00H
TCON	00H
T0	00H
T1	00H
SCON	00H
SBUF	****
PCON (HMDS)	0*****B
PCON (CHMDS)	0***0000B

▪ Bit indeterminado
 B Palabra en binario
 H Palabra en hexadecimal

FIGURA III-19: Contenido de los registros internos del microcontrolador después de aplicarse la señal de reinicialización (RST).

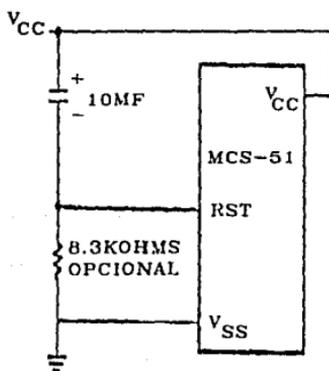


FIGURA III-20: Circuito de Reinicialización o RESET (pin RST del microcontrolador).

más, lo que son 19 períodos después de generarse la señal RST interna y las señales ALE y PSEN son puestas en nivel alto.

Si la señal externa RST es puesta en bajo, a las señales ALE y PSEN les toma de 1 a 2 ciclos de máquina sincronizarse con el reloj. Por ello, nunca deberán sincronizarse con el reloj interno, otros dispositivos externos. Las condiciones iniciales después de aplicarse la señal externa RST son las mostradas en la figura III-19. Y son mediante un algoritmo interno de reinicialización que se escriben 0's a todos los SFR's, excepto a los latches de los puertos, al apuntador de pila (SP) y al área temporal del puerto serial (SBUF). La RAM interna no es afectada y al encendido contiene valores indeterminados.

III.1.11 AUTO-INICIALIZACION

Es posible reinicializar al microcontrolador de manera automática al momento de encender el sistema, a través de un capacitor de 10 microfaradios conectado al pin RST y a Vcc, así como una resistencia de 8.2 K a tierra (figura III-20). El capacitor se carga en un microsegundo y el oscilador inicia en menos de un milisegundo. Cuando se alimenta al circuito, existe un voltaje alto en el pin RST por un período de tiempo, que depende del valor del capacitor. Para asegurarse de una señal de RST perfectamente detectable, debe estar en alto por un lapso de 2 ciclos de máquina (pocos milisegundos). Si reducimos el voltaje Vcc rápidamente hasta 0 voltios, nos causa un voltaje menor de 0 voltios en el pin RST, pero esto no causa daño alguno.

III.1.12 REGISTRO DE CONTROL DE CONSUMO DE POTENCIA (PCON)

Existen sistemas, sobre todo portátiles, donde el consumo de potencia es vital, crítico e importante; en las versiones CHMOS

existen modos de reducción de consumo de potencia; como característica general tenemos dos modos:

- Modo inactivo
- Modo de potencia mínima

Para la selección del modo de reducción de consumo de potencia se utiliza el registro PCON cuyo formato está en la figura III-21. En dispositivos elaborados con tecnología HMOS el registro PCON sólo tiene disponible el bit SMOD (el bit más significativo del registro PCON). En dispositivos CHMOS se tiene además cuatro bits disponibles (los bits menos significativos) para elegir el modo de reducción de consumo de potencia y dos banderas de propósito general.

MODO INACTIVO:

Del circuito anterior, si la señal IDL es un 1, el oscilador continua funcionando, y manda su señal a las interrupciones, al puerto serial y a los temporizadores, pero no al procesador. Cuando una instrucción activa al bit PCON.0 es la última instrucción ejecutada antes de ir al modo inactivo. La CPU guarda su estado: el SP, el PC, el PSW, el ACC y todos los registros de datos conservan sus contenidos. Los pines de los puertos retienen sus estados lógicos que tenían cuando el modo inactivo es puesto en marcha. Las señales ALE y PSEN se mantienen en niveles altos. Para poner fin al modo inactivo existen dos formas: activando cualquier interrupción que causaría que PCON.0 cambie a nivel bajo y al ser atendida, después de ejecutar un RETI; la siguiente instrucción sería la que continuara a la instrucción que activó el modo inactivo, ya que el registro PC mantiene su valor. Otra manera es con una señal de RST (reinicialización) ya que el valor inicial del bit PCON.0 es de 0.

MODO DE MINIMA POTENCIA DE CONSUMO:

Cuando el bit PCON.1 tiene un valor de 1, el microcontrolador entra en el modo de operación de consumo de mínima potencia, donde el oscilador es bloqueado por medio de una compuerta NAND (como se puede ver en el circuito anterior). Cuando una instrucción activa al bit PCON.1 (llamado también bit PD, siglas del término "power down"), ésta es la última que es ejecutada para ir al modo de mínima potencia. El oscilador es parado, y con ello

III.2 DISEÑO ELECTRONICO DEL SISTEMA MEV

Un *Módulo de Evaluación* (MEV) es un sistema complejo basado tradicionalmente en un microprocesador, sin embargo, en nuestro caso estará centrado en el manejo de un microcontrolador, por todas las razones dadas en el capítulo anterior. El MEV es una microcomputadora, entonces sus unidades funcionales son:

- Una unidad aritmética-lógica
- Una unidad de control
- La memoria
- Las unidades de entrada/salida (E/S)

Al usar un microcontrolador como parte central del sistema y recordando que dentro de un controlador de un sólo "chip" reemplaza toda la lógica digital que normalmente usamos para control y sincronización, entonces únicamente es necesario desarrollar diseños para las unidades de E/S.

Como el MEV es una microcomputadora, puede ser usado como una computadora para propósitos generales de bajo costo, para proporcionar capacidades similares a cualquier otro sistema de computación. El programa de una aplicación para propósitos especiales es inalterable y por ésta razón puede ser almacenado en una memoria de solo lectura (ROM).

La ejecución de los programas en ROM causa que el MEV se comporte de una manera preestablecida, justamente como se comportaría el sistema digital correspondiente basado en circuitos integrados MSI. Este método de diseño digital no era económicamente factible de configurar antes del desarrollo de los componentes de microcomputadoras pequeñas y de bajo costo. Además podremos ejecutar programas que se encuentren en memoria de lectura-escritura (RAM o RWM) lo que le da una gran flexibilidad al sistema.

El microcontrolador es un componente central en el sistema. La cantidad y tipo de memoria, lo mismo que la naturaleza de las unidades de interfaz de E/S que se usen, son una función de la aplicación particular.

El programa fijo que reside en la ROM del sistema en particular es también dependiente de la aplicación específica.

El diseño de todo sistema a este nivel puede ser dividido en dos partes: *Diseño de materiales o "hardware"* y *Diseño de programación o "Software"*.

En el *diseño de materiales y componentes electrónicos* se plantea la conexión de todos los dispositivos físicos para producir un sistema de desarrollo completo.

En el *diseño de programación* se trata del desarrollo de los programas de control, inicialización y de aplicación particular. Escribir programas a nivel máquina es esencialmente muy parecido que hacerlo para cualquier tipo de computadora, las diferencias radican en el conocimiento del dispositivo central (el microcontrolador) y en la familiarización con la configuración de los materiales y tener en cuenta los problemas asociados.

En esta parte definiremos los componentes necesarios del MEV y la forma en que se pueden interrelacionarse.

III.2.1 ORGANIZACION GENERAL DEL MEV

Un sistema tradicional de Evaluación consiste en un microprocesador más memoria y una interconexión de E/S. Los componentes se comunican a través de líneas donde se transfieren datos, direcciones y señales de control. Típicamente se tiene un solo microprocesador, pero si la aplicación lo requiere se pueden incluir varios microprocesadores teniendo un sistema multiprocesador.

Tenemos que el microprocesador se hace cargo de las funciones de la Unidad Central de Procesamiento y por lo tanto contiene una unidad aritmética-lógica y una unidad de control; las funciones de las tres unidades son llevadas a cabo por varios circuitos integrados, cuando se logra materializar una microcomputadora.

Así se tienen todas las ventajas de un sistema basado en CI's:

- mayor integración (sistemas más pequeños)
- más fiables (pocos componentes)

- menor consumo de potencia (tecnologías CMOS, etc.)
- menor costo (procesos de fabricación automatizados)

Ahora bien, el circuito central del MEV será un microcontrolador de la familia MCS-51, para su fabricación se requiere de alta tecnología; sucede que es de una estructura de alta complejidad, no obstante es un producto de uso masivo, pero que no es capaz de desempeñar ninguna función por sí solo, ésta le debe ser "comunicada" previamente lo cual ocurre gracias a las posibilidades que tiene el diseñador de circuitos para lograrlo, por lo tanto:

"Cada diseñador tiene la posibilidad de asignar al microprocesador, una función específica."

Pero esta función ya no está incluida en el propio CI, en cambio, lo está en un programa, o sea, en una secuencia de indicaciones para el microprocesador, almacenadas en uno o más módulos de memoria.

III.2.1.2 MODULOS DEL MEV

En esta parte, el problema global de diseñar al MEV, se modulariza en sus elementos funcionales, para resolverlo en forma sistemática. Las características principales del MEV son su portabilidad, funcionalidad e independencia total como sistema autónomo basado en un microcontrolador. El circuito central deberá monitorear diversas unidades de E/S en forma simultánea, como lo son un teclado y algún indicador visual de resultados. Siguiendo con la metodología formal en el diseño, se contemplan los módulos organizados según se muestra en la figura III-22.

Se tienen tres módulos: el microcontrolador, la memoria y las unidades de E/S. Los módulos se comunican entre sí a través de las líneas o "buses". A la vez, cada módulo estará integrado por varios subsistemas o submódulos, cuyas funciones específicas proveen resultados concretos al módulo que integran.

El primer módulo es el del microcontrolador, que controla a todo el sistema mediante su procesador, que a su vez contiene a la ALU encargada de realizar las operaciones secuenciales encargadas por el mundo exterior, tiene un subsistema de

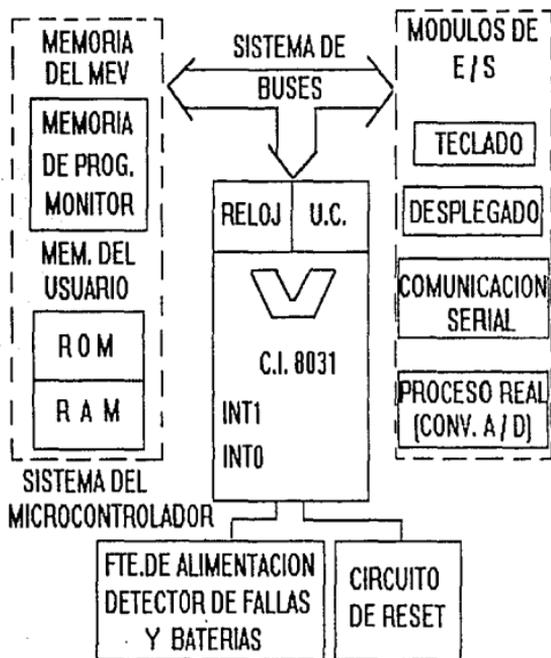


FIGURA III-22: Modularización por funciones del sistema de desarrollo MEV.

oscilación, que da la pauta de sincronización entre todos los componentes del MEV mediante una señal de reloj; un subsistema de control, que genera todas las señales de control de flujo de información y los niveles de comunicación entre los módulos; al encender el sistema, el subsistema de alimentación, permite conectar el MEV a la línea de corriente alterna, dando las características necesarias y suficientes para proveer a todas las partes del voltaje y corriente requeridos; y un subsistema de

detección de fallas de suministro de corriente que activa a una alimentación auxiliar de baterías.

Dentro del segundo módulo, la memoria exterior, se encuentran dos subsistemas: la memoria fija (no volátil) y la memoria de lectura-escritura. La memoria fija se utiliza sólo para leer información, por lo tanto es esencial cuando el sistema microcomputador inicia su labor, pero en esta memoria es imposible almacenar información nueva, generada durante el procesamiento de los programas, así en este sector de la memoria se tiene el programa de inicialización y control, las rutinas de usuario y los procesos interactivos con el usuario. El otro sector de la memoria, el subsistema de memoria de lectura-escritura, es para almacenar datos generados durante los procesos y para programas de usuario.

Las unidades de E/S conforman el tercer módulo y proveen de todos los medios de comunicación del MEV con su entorno: contemplan una comunicación visual con el usuario a través de alguna forma de despliegado de resultados; el operador puede comunicarse con el MEV mediante un sistema de lectura de datos, como lo es un teclado; con el mundo real se tiene un canal directo para adquisición de datos y con una computadora utilizando un puerto de interface serial, esto último para control de procesos industriales, etc.

III.2.2 DISEÑO DEL MÓDULO 1: EL MICROCONTROLADOR

El módulo 1 es el núcleo del sistema, por lo tanto supervisa todas las funciones y da las características principales del MEV: portabilidad y funcionalidad. Los subsistemas que componen al módulo son:

- *Subsistema de alimentación y detección de fallas de energía.*
- *Subsistema de oscilación.*
- *Subsistema de Control.*
- *Subsistema de reinicialización.*
- *Subsistema de bus y la CPU.*

La integración de todos los subsistemas para la conformación de las funciones del módulo se muestra en la **figura III-23**.

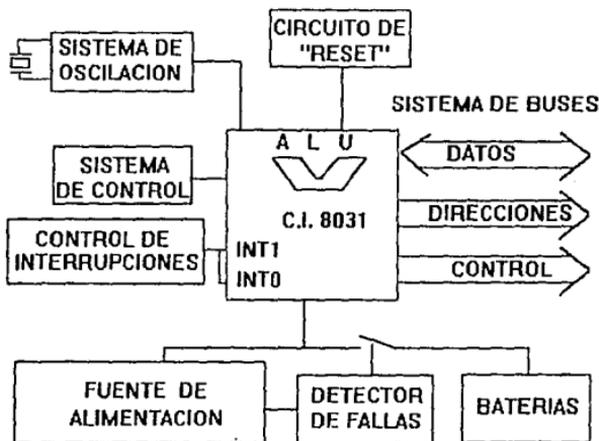


FIGURA III-23: División en subsistema del Modulo 1: El microcontrolador.

a) Subsistema de alimentación y detección de fallas de energía.

Considerando las ventajas de un equipo con fuente de alimentación dependiente de la línea de CA (Corriente alterna) tal como la portabilidad, se diseña un subsistema que tenga independencia de suministrar los voltajes y las corrientes convenientes para el MEV.

Así tomaremos en cuenta que el sistema toma la alimentación de la línea CA, pero es sustituida por un resguardo de alimentación por baterías en caso de detectarse alguna falla en la línea general. Además al reestablecerse las condiciones propias de corriente, se nulificará la acción de las baterías, sin cambios en la información, ni en el estado total del MEV.

Cuando el sistema MEV se encuentra bajo la alimentación de las baterías de resguardo podrá continuar operando bajo un estado de reducción de funciones, solamente monitoreando las

acciones principales como un transductor, o quizás manejando solamente un despliegue de datos, para que haya un ahorro de energía y un uso eficiente de la alimentación auxiliar. O puede mandarse al microcontrolador a un modo de ahorro máximo, salvándose la información crítica, hasta que se reestablezca la situación.

Siempre es necesario tener presente una falla ocasional que puede ser superada con la trasferencia ordenada a alimentación por baterías. Una advertencia de fallas se puede obtener mediante el monitoreo de la línea de CA, por una irregularidad en la salida del rectificador, o también monitoreando el voltaje regulado CD (corriente directa).

En el primer caso, monitorear la línea de CA, nos dará una advertencia a tiempo. De esta manera se puede conocer con 1 ó 2 medios ciclos de la frecuencia de la línea CA cuando el voltaje se cae. En muchos casos, ocurre otro medio ciclo, antes de que el voltaje regulado V_{cc} empiece a caerse. Tomando en cuenta que en 1/2 ciclo, el microcontrolador puede ejecutar 5,000 instrucciones -existe tiempo suficiente para arrancar una forma de alimentación auxiliar.

Partiendo de un voltaje de suministro CA, se puede desarrollar un voltaje CD de estado estacionario rectificando el voltaje de entrada CA, posteriormente filtrando a un nivel CD, y finalmente regulando con un circuito regulador de voltaje.

El circuito de rectificación convierte una señal de valor promedio cero a una con un valor promedio diferente de cero. La salida es una señal CC -corriente continua- pulsante, no es de CD pura y la operación en el MEV sería pobre, porque necesita un voltaje CD más liso. La línea de CA entra a un transformador que reduce el voltaje en un valor proporcional según un factor, la señal reducida en voltaje efectivo (V_{rms}) pasa a un circuito rectificador de onda completa con un factor de rizado menor, que consiste en un arreglo de cuatro diodos, que dan a la onda un valor promedio diferente de cero, sin embargo, la señal es todavía pulsante.

Un circuito filtro, muy utilizado y constituido por solamente un capacitor aplana a la señal pulsante, conectado a la salida del

circuito de rectificación; el voltaje DC se encuentra disponible a través del capacitor. La señal rectificada se encuentra ahora filtrada.

Los tiempos de carga y descarga del capacitor y las condiciones de inestabilidad dan una cierta impureza al voltaje CD obtenido hasta ahora. Mediante un circuito regulador se le da estabilidad al voltaje, aún a las variaciones de la carga. Un circuito regulador contiene una fuente de referencia, un amplificador de error, los dispositivos de control y la protección de sobrecarga, actualmente contenidos en un sólo CI. El regulador proporciona un voltaje positivo regulado (V_o) obtenido de un voltaje mayor no regulado (V_{in}), referenciados de manera común a una tierra; la diferencia de potencial entre la salida y la entrada permite al CI funcionar; en la última etapa se filtra con un capacitor pequeño para evitar el ruido.

El circuito detector y fallas y de activación de la batería de resguardo monitorea a la línea de CA mediante un diodo Zener (2N3904, que puede ser sustituido por un comparador de voltaje con referencia variable), cada medio ciclo y junto con el transistor para la información al MEV, y utilizando una interrupción externa (INT0) es recibida por el microcontrolador. Cuando un nivel de voltaje de línea CA alcanza un valor aceptable (alto), se abre el diodo zener, mandando al transistor a saturación e interrumpiendo al microcontrolador. La interrupción es un cambio a flanco positivo, entonces el microcontrolador ejecuta la rutina de servicio que recarga al contador con un valor que hará llegar entre uno y dos ciclos de la frecuencia de la línea. Así, si la línea de voltaje no tiene fallas, el contador nunca cambiaría, porque es recargado cada 1/2 ciclo, pero si en algún período la línea de voltaje no alcanza su valor normal, no se genera la interrupción y el contador genera una interrupción interna.

Al existir una interrupción, se hace la transición de alimentación a baterías. Los datos de mayor importancia son copiados a una RAM protegida por las baterías y de muy bajo consumo de potencia. Las señales que van a los circuitos que provocan la mayor disipación de energía deben tener 0's. Los circuitos protegidos con el voltaje de la batería (V_2) que se comunican con los circuitos no protegidos y alimentados con V_1 deben ser ignorados.

El microcontrolador debe ser puesto en modo de reducción de consumo, para responder a alguna interrupción. Se puede usar un interruptor manual para generar el restablecimiento del sistema, sin embargo, el mismo transistor lo puede generar. La rutina de servicio determinará su acción según el estado de los bits GFO y GF1 del registro PCON.

El circuito de baterías es automático, tal que a una caída de voltaje de la fuente sustituya, sin ningún retardo el voltaje de alimentación; el voltaje directo de la fuente es llamado V1 y alimenta a todo el sistema, el voltaje de las baterías es V2 y sólo llega a ciertos circuitos protegidos que contienen datos importantes.

Si la fuente no falla y $V1 = VBAT$, entonces $V1 = V2$ y las baterías no se encuentran en funcionamiento; sin embargo si existe alguna falla y $V1 \neq VBAT$, entonces $V2 = VBAT$ y las baterías alimentan al sistema protegido con voltaje de resguardo.

b) Subsistema de Oscilación.

El subsistema de oscilación genera una sola señal llamada reloj, que dará la pauta en tiempo y rapidez en la ejecución de las instrucciones. El oscilador maneja un generador interno de reloj, que provee de las señales internas de tiempo para todo el sistema MEV. Las señales de reloj son de 1/2 de la frecuencia del oscilador y definen los ciclos de máquina.

La señal de reloj generada, o simplemente el reloj, tendrá las características requeridas para el perfecto funcionamiento del microprocesador, como son una oscilación de una fase con niveles TTL, donde un "0" lógico tiene un voltaje máximo de +0.8 voltios y de 2.4 es el mínimo aceptable para un "1" lógico. El circuito externo oscilador se forma de la manera recomendada al inicio del capítulo, llamado oscilador de Pierce. El oscilador puede ser un cristal de cuarzo o un resonante cerámico, pero para el primer caso $C = 30 \text{ pf} \pm 10 \text{ pF}$ y en el segundo caso $C = 40 \pm 10 \text{ pF}$.

El oscilador es controlado por el circuito inversor-amplificador interno. Los valores de los capacitores no son críticos. El circuito interno de oscilación consiste en un amplificador de baja

frecuencia y ganancia de voltaje entre -10 y -20 voltios. La frecuencia de oscilación permitida es entre 3.5 y 12 Mhz. El oscilador elegido es de 4 Mhz para pruebas y finalmente de 11.059 Mhz para una perfecta comunicación serial. Los capacitores recomendados son de 30 pF.

c) Subsistema de control

El control de todo el sistema se genera a partir de la señal de reloj, dentro del microcontrolador, y su principal función es mantener la secuencia de eventos necesarios para poder efectuar cualquier procesamiento de datos. Por lo tanto, el sistema es síncrono.

Las señales de control forman el bus de control, que regulan las actividades del microprocesador (o microcontrolador), la cual es cíclica y siempre consiste en buscar y obtener datos e instrucciones y ejecutar a éstas últimas. Para iniciar la acción de procesamiento indicada por la instrucción, la circuitería de control manda tanto señales internas como externas a la CPU, así controla todas las unidades exteriores (memorias, interfaces, etc.), diferenciando el bus de datos del bus de direcciones, que físicamente se halla en el puerto P0.

Además la unidad de control responde a señales externas que alteran las funciones del microprocesador, ya sea interrumpiendo temporalmente o haciéndole ejecutar rutinas especiales como es el caso de una reinicialización. La unidad de control lo constituye primordialmente el generador de ciclos de máquina (GCM), que produce las señales de control.

Sin embargo, una de las grandes ventajas de los microprocesadores actuales es la de contener todo el sistema de control interno, únicamente es necesario un perfecto manejo de las señales que forma las líneas de control. Además recordemos que la unidad de control junto con la ALU físicamente se encuentran dentro del microprocesador.

d) Subsistema de Reinicialización.

La función del sistema de reinicialización (del término en inglés "reset") es generar una señal de control que deberá restablecer al

microprocesador y colocarlo en un estado inicial preestablecido, debido a que al encender por primera vez la fuente de alimentación, los registros y los "flip-flops" internos adoptan valores aleatorios y el funcionamiento es impredecible.

También es necesario tener una señal que pueda parar la ejecución de un programa de forma manual, evitando ejecuciones erróneas, ciclos infinitos, etc.

Al restablecerse el sistema, el microcontrolador regresa a la dirección más baja de memoria. La terminal de entrada de la señal de "reset" es designada como RST y es necesario que tenga las siguientes características:

- *Duración de al menos 2 ciclos de máquina (24 períodos de oscilación).*
- *Un pulso alto de tipo TTL.*

En esencial, una señal de reinicialización consiste en un pulso de nivel alto de voltaje ("1" lógico) por dos ciclos de máquina. Cuando en el pin RST existe un "1" lógico, el bus de datos y el de direcciones se encuentran en alta impedancia, y las de líneas de control pasan a un estado inactivo. Dado que es necesario un "reset" manual es preciso un interruptor conectado a Vcc, pero existen problemas de ruido.

No se utilizan circuitos Schmitt-trigger que introducen histéresis y aceleran la transición entre niveles lógicos, ya que internamente existe uno. El diodo tiene por objeto descargar rápidamente al capacitor, para asegurar la generación del pulso de restablecimiento aún en los casos en que se apague y se vuelva a encender.

Un auto-reset se ejecuta al encender el sistema que se logra cuando Vcc pasa a través de un capacitor de 10 microfaradios hasta que es cargado y conectado a tierra mediante una resistencia de 8.2 kohms.

Cuando se enciende la fuente de alimentación, la corriente fluye hasta cargar el capacitor. El voltaje en el pin RST es igual a la diferencia entre Vcc y el voltaje del capacitor, hasta un voltaje de carga (Vc). La señal de "reset" se hace más lento si el capacitor es mayor y deberá permanecer el tiempo necesario arriba del

voltaje del umbral del schmitt-trigger interno del microcontrolador.

Uniendo ambos circuitos obtenemos el diseño electrónico del sistema de reinicialización o de "reset".

e) Subsistema de bus y la CPU

Mediante tres conjuntos de líneas el microcontrolador puede comunicarse con los otros módulos, recibir instrucciones, dar resultados, tomar datos o almacenarlos.

La CPU está formado por la UC, por la ALU y por los registros de propósitos generales o específicos. Dada la importancia de la UC, se describió anteriormente como un subsistema independiente. La ALU lleva a cabo las operaciones matemáticas con uno o dos operandos.

El microprocesador, se comunica con los otros circuitos del sistema a través de grupos de líneas de interconexión (bus) y según el tipo de información que contengan se les denominan como buses de datos, de direcciones y de control.

El bus de datos, P0, es bidireccional y transporta información del microcontrolador a la memoria o a los puertos y de éstos al microcontrolador. El tamaño del "bus" es de 8 bits y se denominan desde D0 a D7 o también como P0.0 a P0.7. El microcontrolador es quien ordena que circuito puede mandar o recibir, es decir, usar el bus, para ello utilizamos líneas de control.

El bus de direcciones, P0 y P2, es unidireccional, sólo puede mandar datos a la CPU y todos los otros circuitos únicamente pueden recibirla. Mediante éstas líneas se define el mapa de memoria del MEV, como son 16 líneas se puede tener un mapa de 64 kbytes de memoria externa de datos y 64 kbytes de memoria externa para memoria de programas (si al tiempo de ejecución no se ocupa o no existe la memoria interna de programas). Ambas memorias ocupan las mismas 16 líneas de direccionamiento, la diferencia entre activar un bloque o el otro se lleva a cabo por señales de control como PSEN. Las 16 líneas de control están contenidas en dos puertos: P0 contiene la parte baja y P2

proporciona la parte alta de una dirección. Y mediante otra línea de control se difiere un dato de una dirección.

El bus de control está formado por las señales que permiten la sincronización y el sentido de la transferencia de información en el bus de datos (sentido del bus), el tipo de información y la diferenciación de un dato de una dirección. Es unidireccional, algunas son salidas y otras son entradas al microcontrolador.

Los buses internos son los contenidos dentro del microcontrolador y que comunican a la ALU, a los registros internos y a la UC.

III.2.3 MAPA DE MEMORIA DEL MEV

El microcontrolador, núcleo del MEV, tiene un mapa de memoria dividido en dos partes: memoria de programas y memoria de datos. Cada tipo de memoria con una longitud de 64 kbytes direccionables a través de 16 líneas, habilitadas por señales de control llamadas PSEN y ALE para los programas y WR y RD para los datos; entonces la memoria está dividida en dos sectores:

- *Memoria fija (no volátil).*
- *Memoria de acceso aleatorio o de lectura-escritura.*

Además, el bus de direcciones nos proporciona la localidad de memoria a acceder, el bus de datos contiene o contendrá la información circulante y el bus de datos regulará las funciones a llevarse a cabo con la información; todas las líneas mencionadas son comunicadas directamente a todas las partes de la memoria y al microcontrolador.

El bus de datos es de 8 líneas contenidas en el puerto P0, el bus de direcciones tiene 16 líneas, conteniéndose en el puerto P0, el byte bajo, y en P2, el byte superior; las líneas de control para manejo de memoria son las señales: PSEN, ALE, EA, RD y WR.

En la memoria fija -no volátil- solamente podemos leer información, mientras el microcontrolador trabaja puede hacer acceso a la memoria de programas en cualquier momento y leerla, pero no puede almacenar información nueva en ella. Además, si utilizamos alguna variedad de ROM, la información contenida no se pierde, ni aún cuando se interrumpa la alimentación. Es por ello que utilizamos una o más ROM's

cuando la información no es variable, ni deberá perderse, de este modo al momento de ser conectada, está lista para ser leída. Para introducir datos en una ROM, en general, se emplean aparatos especiales llamados programadores.

Dentro de la memoria de datos, una vez accesada una localidad, podemos realizar diversas funciones como leer, escribir o "borrar"; así utilizamos una RAM para almacenar los datos originados durante el funcionamiento del sistema. Una de sus desventajas de la RAM consiste en que la información se pierde al interrumpirse la alimentación.

III.2.3.1 MEMORIAS DE SEMICONDUCTORES

Las memorias de semiconductores, ya sean RAM's o ROM's, tienen diversas variedades que se han ido desarrollando según la tecnología (figura III-24), con nuevas características como la

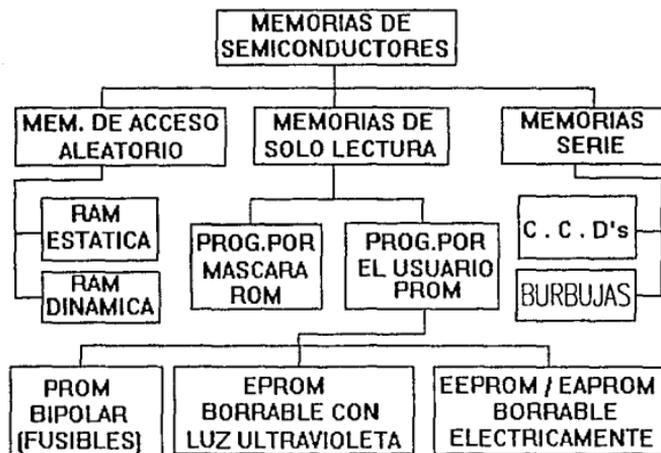


FIGURA III-24: Clasificación de las memorias de semiconductores.

reducción de tiempos de acceso, elevando la eficiencia de los sistemas y evitando tiempos muertos de procesamiento.

Para unir el área de memoria de programas y de datos, y para poder acceder RAM durante el ciclo de "FETCH" de programas de usuario, es necesario utilizar cierta lógica digital muy sencilla: unir las señales PSEN y RD mediante una compuerta AND (CI 74LS08). La señal obtenida es designada como OE -del término "Output Enable". Esto es porque PSEN habilita al área de programas para una lectura y RD habilita al área de datos. Como tienen lógica negativa la operación AND es una suma lógica y el

resultado es una solicitud de acceso a memoria (de datos o de programas) para una lectura. La escritura se habilita con la señal WR del microcontrolador y sólo le concierne a todas las RAM's.

III.2.3.2 DIRECCIONAMIENTO DE MEMORIA COMPLETA

El microcontrolador tiene 16 líneas para direccionar una localidad de memoria, así podemos acceder hasta 65 536 posiciones (ó 64 kbytes). Primero es necesario decodificar la dirección para tener activada la parte adecuada de memoria y alcanzar la posición deseada. Cada módulo de memoria, según su capacidad, decodifica internamente y nos da acceso a una sola celda, también tienen una entrada de habilitación del módulo (CE del término Chip Enable), que permite activar al módulo. Dada la compatibilidad de señales TTL, la lógica de decodificación externa se hace basado en circuitos de esta tecnología, como el 74LS138 (decodificador 3 a 8), que con tres líneas de entrada las decodifica a 8 líneas de salida, llamadas A, B, C y Y0,..., Y7 respectivamente.

Tenemos un mapa de 64 kbytes de memoria, si utilizamos memorias (RAM o ROM) de 2 kbytes cada módulo, para direccionar una celda de un módulo del mapa, se necesitan 11 líneas de direcciones (llamadas A0-A10) y una de habilitación para esa parte de memoria. Las líneas restantes (A11-A15) servirán para la decodificación de la memoria, por lo tanto es menester decodificar en dos niveles, en el primer nivel utilizaremos tres líneas (A11-A13) y en el segundo las líneas restantes (A14 y A15).

El circuito funcional de decodificación total de la memoria es un mapa de 64 kbytes dividido en módulos de 2 kbytes cada uno. El equipamiento es parcial y recomendaciones usuales indican que es conveniente ubicar a las ROM's en las direcciones más bajas y las RAM's sobre las más altas. En nuestro caso, la memoria necesaria es mínima y no es necesario utilizar todos los conductores de direcciones, por lo tanto, se hace una **decodificación incompleta** ahorrándonos ocho decodificadores, ya que utilizamos un direccionamiento lineal como señales de habilitación. Para casos donde se requiere mayor capacidad de memoria es posible superponer módulos como bancos y utilizar un sistema de conmutación.

III.2.3.3 DISEÑO DEL MAPA DE MEMORIA DEL MEV

El diseño anterior tiene las líneas de datos y direcciones separadas físicamente, lo que no es nuestro caso, ya que el puerto P0 tiene más de una función -contiene al bus de datos y parte del bus de direcciones-, para diferenciar la información del puerto P0 se utiliza un latch, que almacena datos del puerto P0 si y sólo si son parte de una dirección, esta función se lleva a cabo mediante la señal de control ALE, si los datos no son una dirección, se mantiene la información anterior. A la entrada del circuito integrado "latch" llegan los bits llamados AD0-AD7 o también P0.0-P0.7, que aún no sabemos si consisten en datos (D0-D7) o una parte de una dirección (A0-A7, del término "address"), a la salida tenemos la parte baja de la dirección de memoria actualmente accesada A0-A7, la habilitación se hace a través de ALE que al tener un nivel de un 1 lógico nos indica que en P0 se tiene parte de la dirección de la celda a acceder y por lo tanto la almacenamos en el "latch".

Para acceder una localidad se utilizan 16 líneas, el mapa finalmente es:

<i>Direcciones</i>	<i>Tipo de memoria</i>	<i>Uso</i>
0000-07FF	ROM	(Monitor)
0800-0FFF	RAM1	Usuario
1000- 17FF	RAM2	Pila de datos
1800 - FFFF		No utilizada

III.2.4 DISEÑO DE LOS MÓDULOS DE ENTRADA/SALIDA

En el transcurso de los años se han desarrollado los más variados módulos de E/S, en cada caso dedicados a una determinada aplicación. Estos módulos forman la interfaz entre el sistema MEV y su entorno (periferia). Una unidad de E/S se compone de uno o más circuitos integrados y abastecen de información en uno o ambos sentidos, por medio de una interfaz, a un periférico, suministrándole de las velocidades de información solicitadas.

Los módulos de E/S, también se seleccionan a través del bus de direcciones de 16 líneas, como las memorias. Existen dos tipos de organización de E/S:

- El método de E/S aislado
- El método de E/S de memoria mapeada

Así, la eficiencia de la microcomputadora depende de la organización de los módulos de E/S para comunicarse con el mundo exterior y proporcionar las respuestas que le son solicitadas. Los periféricos permiten que el sistema tenga una interacción, con el mundo físico o con el usuario mismo. Dispositivos tales como convertidores A/D y relevadores logran que el MEV interactúe sobre variables físicas como la presión arterial, la respiración, la temperatura de un fenómeno. En cambio periféricos como los teclados y monitores permiten que el operador programe y obtenga resultados del MEV. El MEV utiliza sus puertos para comunicarse, que consisten en un registro de 8 flip-flop's, para almacenar un byte de dato o en compuertas de tres estados para leerlo. Cuando leemos o escribimos a un puerto, es cuando el MEV se comunica, lo cual nos concede facultades de manejar cualquier periférico, lo complejo radica en los circuitos de acoplamiento electrónicos.

METODO DE E/S AISLADO

Cuando el microprocesador tiene un bus de N líneas (generalmente 8), es posible direccionar un puerto en forma específica de los canales (256 en este caso). Existen instrucciones de entrada (IN) y de salida de datos (OUT), donde se especifica el número del canal, siendo todos bidireccionales. Es posible decodificar al bus de canales mediante módulos de E/S o de decodificadores. El número de canales es limitado y unifuncional, es decir, dedicados a una función específica, así mismo, son

difíciles de examinar bit-a-bit y se utiliza al acumulador para cualquier transferencia de información.

IN CanalX Coloca el dato del canal X en el acumulador
OUT CanalY Coloca el dato del acumulador en el canal Y

METODO DE E/S DE MEMORIA MAPEADA

Se utiliza cuando el número de canales necesarios es mayor a los disponibles, utilizando las zonas de memoria no ocupadas y el bus de direcciones, para "mapear" nuevos canales de E/S. El espacio de memoria disminuye, las instrucciones de transferencia son orientadas a la memoria y podemos usar otros registros diferentes al acumulador. Es posible hacer de manera simultánea un proceso aritmético o lógico con los datos como se ve:

MVI M, 100 Llevar la expresión 100 al puerto de salida cuya dirección esta en los registros H y L.
MOV B,M Asigna un byte del canal de entrada (HL) al registro B.
ADD M Toma un byte del canal de entrada (HL) y lo suma al acumulador.

III.2.4.1 MODULOS DE E/S

Los periféricos pueden ser de entrada -el MEV solamente puede leer datos de él-, de salida -el MEV solamente puede mandarle datos- o de E/S -existe una comunicación bidireccional entre el MEV y el periférico-. La transferencia de información entre el microcontrolador, la memoria y los acoplamientos de E/S se realiza a través de los "buses", controlados siempre por el núcleo del sistema, el microcontrolador. De allí que se requieren interfaces para evitar el congestionamiento del procesador tanto en la entrada como en la salida de datos, principalmente en procesos simultáneos como son el sondeo del teclado y la emisión de datos en el despliegue de información. La potencialidad de un sistema electrónico complejo está directamente ligado con las facilidades que tenga para comunicarse con el medio que lo rodea: procesos reales, operadores y usuarios.

Existe gran cantidad de periféricos cuya elección depende de las aplicaciones del sistema. Las principales interfaces son:

a) Interfaces simples de señales:

Un módulo de salida simple de señales (como un "latch" o un cerrojo) tiene como función la de almacenar datos de aplicación constante, breve o simplemente muy bien definidos, hasta que se empiecen a procesar. En una forma muy popular de construir se puede utilizar flip-flop's en bancos para formar registros. Un ejemplo típico puede consistir en el CI 8212, con salidas de tres estados, por lo que el 8212 es un módulo simple de señales de E/S y puede transmitir una demanda de interrupción de programa (INT).

b) Interfaces serie y paralelo:

La transferencia de datos, en principio se diferencia entre comunicación en serie y comunicación en paralelo. En la interfaz paralelo se aceptan simultáneamente 8 bits y se transfieren en la misma forma, por lo tanto requerimos de 8 conductores para datos y otro tanto para control, lo que lo hace muy costoso para distancias mayores. Es recomendable que en el lado del transmisor los datos sean convertidos en un flujo serie y luego invertir el proceso en el receptor, reduciéndose el gasto de conductores.

El CI 8255A es un módulo de E/S paralelo programable (PPS) para conectar equipos periféricos con el bus de datos de la microcomputadora, por sus características de multifunción es innecesaria una lógica adicional, de tal forma que, su servicio se determina por medio del programa. Los servicios pueden ser de 3 clases:

- E/S paralela sin sincronización: para control de lámparas, relés, etc.
- E/S con sincronización, donde el microprocesador informa cuándo hay datos disponibles, el equipo avisa cuándo tomó los datos y viceversa.
- E/S bidireccional con sincronización.

El 8255 tiene 3 canales A, B y C, programables para entradas, salida o ambos. Para servicios sincronizados se ocupan las líneas del canal C, por lo tanto se pierde su capacidad de comunicación exterior. No se requiere de control de interrupciones porque el microcontrolador lee el canal C para saber el estado de las señales de recepción.

Para interfaz de comunicación serial se utiliza comúnmente al 8251A, donde la eficiencia exige se establezcan estipulaciones exactas que determinen las condiciones de transmisión llamado **protocolo**, como son la velocidad de transmisión (bit/seg, tasa de bauds o "baudaje"), el formato de los datos, la sincronización y el aseguramiento de los datos. La transmisión puede ser asíncrona o síncrona. Para la asíncrona se tiene una señal de inicio que habilita al receptor y se tiene una señal de finalización, es decir, existe un establecimiento de fases por signos. En el proceso síncrono se transmiten señales de sincronización preestablecida (fases por bloque) que se anteponen al dato.

c) Módulo de control de interrupciones:

En computación, una interrupción significa que se llama a un subprograma, pero la llamada no proviene de un programa sino del "hardware", como un equipo periférico, siempre más lento que el microprocesador; además el procesador tiene ciertas "libertades" como ignorar todas las interrupciones, ignorar sólo algunas, priorizarlas, hacerlas esperar hasta terminar su actual tarea, atenderla urgentemente, etc. Para descongestionar al procesador de estas "libertades" se utilizan controladores de interrupciones programables (PIC); dicho módulo envía al CPU las instrucciones de CALL completas. Un apoyo al control de interrupciones puede ser el CI 8259, puede tener funciones múltiples, logra atender hasta 8 interrupciones de programa vectorizadas y priorizadas, es posible conectar varios CI's 8259 en cascada, con un máximo de 64 interrupciones (un maestro y 8 esclavos). La función del 8259 consiste en recibir las demandas de interrupción de los periféricos, determinar su prioridad y en base de ello pasarla o no a la CPU. Si es aceptada por la CPU, el 8259 le manda el vector de la rutina de atención.

d) Módulos de temporización y conteo:

Sus funciones son, por una lado, retener la cantidad de veces que se producen determinados acontecimientos (contador), tal como la exploración de una banda, y durante el proceso de conteo la CPU no puede ser perturbada; por otro lado, deberá emitir pulsos y actuar como temporizador (del término "timer"), como un reloj de tiempo real, como un multiplicador de frecuencias, etc. La tarea de conteo puede ser implementada mediante una programación con interrupciones deshabilitadas, sin embargo, una labor que se presenta a menudo consiste en generar retardos exactos, donde la solución por programación, deberá calcular los tiempos de toda la secuencia de instrucciones a utilizar. El tiempo total se obtiene sumando todos los pulsos y multiplicándolos por el tiempo de un ciclo del microprocesador (si el reloj es de 2 MHz el tiempo de un ciclo es de 0.5 microsegundos), pero al tener RAM's dinámicas pueden existir dificultades, también existen otros inconvenientes, se requieren cambios en la programación cuando existen variaciones de la frecuencia del reloj, su ventaja radica en que no se necesitan dispositivos adicionales.

Las tareas de conteo, temporización y retardos pueden dejarse a cargo de un "hardware" adecuado como el CI 8253, que libera al CPU de esas labores; el 8253 es programable y tiene tres contadores de 16 bits cada uno, que realizan un conteo regresivo desde una carga constante por programación hasta cero, el conteo puede ser en hexadecimal o de tipo BCD, asimismo puede conectarse en cascada mediante la salida de "paso por cero" (Z0). La programación define una de las seis clases de servicio que se pueden obtener del 8253:

- Contador de entrada positiva: interrumpe el conteo cuando se cambia a un nivel bajo en la entrada.
- Monoflop: el conteo se inicia cuando a la entrada se detecta un cambio de nivel (de bajo a alto).
- Divisor de frecuencia: la frecuencia de entrada es dividida entre una constante cargada por el usuario.
- Generador de onda cuadrada: el ancho de pulso de salida esta en un nivel alto la primera mitad del conteo y la otro en un nivel bajo.
- Pulso retardado: se lleva un conteo de los pulsos de entrada por software hasta cero.

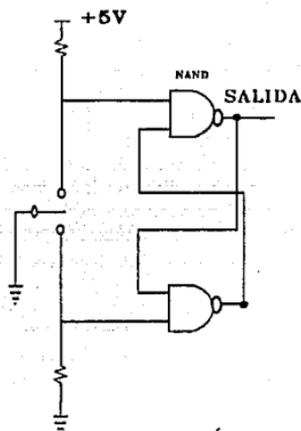


FIGURA III-25: Eliminación del rebote por medio de hardware.

- Pulso retardado inicializado por hardware.

e) Módulo para teclado, desplegado y señalización:

La interfaz hombre-máquina entre el sistema MEV y el usuario es, sin lugar a dudas, la más importante, sin embargo los procesos de visualización y entrada de datos por medio de un desplegado y un teclado son paralelos, lo que produce una limitante de atención por parte del microcontrolador. El módulo para teclado y señalización 8279 puede lograr

descongestionar al microcontrolador de éstas labores; tiene dos funciones principales: sondeo del teclado y emisión de señales, que son independientes entre sí y pueden funcionar en paralelo, también contiene una lógica antirrebotes (figura III-25 y III-26) que se pone en marcha después de pulsar una tecla.

El CI 8279 se comunica directamente con el bus de datos del procesador, además mediante la señal IRQ (petición de interrupción) le demanda atención, y a través de las señales de control como RD, WR, CS y Ao es posible mantener una comunicación con el circuito.

El 8279 proporciona tres tipos de servicios de entrada y un sólo servicio de salida. Los servicios de entrada son:

- Lectura del teclado
- Exploración de una matriz de sensores
- Lectura de pulsos (muestreo)

En la lectura del teclado, el 8279 explora una matriz de interruptores de hasta 64 contactos, tiene dos conductores adicionales SHIFT y CNTL, ambos de lectura directa, por lo que

se pueden asignar varios caracteres a cada tecla. Se tienen ocho conductores para las filas y otros ocho para las columnas; al pulsar una tecla, suministra 6 bits: 3 bits nos indican la fila y los otros 3 la columna de la tecla, ambos en código binario. Junto con el estado de las líneas SHIFT y CNTL, se almacena la posición de la tecla en una RAM de 8X8 bytes organizados en una estructura tipo FIFO. El CI 8279 tiene dos modos de lectura de teclado:

- *Modo 2-rebote de tecla ("key lockout")*
- *Modo N-teclas-presionadas*

En el modo de 2-rebote de tecla, se verifica otras 2 veces si hay más teclas pulsadas después del reconocimiento de una tecla, sino hubieron se toma como válida la primera pulsación. En el arreglo FIFO se pueden almacenar hasta 8 registros o pulsaciones, si existe sobre flujo (una novena tecla pulsada) se registra un error. Dentro de la estructura FIFO se tienen otros elementos almacenados como los errores, la cantidad de signos, etc.

Cuando se trabaja en el modo de N-teclas-presionadas podemos pulsar varias teclas de manera simultánea. Todas son almacenadas en el orden de la exploración, si se hallan activas en los siguientes dos rastreos, así podemos incorporar otra forma de

reconocimiento de errores.

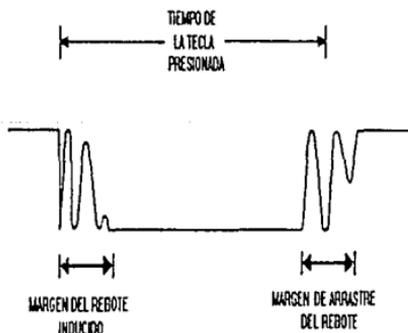


FIGURA III-26: Efecto de rebote al presionar una tecla.

Para el modo de exploración de una matriz de sensores, todos los estados de la matriz se encuentran registrados en una RAM de sensores (semejante a la RAM-FIFO), donde el estado de cada interruptor puede ser abierto o cerrado; si existe algún cambio de estado en cualquiera de los contactos, se genera un aviso sobre la línea IRQ y se almacena en la RAM, la nueva imagen de la matriz. Las líneas SHIFT y CNTL, así como la lógica antirrebotes no funcionan.

Cuando el 8279 trabaja en el modo de lectura de pulsos, cada línea RL0 a la RL7 se toman como entradas de pulsos, como un byte; sin embargo, el estado de las ocho líneas es registrado cuando en la conexión CNTL/STB se presenta un flanco ascendente de un pulso (reloj de muestreo).

El servicio de salida del CI 8279 tiene una RAM de 16X8 bytes para almacenar datos de salida para 16 señalizaciones alfanuméricas, para activar led's, despliegues de siete segmentos, etc. Las salidas se multiplexan por división de tiempo -activación de una tras otra-. Las señales se pueden oscurecer, activar los despliegues de derecha a izquierda o viceversa. En la RAM se encuentran los datos que el microprocesador va mandando, que pueden ser protegidos contra escritura, además podemos accederla hacia atrás, con auto-incremento mediante un acceso aleatorio.

D) Módulo de comunicación con procesos reales:

Los valores analógicos rodean a un sistema digital y para interactuar con ese entorno, en general, se desarrollan conversores que nos proporcionan magnitudes continuas y que a la vez procesen, de alguna manera, cantidades captadas de procesos como temperaturas, presiones, flujos, etc., o por medio de sensores convertirlas en señales eléctricas.

Como se ha visto, los procesos del ambiente no siempre son eléctricos, entonces, además del módulo de E/S se necesitan otras etapas: los sensores, que tienen tres unidades funcionales.

- Un transductor: Transforma la magnitud en una variable medible.
- Un convertidor: Convierte la variable medible en una cantidad eléctrica evaluable (corriente, voltaje, etc).
- Conformación de la señal: Manipulación de la cantidad eléctrica a una señal libre de ruido, amplificada, estable, etc.

En la etapa siguiente, se acopla al mundo digital utilizando convertidores del tipo analógico-digital (A/D), que hacen la conversión de las señales continuas suministradas por los sensores en señales digitales, que pueden ser manipuladas por el sistema de procesamiento de datos.

III.2.4.2 DISEÑO ELECTRONICO DE LOS MODULOS DE E/S DEL MEV

Las interfaces de E/S proveen una interacción entre el entorno natural y el MEV, y lo hacen un sistema electrónico complejo, dicha complejidad puede ser una barrera, sin embargo, una excelente comunicación con el exterior y una "amigable" respuesta y facilidad de manejo con el usuario pueden eliminarla, haciendo del sistema un elemento de gran utilidad. Los módulos de E/S, vistos en el punto anterior, son los que todo sistema de evaluación tiene -excepto por los convertidores A/D-, y que son incluidos en el MEV, pero no siempre como dispositivos electrónicos adicionales, ya que muchos de ellos se encuentran incluidos en el controlador.

A. INTERFACES SIMPLES DE SEÑALES:

Para la comunicación exterior tenemos 32 líneas de E/S, es decir, cuatro bytes de manera simultánea, cada byte forma un puerto de E/S, refiriéndonos a ellos como P0, P1, P2 y P3. Se puede acceder de manera individual, se puede tener bits de entrada y otros de salida en cada puerto. Todas las salidas y entradas son compatibles con TTL, el fan-in es 1 al igual que el fan-out, excepto para el puerto P0, cuyo fan-out es dos. Cada puerto tiene asignado un flip-flop, con un 0 si el pin es de salida y con un 1 cuando es bidireccional (E/S). Al aplicar una señal de reinicialización "RESET", todos los flip-flop's pasan a un nivel de 1, entonces todos los puertos contienen su propio latch y no es necesario ningún tipo de electrónica adicional.

B. INTERFACE SERIE Y PARALELO:

La comunicación paralela es costosa y poco funcional para distancias mayores, sin embargo, es posible dedicar uno de los 4 puertos a esta tarea. La interfaz serie, incluida en el microcontrolador, permite la transferencia síncrona o asíncrona de datos en forma dúplex, ya que incluye una UART dentro del circuito, y con un mínimo de dispositivos adicionales es posible adaptarnos a cualquier estandar elegido. Los datos son recibidos en la conexión RXD y se almacenan en un registro temporal, y cuando se tienen los bits de paro, se eliminan junto con los bits de inicio, filtrándose únicamente el dato, que es almacenado en el registro SBUF; mientras que el registro

temporal puede estar recibiendo otra transmisión, el registro SBUF avisa que ya tiene un dato mediante una interrupción a la CPU.

Los dispositivos adicionales consisten en acoplamientos al estándar elegido, por ejemplo el RS232, y que estriban en conversores de +5 voltios a -12 y de 0 a +12 voltios.

C. CONTROLADOR DE INTERRUPCIONES:

El microcontrolador puede responder a interrupciones de programa generadas, desde el exterior o desde el interior, existen cinco fuentes de interrupciones:

- Dos a través de las conexiones INTO e INT1.
- Dos por medio de los temporizadores.
- Una por medio de la interfaz serial.

Al demandar la interrupción, se salta a una dirección asignada de entrada fija en la memoria de programas -la parte más baja-. Las interrupciones se pueden priorizar, consultando al registro IPC, también se puede hacer una selectividad de ellas -flancos de activación-, accediendo al registro IEC. Todo este control de interrupciones se encuentra dentro del controlador y no requiere de elementos adicionales, solamente de establecer prioridades, máscaras y características de habilitación mediante la programación.

D. MODULOS DE TEMPORIZACION Y CONTEO:

Para contar eventos, medir intervalos de tiempo y generar las tasas de bauds, útiles en sistemas de control, el microcontrolador contiene 2 contadores/temporizadores de 16 bits, programables (en 4 modos) e independientes entre sí. El control de los contadores/temporizadores, así como ponerlos en marcha y detenerlos, se hace a través del registro TCON; y con el registro TMOD seleccionamos el modo de funcionamiento de cada uno. Ambos contadores son ascendentes y al pasar del límite -FFH ó FFFFH- a cero, existe un sobreflujo que se registra en las banderas TF0 y TF1 del registro TCON, además se genera una interrupción al CPU, también pueden ser revisados por programación y darnos cuenta del estados de cada contador/temporizador.

Para el conteo de eventos se tienen dos entradas T0 y T1 (de 0 a 500 KHz de frecuencia). Las condiciones de funcionamiento se establecen por programación y no se necesita ningún tipo de conexiones extras.

E. MODULO DE TECLADO Y SEÑALIZACION:

El conjunto de información visual, que el usuario, podrá obtener del MEV, consta, en su parte más básica, de 8 despliegues de 7 segmentos o "displays", manejados a través del módulo para teclado y señalización, el CI 8279, que se encuentra en la parte más alta de memoria desde la dirección 3800H. Para desplegar un dato éste se escribe en la dirección de memoria (3800H), donde un 0 es un led encendido y para habilitar un "display" se le manda un 1.

Segmento	Pin del 8279
A	B0
B	B1
C	B2
D	B3
E	A0
F	A1
G	A2
H	A3

Para el teclado matricial, se organiza en dos partes: teclas de control y teclas hexadecimales. Aunque el 8279 puede tener control de 64 teclas, el MEV únicamente utiliza 20 contactos. El teclado se encuentra organizado como se muestra en la **Figura III-27**. Mediante un control interno se genera un conteo de 000 a 111, que sondea al teclado. Si un 000 entra por el decodificador a la salida tenemos que $Y_0=0$, para todos los demás será un 1. Si ninguna tecla del renglón es presionada, en el latch existe solamente 1's, es decir, FFH. Si una tecla es pulsada se transfiere un 0 en esa columna, y se activa el circuito antirrebotes. Las funciones asignadas a las teclas se muestra en la **Figura III-28**. Donde, para escribir o leer la memoria RAM, se escribe la dirección deseada con 4 dígitos -por ejemplo 2000H- y se presionan las teclas [SHIFT] + [MEM/C]. Si el dato va a ser alterado, se tecléa el nuevo dato. Si se desea acceder la siguiente localidad se presiona [SIG] (siguiente), entonces la dirección se incrementa en uno, si se desea acceder una localidad previa se presiona [PREV] y la dirección se decrementa en uno.

Para acceder un registro de propósito específico del procesador, tal como el Acumulador (ACC ó A), se presiona [SHIFT] + [A]. Si se desea cambiar su valor basta con teclearlo.

Es posible, correr paso a paso un programa, para hacerlo tecleamos la dirección de inicio y después [SHIFT] + [PASO], lo que hará que se ejecute una sola instrucción y se devuelva el control al monitor.

La función de abortar la ejecución de un programa se hace a través de la tecla [ABORT].

La ejecución secuencial de un programa que se encuentra en RAM se logra tecleando la dirección de inicio del programa y después [SHIFT] + [RUN].

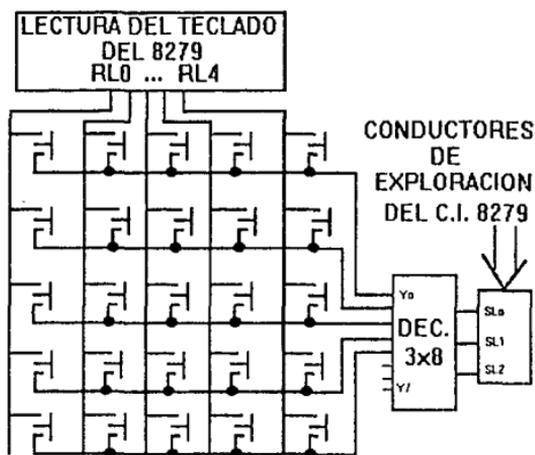


FIGURA III-27: Estructura electrónica y sondeo del teclado matricial del MEV

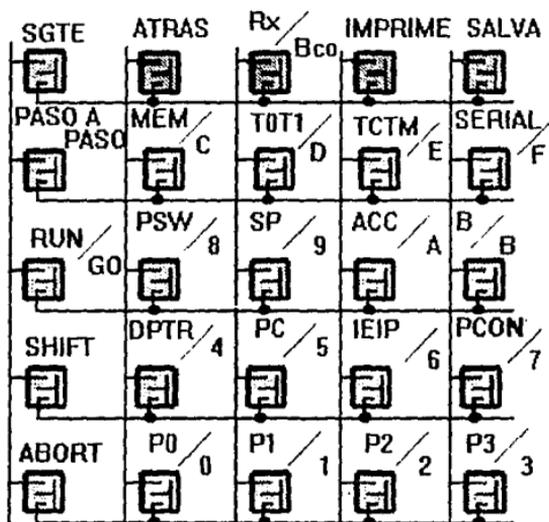


FIGURA III-28: Funciones de las teclas del MEV.

III.3 IMPLANTACION DEL MEV

A partir de 1950, los prototipos electrónicos se realizan en circuitos impresos, que consisten en placas de materiales sintéticos de tamaño variable con pistas de cobre que reemplazan los cableados, lográndose mejorar la fiabilidad del circuito y la reducción del costo de fabricación.

Todas las unidades funcionales se ubican sobre una única placa con todo lo necesario para requerimientos, medios de capacidad de memoria e interfaces para periféricos.

III.3.1 CONFORMACION MECANICA

Las características físicas del MEV que son implantadas con el fin de darle la mayor flexibilidad son las siguientes:

- El conjunto modular tiene un ancho de 20.32 mm, formado por las alturas de las bases, los CI's y las patas de las bases.
- El intercambio de información con los equipos periféricos se efectúa a través de bornes.
- El intercambio de información entre los módulos internos -memorias, microcontrolador, etc.- se realiza mediante el bus integrado por 8 líneas (cableado de cara posterior) y los contactos del bus.
- Para todos los circuitos se han previsto zócalos (bases), especialmente para los integrados mayores (memorias, microcontrolador e interfaces).
- El ajuste de funciones sobre los conjuntos modulares -áreas de direccionamiento- se lleva a cabo con interruptores -deep switch-ó de terminales.

III.3.1.1 IMPLANTACION DEL MODULO 1: EL MICROCONTROLADOR

El módulo 1 forma el centro del sistema, ya que consiste en el microcontrolador, principalmente, y junto con los otros 5 submódulos integran las funciones más importantes del MEV. La relación del material electrónico y de CI's necesarios para su construcción es dada a continuación:

- Una tableta fenólica perforada de 21.5 X 27.8 cm.

- Cuatro postes de 0.8 cm
- Tres bases para wire-wrap (de patas largas) con 40, 16 y 14 pines.
- 2 jacks de diferentes colores (preferentemente uno rojo y otro negro)
- Un diodo emisor de luz (led)
- Una base para led
- Dos interruptores: uno de un polo, dos tiros y otro tipo push- button
- Un cristal de 11.0595 Mhz
- Un diodo IN914
- Cuatro capacitores de bajo voltaje de las siguientes medidas:
 - 2 de 30 picofaradios
 - 1 de 0.01 microfaradios (de tantalio)
 - 1 de 33 microfaradios
- Tres resistencias de 1/4 W:
 - 2 de 10 Kilohms
 - 1 de 330 ohms
- Un CI 74LS00 (compuertas NAND)
- Un rollo de alambre wire-wrap

a) Preparación de la Tableta:

A la tableta fenólica perforada se le colocan los cuatro postes en las esquinas, para apoyo y protección de las patas de las bases y del cableado. Para situar los postes, se perfora con un taladro y una broca de 1/8. A continuación se colocan los jacks, para alimentación auxiliar mediante una fuente independiente, el jack rojo para el voltaje positivo (+5 voltios) y el negro para la referencia (0 voltios). A partir del jack rojo se conecta el interruptor de un polo, dos tiros, y servirá como "switch" de encendido. Se cablea el led y la resistencia de 330 ohms, que será la luz de indicación de encendido.

b) Preparación de la base para el microcontrolador:

Se colocan las dos bases, la de 16 pines a la altura del 12o. pin de la otra base. La base pequeña contendrá los dispositivos

electrónicos del circuito de oscilación y del circuito de reinicialización. Según los diagramas del diseño se cablea los circuitos de reinicialización y de sincronización utilizando cable wire-wrap y el enrollador.

III.3.1.2 IMPLANTACION DEL MODULO 2: LA MEMORIA DEL MEV

El mapa de memoria del MEV consiste en dos bloques: la memoria del programa monitor (no volátil) y la memoria del usuario (memoria de lectura-escritura), ambas se comunican con el módulo 1 a través del sistema de buses formado por tres conjuntos de líneas. A continuación se muestra la lista de los dispositivos electrónicos necesarios para la realización del módulo:

- Cinco bases para wire-wrap (patas largas):
 - Dos de 24 pines
 - Una de 20 pines
 - Una de 16 pines
 - Una de 14 pines
- Una memoria EPROM 2716 (2 KbytesX8)
- Una memoria RAM 6116 (2 kbytesX8)
- Un CI 74LS373 (latch)
- Un CI 74LS138 (decodificador 3 a 8)
- Dos compuertas NAND (proporcionadas por el CI 74LS00 del módulo anterior)
- Dos capacitores de tantalio de 0.01 microfaradios
- Un rollo de alambre wire-wrap

Para la implantación, el módulo se ha dividido en tres etapas.

a) Primera etapa: *Demultiplexación de datos y direcciones.*

Se coloca la base de 20 pines a la altura del pin 40 de la base del microcontrolador y se cablea todo el puerto P0 a las entradas del latch, también cableamos las polarizaciones y la habilitación que es la señal ALE/P del controlador (pin 30) al conector 11 de la base del latch. Así a la salida del latch tenemos únicamente la parte baja del bus de direcciones.

b) Segunda etapa: Decodificación de la memoria

Se unen las señales de PSEN y RD a través de una NAND (del módulo 1) y se introduce el resultado a otra compuerta NAND, para obtener finalmente una operación AND. Se coloca la base de 16 pines para de decodificador de 3 a 8 líneas, se polariza y se conectan a las entradas A, B y C las direcciones A14, A13 y A12 respectivamente. Los habilitadores son conectados a tierra. Con esto hemos paginado la memoria en ocho partes de 2 kbytes cada una.

c) Tercera etapa: Las memorias.

Se colocan las bases de 24 pines y se cablean sus polarizaciones (pines 12 y 24). A continuación se conectan el bus de direcciones del latch a las bases de las memorias. El bus de datos se hace común con el puerto P0 del controlador y la entrada del latch. Para activar la RAM se utiliza el resultado de la NAND negada de la primera etapa. Finalmente hacemos útil la paginación y conectamos las salidas del decodificador a las memorias. Las salidas no utilizadas ahora, lo serán en la implantación de las interfaces de E/S.

III.3.1.3 IMPLANTACION DEL MODULO 3: LAS INTERFACES DE E/S

El método de diseño de E/S usado es el de memoria mapeada, por lo tanto se utiliza parte de la paginación de la memoria -el sector más alto- generado por el decodificador del módulo 2 implantado en la segunda etapa. Para el módulo 3 se necesitan los siguientes componentes:

- Ocho bases para wire-wrap (patas largas):
 - Una de 40 pines
 - Dos de 28 pines
 - Una de 24 pines
 - Una de 20 pines
 - Una de 16 pines
 - Dos de 14 pines
- Un CI 8279 (interfaz de teclado y señalización)
- Un CI 74LS240 (buffer)
- Un CI 74LS138 (decodificador 3 a 8)
- Un CI 1488 (convertidores TTL a RS232C)
- Un CI 1489 (convertidores RS232C a TTL)

- Siete resistencias de 330 ohms
- Cuatro despliegues de siete segmentos de dos dígitos ánodo común (8710)
- 25 microteclas
- Dos capacitores de tantalio de 0.01 microfaradios
- Dos jacks (para +/- 12 voltios)
- Conectores RS232C con cable plano
- Un rollo de alambre wire-wrap

El conjunto de interfaces de E/S se basan en el CI 8279, que principalmente descongestiona las peticiones de atención hacia el microcontrolador. Se implantado en tres etapas:

a) La interfaz 8279:

Se coloca la base de 40 pines y se cablean las polarizaciones, utilizando una compuerta NAND del 74LS00 del módulo 1 se conecta IRQ, obtenemos IRQ negada que se alambra con el pin 12 del controlador (INT0). Después conectamos todo el puerto P0 con las entradas DB0 a DB7. Las señales RD, WR y RST son las mismas que las del controlador. La habilitación (CS) se obtiene del decodificador de la memoria (Y7). La A0 se obtiene después de demultiplexarla (a la salida del latch). Todas las salidas son utilizadas en las dos etapas siguientes.

b) El despliegado de información:

Las bases de 28 pines y la de 24 son colocados en la tableta y se alambra de manera común cada segmento A, B, C, D, E, F, G y H, y a su vez a los pines de salida del CI 8279 que corresponden a las salidas OA0-OA3 y OB0-OB3 a través de las etapas de amplificación para cada segmento y la resistencia de 330 ohms. Las salidas SL0-SL2 definen, en binario, el dígito a encender, uno de los ocho disponibles en el MEV, para habilitar físicamente uno y sólo uno de ellos a la vez, el valor es decodificado y mandado al conjunto de buffers, todo esto significa colocar las dos bases restantes -de 16 y 20 pines-, cablear sus polarizaciones, las entradas del decodificador con las líneas SL0-SL2, las salidas del decodificador a las entradas del 74LS240 y sus salidas a los cátodos comunes de cada dígito.

c) El teclado matricial:

Una tecla contiene cuatro terminales o pines, sin embargo, dos a dos son comunes y por lo tanto solamente se tienen dos terminales físicamente disponibles. Se colocan en forma de una matriz de 5X5 teclas, todos los renglones son alambrados a una línea en común e igualmente con todas las columnas. Cada renglón es conectado con la etapa anterior, a los pines de salida del decodificador 74LS138, en las terminales Y0 a Y3, mediante estas líneas es sensado el teclado. A través de las columnas se lee la tecla presionada que es un dato aceptado por el CI 8279 en los pines referenciados como RL0 a RL4

III.3.2 RENTABILIDAD DE LA IMPLANTACION

Los gastos del prototipo tienen cuatro fuentes principalmente:

- Los costos de los materiales (20-25 %).
- El costo del diseñador (medio año-hombre).
- Costos del montaje, según las posibilidades de equipamiento.
- Gastos de ensayos (proceso de pruebas).

Para una producción menor de 400 sistemas no sería rentable.

III.4 PRUEBAS DE FUNCIONAMIENTO

La calidad y fiabilidad de un instrumento dedicado a la medición, control o regulación de un proceso, donde la intervención humana es casi o completamente nula, se basa en un conjunto de pruebas de funcionamiento, permitiendo que el diseño y construcción del prototipo llegue a su punto culminante.

Las etapas de pruebas, pueden ser tantas como se deseen, tales como: preensayo (sin tensión de alimentación), prueba funcional, prueba comparativa con grupos de referencia, prueba de memoria, prueba de los módulos analógicos, prueba del sistema (junto con otros módulos) y la verificación final.

III.4.1 MEDIOS DE PRUEBAS

Los medios de pruebas consisten en sistemas electrónicos de apoyo en la búsqueda de fallas que impidan el funcionamiento

del prototipo en construcción, a continuación se presentan a los medios de pruebas utilizados:

- **Control paso a paso:** utilizando como indicadores de los estados lógicos del bus, dispositivos tales como diodos emisores de luz (led's) e indicadores de siete segmentos (display's) es posible ir verificando los niveles lógicos de las líneas.
- **Programas prueba:** Tales como son bloques del programa monitor y programas de aplicación ya terminados. Utilizamos interrupciones en direcciones preelegidas (puntos de ruptura) y a continuación se revisa el estado de los registros internos -dump-.
- **Punta lógica:** Se utiliza como trazador de señales y basados en la indicación del "led" de la punta, se verifican los potenciales instantáneos de las líneas.
- **Osciloscopio y analizador de estados lógicos:** Equipo especial para circuitos lógicos para seguir de manera simultánea diez o más señales, mediante un diagrama de líneas o muestras de bits -binarios o hexadecimales-.

III.4.2 PRINCIPIO GENERAL DE PROCEDIMIENTO

Un sistema electrónico que es incapaz de funcionar adopta un estado estable después de ser conectado a la fuente de alimentación o de ser encendido, entonces podemos tener los siguientes casos más comunes:

- El microcontrolador se encuentra en estado de alto (HALT).
- El microcontrolador no llega a recoger ninguna instrucción de la ROM.
- El microprocesador corre en un bucle cerrado -loop infinito-.

Debemos recordar que nuestro sistema es realimentado y consiste en:

CPU--> bus de direcciones--> ROM--> bus de datos--> CPU
Para interrumpir el círculo de trabajo del sistema y realizar el proceso de pruebas, deben estar conectados únicamente las partes electrónicas que en ese instante se desean analizar. Así evitamos que como consecuencia de los cortocircuitos se dañen otros dispositivos conectados en el mismo bus.

- Se deben incluir bombes de medición sobre la placa.
- Tenemos que retirar los módulos que no se encuentran en análisis.

Para nuestro caso la interrupción del círculo se puede realizar mediante el retiro de módulos de alimentación -buffer's-.

III.4.3 CASOS DE FALLAS

1) El microcontrolador se encuentra en estado de alto (HALT).

Según el microcontrolador en el pin de RST no llega el aviso de reinicialización, con la consecuencia de no poder salir de su estado latente de espera, hasta no aplicar la señal de RESET. Para este caso debemos revisar el voltaje en el pin número 9 y el nivel de ruido.

2) El microcontrolador no llega a recoger ninguna instrucción de la ROM.

Para resolver este problema debemos seguir la siguiente secuencia de verificaciones:

- Controlador la tensión de alimentación en el microcontrolador.
- Verificar la existencia de pulsos de reloj -pines 18 y 19- y si la oscilación es correcta -en frecuencia y características eléctricas-.
- Comprobar si la entrada RST -pin 9- se encuentra habilitada -nivel 0-, en caso contrario el contador de instrucciones (PC) estará permanentemente en cero.
- Controlar la señal de HOLD (que se encuentre activá y no conmuta a los bucles al estado de alta resistencia).
- Comprobar si las entradas de interrupción están inactivas -INT0, INT1, T0 y T1 todas en nivel lógico alto-.
- Ver si las señales de estado del procesador dan otras indicaciones.

3) El microcontrolador corre en un bucle cerrado.

Primero: se analiza el bus de direcciones:

- Todos los datos ponerlos en 0.
- Simular una instrucción NOP (No operación).

Dado que nuestro bus es multiplexado, generamos nuestra instrucción NOP por un elemento de tres estados es activado mediante la señal ALE. El efecto es que únicamente se incrementa el registro PC, lo cual repetimos sin interrupciones desde 0000H hasta FFFFH, así detectamos los cortocircuitos.

Segundo: Verificar la inexistencia de cortocircuitos en el bus de datos.

- Colocar en el bus de datos unos y ceros alternados.
- Poner en un EPROM, la muestra de 1's y 0's.
- Ejecutar una instrucción NOP

Mediante un bucle, ejecutar este proceso por un tiempo determinado.

Tercero: Prueba de módulos de RAM:

- Lectura de una celda de memoria RAM.
- Escribir un dato en esa misma celda.
- Leerla con un valor invertido.
- Comparar lo leído.
- Escribir nuevamente el valor original.
- Leerla nuevamente.
- Compara con el valor original.

La ventaja del método es que el valor original de la RAM sigue inalterado.

Cuarto: Prueba de los canales de E/S:

- Interconectar los canales de salida con los de entrada.
- Leer los valores de los canales de salida a través de los canales de entrada.
- Comparar los valores obtenidos.

III.4.4 AUTODIAGNOSTICO

El autodiagnóstico consiste en una prueba del sistema con indicación de fallas que se desarrolla automáticamente al conectarse el sistema y es una importante característica de calidad en sistemas computarizados. El sistema de diagnóstico de perfecto funcionamiento del prototipo debe correr inmediatamente después de conectar el equipo -arranque en frío- y antes del programa monitor, aunque es posible intercalar estas rutinas de diagnóstico durante el servicio del prototipo.

III.4.4.1 PRUEBAS DE DIAGNOSTICO

Diagnóstico de la memoria de instrucciones:

Interpretar a todos los bytes de instrucciones de la memoria del programa como números mediante un programa de pruebas.

- Sumar todos los números interpretados.
- Compararlo con la suma calculada por el programador y almacenado en una EPROM.

Diagnóstico de la RAM:

Se puede utilizar el proceso de prueba de la RAM, perfectamente descrito anteriormente.

Diagnóstico de los canales de E/S:

Solamente adherir un circuito multiplexor para desconectar los equipos periféricos y proceder con el método de pruebas de canales de E/S ya presentado.

CAPITULO IV

DISEÑO DEL PROGRAMA MONITOR DEL SISTEMA DE DESARROLLO: MEV

En un sistema controlado por una microcomputadora, la parte de programación absorbe hasta un 80% de los costos totales, por lo que su desarrollo y diseño efectivo es muy importante. Como un programa monitor es complejo y tiene un gran número de funciones debe ser dividido en componentes parciales llamados estructuras, con las siguientes características:

- Cada bloque estructural tiene una sola entrada y una salida (dipolo).
- Un conjunto de bloques estructurales conforman un programa.
- Las instrucciones de salto deben ser evitadas o limitadas en alcance.

A continuación se muestran las ventajas proporcionadas por la estructuración del sistema:

- La estructura resultante es jerárquica.
- El programa puede modificarse fácilmente.
- El programa es más fácil de probar y mantener.
- El desarrollo del programa es legible.
- Debe suministrar un plano constructivo, es decir, un modelo del software a construir.

El nivel de software a desarrollar en el más bajo, ya que se encuentra íntimamente ligado con el procesador a usar, por lo tanto, es necesario un conocimiento detallado de la forma organizada del microcontrolador y los requerimientos de programación útiles.

El microcontrolador separa el espacio de programas de la memoria de datos. La instrucción MOVX accesa a la memoria externa de datos. Dentro del CI existen 128 bytes de RAM y los SFR's. La parte baja de la RAM puede ser accesada por direccionamiento directo (Vgr. MOV dir-del-dato, destino) o por direccionamiento indirecto (Vgr. MOV @Ri, destino).

IV.1 CONJUNTO DE INSTRUCCIONES DEL MICROCONTROLADOR

El conjunto o "SET" de instrucciones de los microcontroladores de la familia MCS-51 está constituido por 111 instrucciones clasificadas por su longitud así:

- 49 instrucciones de un sólo byte
- 45 de dos bytes
- 17 de tres bytes

El formato de una instrucción está formado por un mnemónico, un operando destino y un operando fuente, que especifican de manera implícita el tipo de dato y el modo de direccionamiento utilizado. Las instrucciones que un microcontrolador 8051 puede ejecutar son de alguno de los cuatro grupos funcionales: transferencia de datos, instrucciones para operaciones aritméticas, instrucciones para operaciones lógicas e instrucciones de control de secuencia.

a) Instrucciones de transferencia de datos:

Consisten en operaciones directas de manipulación de la información y pueden ser de tres tipos: de propósito general, transferencia con el acumulador y transferencia indirecta sobre una localidad de memoria; ninguna de ellas afecta a la palabra de estado del procesador -PSW-, excepto las operaciones directas sobre ese registro (Vgr. MOV PSW,#30).

Las instrucciones de transferencia de propósito general consisten en:

MOV Transfiere un bit o un byte desde un operando fuente a un operando destino.

PUSH Transfiere el byte del operando fuente a la localidad actualmente apuntada por el registro SP, después de incrementarlo.

POP Transfiere el byte de la localidad apuntada por el registro SP al operando destino y decrementa al registro SP.

El registro ACC o A es utilizado como operando -fuente o destino-en las instrucciones de transferencia con el acumulador:

XCH Intercambia el byte del operando fuente con el acumulador.

XCHD Intercambia la mitad menos significativa del byte del operando fuente con la mitad menos significativa del acumulador.

MOVX Ejecuta una transferencia de un byte entre la memoria entera de datos y el acumulador. La dirección se especifica en el registro DPTR de 16 bits o en R1 y R0 de 8 bits.

MOVC Transfiere un byte desde la memoria del programa al registro A. El operando contenido en A es usado como un índice de la tabla apuntada por el registro DPTR o por PC.

Las instrucciones de transferencia indirecta sobre una localidad de memoria consiste en la operación MOV DPTR, #dato, que carga el dato inmediato (de 16 bits) en los registros DPH y DPL.

b) Instrucciones para operaciones aritméticas:

Las cuatro operaciones matemáticas pueden ser ejecutadas con operandos sin signos de 8 bits, sin embargo, mediante las banderas de sobreflujo -OV- es posible hacer operaciones de suma y resta con enteros con signo. Las operaciones pueden ser ejecutadas sobre números representados en BCD. La operación de adición se puede realizar a través de las instrucciones:

INC Incrementa en uno al operando fuente, donde es dejado el resultado.

ADD Suma el valor del acumulador al operando fuente y el resultado es colocado en el acumulador.

ADDC Es igual que ADD, pero suma un uno si la bandera de acarreo se halla habilitada (CY = 1).

DA Suma dos operandos en BCD, el resultado se encuentra en A. La bandera de acarreo es activada si el resultado es mayor que 99.

Para ejecutar una operación de substracción entre dos números enteros sin signos se utilizan los siguientes mnemónicos:

DEC Resta uno al operando fuente, colocándole el resultado.

SUBB ("Subtract with borrow") Resta el segundo operando al primero (ACC) y subtrae uno si la bandera de acarreo está activa, el resultado es colocado en el acumulador.

La multiplicación se lleva a cabo mediante el mnemónico MUL que utiliza los registros A y B como operandos, sin signos; el resultado es un número de dos bytes: en el registro A se encuentra el byte menos significativo y en el registro B el más significativo. La bandera de OV nos indica si el resultado es cero o no, el bit de acarreo es puesto en cero y el bit AC -acarreo auxiliar- no es afectado.

El mnemónico DIV ejecuta una división de dos enteros sin signos contenidos en los registros A y B. El cociente es puesto en A y el residuo en B. Una división entre cero deja datos indeterminados en ambos registros y activa la bandera de OV.

c) Instrucciones para operaciones lógicas:

El 8051 puede ejecutar operaciones lógicas de uno o de dos operandos, sobre un bit o un byte completo, gracias a su procesador booleano. Las operaciones lógicas de un sólo operando son:

CLR Limpia (pone en ceros) un bit especificado o todo el byte contenido en el acumulador.

SETB Activa (pone en unos) cualquier bit direccionable de manera directa.

CPL Complementa el valor del registro A sin afectar las banderas de estado.

RL RLC RR RRC SWAP Ejecutan operaciones de rotación sobre el acumulador: a la izquierda (RL), a la derecha (RR),

con acarreo (RLC y RRC) y cuatro rotaciones consecutivas a la izquierda (SWAP).

Para las operaciones lógicas con dos operandos se utilizan las siguientes instrucciones:

ANL Ejecuta una operación lógica AND entre los dos operandos fuentes, en bits específicos o todo el byte, y coloca el resultado en el primer operando.

ORL Ejecuta una operación lógica OR entre dos operandos fuente, con bytes o bits específicos, y deja el resultado en el primer operando.

XRL Ejecuta una operación XOR entre los bytes de los operandos fuente y el resultado queda en el primer operando.

d) Instrucciones de control de secuencia:

Causan, según alguna condición específica, la ejecución de instrucciones de un programa en una manera secuencial o no, y pueden consistir en tres tipos de instrucciones: llamadas a subrutinas y saltos incondicionales, saltos condicionales e interrupciones.

Las llamadas a subrutinas -CALL-, los saltos -JUMP- y los retornos de subrutinas -RET- transfieren el control del PC actual a una dirección destino directa o indirecta de una manera incondicional:

ACALL y LCALL Inserta la dirección de la siguiente instrucción dentro de la pila y transfiere el control a la nueva dirección. La instrucción ACALL es usada cuando el salto se encuentra en la página actual de hasta 2 kilobytes, porque utiliza 2 bytes de código. Un salto largo se lleva a cabo con un LCALL, que cubre los 64 kilobytes del espacio de programas (con 3 bytes de código de operación). Cuando una instrucción ACALL es ejecutada, el dato inmediato es concatenado a los 5 bits más significativos del PC.

RET Transfiere el control a una dirección anteriormente salvada en la pila mediante una operación "CALL", decrementa el SP en dos y toma ambos datos para formar una dirección efectiva.

AJMP LJMP y SJMP Transfiere el control a una dirección destino dada por el operando. Los prefijos indican la longitud máxima del salto válido: "S" hasta 256 bytes (un espacio de

-128 a +127 de la siguiente instrucción), "A" hasta 2 kilobytes y L todo el área de memoria direccionable (64 kbytes).

JMP @A + DPTR Ejecuta un salto relativo al registro DPTR, el offset está contenido en A con un rango de 0 hasta 255; el destino puede ser cualquier espacio de memoria de programas.

Los saltos condicionales son ejecutados si y sólo si la condición es verdadera, la dirección efectiva estará en un rango de 256 bytes (-128 a +127 de la dirección siguiente):

JZ Salta si el acumulador es cero.

JNZ Salta si el acumulador no es cero

JC Salta si la bandera de acarreo esta activada.

JNC Salta si la bandera de acarreo no esta activada.

JB Salta si el bit (direccionable en forma directa) está activo.

JNB Salta si el bit (direccionable en forma directa) no está activo.

JBC Salta si el bit (direccionable en forma directa) está activo y a continuación lo desactiva.

CJNE Compara el primer operando con el segundo y salta si son diferentes. La bandera de acarreo se activa si el primer operando es menor que el segundo. Las comparaciones pueden ser hechas entre bytes direccionados de manera directa en la memoria interna de datos o entre un valor inmediato y el registro A o un registro indirecto direccionable por bytes de la RAM interna.

DJNZ Decrementa el operando fuente y salta si el resultado no es cero. El operando debe ser cualquier byte de la memoria interna de datos.

Un operando fuente puede ser direccionado usando un registro o mediante direccionamiento directo.

El retorno de interrupciones transfiere el control a la siguiente instrucción de donde se generó la interrupción mediante la instrucción RETI y permite la habilitación de interrupciones de mayor prioridad.

IV.1.1 MODOS DE DIRECCIONAMIENTO

La memoria de programas y la memoria de datos comparten en parte o totalmente las mismas direcciones, lo cual no es significativo, ya que ambas áreas de memoria se mantienen

separadas durante los diferentes tipos de direccionamiento dentro del conjunto de instrucciones.

IV.1.1.1 MODOS DE DIRECCIONAMIENTO DE LA MEMORIA DE PROGRAMAS

El registro PC sirve para direccionar la memoria de programas, contiene 16 bits y para el secuenciamiento del programa se tienen tres tipos de direccionamiento:

- a) En el *direccionamiento relativo*, el procesador calcula la dirección de memoria, a través de un valor de 8 bits que sigue a la instrucción de ramificación. A los 8 bits faltantes (msb) se les asigna el mismo valor que el séptimo bit (27); a continuación se le suman los 16 bits al valor actual del contador de instrucción.
- b) *Direccionamiento absoluto*: mediante 11 bits se obtiene una dirección de la actual página con una longitud de hasta 2 kbytes.
- c) *Direccionamiento largo*: el procesador utiliza 16 bits, lo que permite un salto a cualquier localidad de la memoria para programas (64K).

Para efectuar cualquiera de los tipos de direccionamiento se utilizan los registros de base y de índice; los registros de base son el DPTR (Data Pointer) y el contador de instrucción (PC).

IV.1.1.2 MODOS DE DIRECCIONAMIENTO DE LA MEMORIA INTERNA DE DATOS

Para la memoria dedicada a los datos se tienen cuatro tipos de direccionamiento por bytes y una por bits:

- a) *Direccionamiento de registros*: podemos acceder directamente a los registros de la memoria interna de datos dependiendo del banco activo especificado en la palabra de estado del programa (PSW).
- b) *Direccionamiento directo*: con dos tipos de operandos -bytes o bits-:

Direccionamiento directo por bytes: el valor de la dirección se encuentra entre 0 y 255 y podemos tener acceso al área interna de datos, así como a los SFR's, ya sea con el nombre (dirección simbólica) o por medio de la dirección física.

Direccionamiento directo por bits: permite el acceso a escritura y lectura de 128 bits de la memoria interna de datos y a 128 bits del sector de la memoria interna de los SFR's.

c) *Direccionamiento indirecto:* mediante los registros y el símbolo @, la CPU va a un registro indicado en la instrucción para buscar la dirección necesaria para el acceso de memoria.

d) *Asignación inmediata de valor:* nos permite tener un valor determinado o constante de 8 ó 16 bits en una posición específica de memoria.

IV.1.1.3 DIRECCIONAMIENTO DE MEMORIA EXTERNA DE DATOS

Para el intercambio de información con el exterior se utiliza al acumulador y la memoria externa a través de la instrucción MOVX y el direccionamiento indirecto.

IV.1.1.4 AREA DE SOLAMENTE DIRECCIONAMIENTO INDIRECTO

Consiste en la parte alta de la RAM interna y está contenida entre las direcciones 80H y FFH, dividida en dos sectores debido a sus dos formas diferentes de acceso: direccionamiento activo y direccionamiento indirecto.

MOV 80H, #AAH ; P0 = AAH
; direccionamiento activo

MOV R0, #80H
MOV @R0, #BBH ; (80H) = BBH
; direccionamiento indirecto

IV.1.1.5 AREA DE DIRECCIONAMIENTO DIRECTO E INDIRECTO

El área de RAM interna con 128 bytes, puede ser dividida en tres sectores:

1) *Los cuatro bancos de 8 registros cada uno:* contenido entre las direcciones 00H a 1FH (32 bytes), los registros son llamados R0 a R7. El registro SP está inicialmente

apuntando a la dirección 07H, al hacer una operación de PUSH, el primer elemento de la pila es insertado en la dirección 08H, que es también el lugar físico del registro R0 del segundo banco. Cuando se utilicen los otros bancos es conveniente inicializar el registro SP fuera de los bancos de registros (parte alta de la RAM).

2) *Area direccionable bit-a-bit*: consiste en 16 bytes contenidos entre las direcciones 20H y 2FH, direccionables directamente. Podemos hacer referencia a éstos bits de dos formas: por su dirección física (00H-7FH) o por su byte (20H a 2FH). Para accesarlos bit-a-bit especificamos la posición del bit a efectuar como 0 a 7, que son los bits 20.0 a 20.7, y los siguientes bits desde 08H a 0FH son los mismos que 21.0 a 21.7.

3) *Area de resguardo*: en la RAM interna de datos podemos utilizar desde las direcciones 30H a 7FH (80 bytes) como área para la pila de datos y prevenir su destrucción.

Los registros de funciones especiales son 21 y algunos son direccionables por bits. Algunos SFR's tienen alguna tarea específica como la programación de puertos, de los contadores/temporizadores, etc.

IV.2 DISEÑO DEL PROGRAMA MONITOR

La división de un problema complejo en estructuras funcionales es un modo de ver superior, para determinar el orden del desarrollo. Finalmente se sigue con el afinamiento, hasta la conversión de los elementos estructurales en instrucciones de máquina. La estructuración deseada, se logra si se evitan las instrucciones de salto o se limita el alcance de los mismos a una estructura o bloque. El programa monitor del MEV tiene tres funciones principales:

- *Inicializar el sistema*
- *Autodiagnóstico*
- *Atención de comandos del usuario.*

IV.2.1 SISTEMA MONITOR GENERAL

El sistema MONITOR es una de las dos partes que conforman el software del MEV, consiste en un programa de ambientación, es decir, crea un ambiente interno y externo de relación y comunicación con su entorno. Para establecer las condiciones de comunicación con el usuario, primero establece sus parámetros iniciales, después verifica sus componentes y finalmente se pone a disposición del usuario, todo este proceso se puede apreciar en la figura IV-1.

IV.2.2 DISEÑO DEL SISTEMA DE INICIALIZACION DEL MEV

Un programa de inicialización establece los parámetros iniciales del microcontrolador, ya que después de una señal de reinicialización (RST) muchos de sus registros contienen valores indeterminados o no convenientes para nuestros propósitos.

En la parte baja del espacio de programas se tienen reservadas localidades físicas para las interrupciones y sus rutinas de servicio. El sistema de inicialización consiste en:

- Para la señal de RST se tiene la localidad 00, donde se encuentra una instrucción de salto al programa de inicialización de los registros de propósito general, para evitar el acceso a las localidades reservadas por las otras interrupciones.
- Las rutinas de inicialización de las interrupciones contienen solamente saltos a direcciones, aún no definidas, de las rutinas de atención ya que entre una y otra sólo existen 8 bytes de memoria, lo que nos restringe a no colocar allí la rutina de atención real.
- Debido a que el valor del registro SP es 07H, al inicializarse el sistema, y que es el mismo lugar físico del registro R0 del segundo banco, el apuntador de la pila es reubicado a la parte alta de la RAM (50H).

Todos los registros de control son inicializados con el objetivo de tener una forma de comportamiento preestablecida para el MEV:

PWS = 00H Establece las condiciones iniciales del microprocesador y activa el banco 0 de registros.

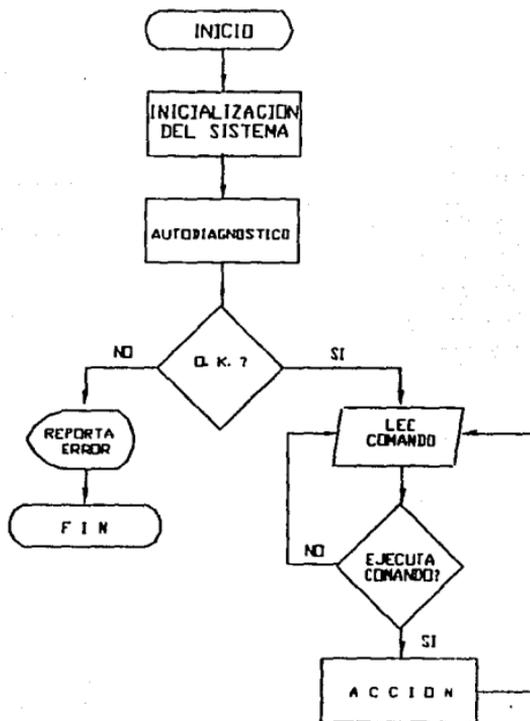


FIGURA IV-1: Diagrama de procesos del sistema monitor del MEV.

PCON = 00H Indica que actualmente no se utiliza ningún modo auxiliar de consumo de potencia.

IE = 01H Nos habilita únicamente la interrupción 0.

IP = 01H Asigna la prioridad más alta a la interrupción 0.

TCON = 01H Flanco positivo para activación de la interrupción 0.

*SCON = *** No importa ya que la interrupción del puerto serial se encuentra deshabilitada.

Se programan las interfaces de E/S siguiendo la secuencia de pasos dada a continuación:

Se manda la palabra de control utilizando al acumulador.

Se hace referencia a la memoria (dirección 3800H).

Se programa la entrada por la izquierda (modalidad del 8279).

Se limpia la RAM de datos de las interfaces.

Finalmente se despliega un mensaje de la terminación exitosa del proceso de inicialización del MEV, el diagrama de procesos del sistema se puede observar en la figura IV-2.

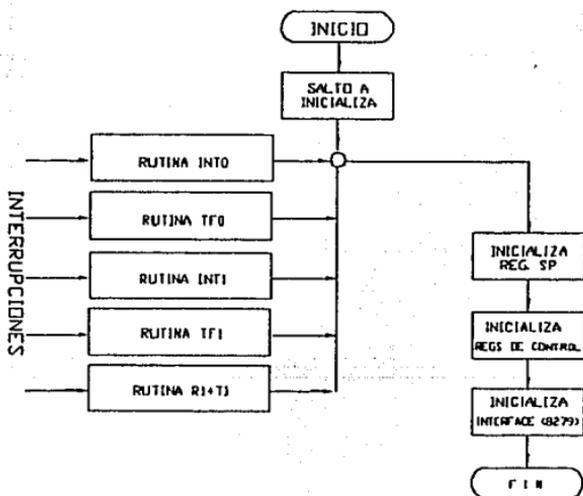


FIGURA IV-2: Diagrama de procesos del sistema de inicialización del MEV.

IV.2.3 DISEÑO DEL SISTEMA DE AUTODIAGNOSTICO

Una de las ventajas de los sistemas basados en microprocesadores es su "inteligencia", tal como lo demuestran en una autoverificación al encendido o al tiempo de operación. Casi todos los dispositivos del mercado se ocupan primeramente de un diagnóstico interno antes de ponerse al servicio total del usuario, checar su configuración o ponerla a punto.

El sistema de autodiagnóstico (figura IV-3) nos proporciona información muy valiosa sobre el estado actual del prototipo, después del encendido -arranque en frío-, tal como:

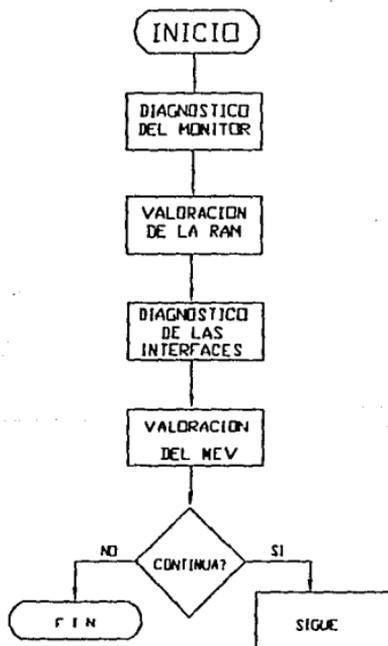


FIGURA IV-3: Diagrama de procesos del sistema de autodiagnóstico del MEV.

- La conservación intacta del programa monitor, es decir, el perfecto estado de la EPROM.
- Las zonas de RAM accesibles para lecturas y escrituras de información.
- Los bancos de memoria disponibles.
- El estado de las interfaces.

a) *Autovaloración del programa monitor*: consiste en el monitoreo de toda la memoria no volátil, para checar la conservación del programa monitor y detectar alguna falla. La programación de la autoverificación de la EPROM interpreta a todos los bytes del código de operación del programa monitor como números enteros que son sumados y el resultado es comparado con una suma calculada por el programador, llamado SUMCHECK, que puede ser hecha por sectores para facilitar la búsqueda de errores.

b) *Valoración de los bancos de RAM externa al servicio del usuario*: la indicación de los bancos disponibles de RAM externa para el usuario es hecha mediante una revisión de todas las localidades de memoria asignadas a programas de aplicación, lo que nos evita los accesos inválidos.

c) *Valoración de las interfaces*: el microcontrolador tiene gran flexibilidad en el manejo de los periféricos, como lo es el puerto serial. Para su diagnóstico simplemente se procede a "interconectar", por software los canales de salida con los de entrada para verificar que los valores recibidos sean los mismos que los enviados.

Como resultado del autodiagnóstico exitoso se despliegan los mensajes adecuados, en caso contrario se especifica el error encontrado.

IV.2.4 DISEÑO DEL SISTEMA DE COMANDOS DE LINEA

Un comando de línea consiste en una secuencia definida de pulsaciones o de teclas presionadas, que si es correcta puede solicitar alguna información adicional, y nos permite interactuar con el MEV. La secuencia de teclas activadas definen una función a ser ejecutada por el MEV, cuando es correcta, si no lo es, necesitamos volver a definirla.

La identificación de la acción, la revisión del comando y su lectura corren a cargo del sistema de comandos de línea del MEV, cuyo repertorio de funciones, así como la relación con el usuario proveen de la mayor parte de la potencialidad del prototipo; todos sus procesos son mostrados en la figura IV-4.

IV.2.4.1. PROCESO DE HABILITACION DEL TECLADO

El teclado esta formado por una matriz de interruptores, que al ser activado uno de ellos hacen una petición de atención al procesador por lo cual el mecanismo inicial para "aceptar" un comando consiste en habilitar el teclado, mediante el siguiente proceso:

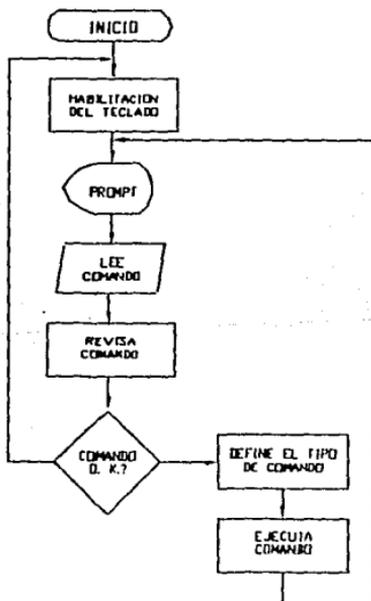


FIGURA IV-4: Diagrama de procesos del sistema de comandos de línea del MEV.

Habilitar las peticiones de interrupción (vía INT0)

Asignar la prioridad de interrupción

Atención de la interrupción por medio de una rutina de atención

La rutina de servicio es la asignada en el proceso de inicialización y está formada por 2 bytes, la rutina de atención, propiamente consisten en:

Inicializar el apuntador DPTR (dirección 3801)

Programación de la interface:

Palabra de control para lectura de la RAM-FIFO (40H)

Acceso a la palabra de datos (lectura de la RAM)

Acceso a la tabla de conversión

Enmascarar los bits necesarios del dato leído

Regreso a la instrucción siguiente a la interrupción

Esta rutina nos permite obtener la conversión de la posición de la tecla presionada (renglón y columna) al valor de la tecla (una A, un 2, etc.) mediante un mapeo con una tabla de datos.

IV.2.4.2 PROCESO DE DESPLEGADO DEL PROMPT

Cuando el MEV se encuentra en condiciones de recibir un nuevo comando se muestra un PROMPT en uno de los despliegues de 7 segmentos y consiste en un guión en el primer "display". El estado de los 8 despliegues se encuentra dentro de un banco de registros, así el contenido de R0 es desplegado en el dígito V0, el de R1 en V1 y así sucesivamente, sin embargo, la visualización de información no es directa ya que, se tiene que escribir en la RAM de la interface, el proceso es:

Salvar el contenido del acumulador

Activar el banco de registros asignado a los despliegues

Almacenar en los registros los datos a desplegar

Transformar el dato a siete segmentos

Inicialiar el contador de datos a desplegar

Inicializar el apuntador de la interface (DPTR = 3801)

Programar la interface:

Palabra de control de escritura con autoincremento (90H)

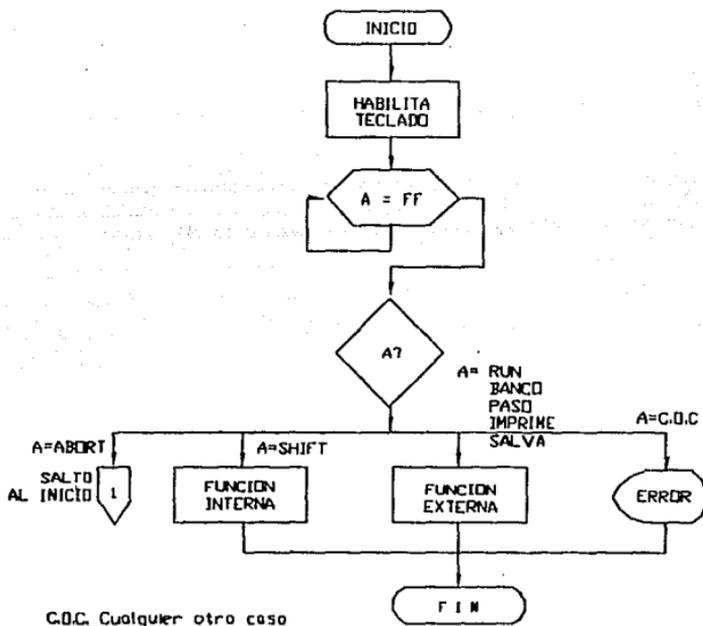


FIGURA IV-5: Diagrama de procesos del Sistema de Comandos de Línea.

*Mandar la palabra de control
 Accesar el registro de la palabra de datos
 Mientras no se hallan desplegado todos los datos
 Asignar al acumulador el datos a desplegar
 Mandarlos a la interface
 Regreso*

La conversión del dato tiene lugar cuando la información consiste en un valor numérico que debe ser mapeado, en este caso, el dato es un guión cuya conversión es un 01H.

IV.2.4.3 PROCESO DE LECTURA DEL COMANDO DE LINEA

Cuando el prompt se encuentra desplegado, nos indica que el MEV se encuentra listo para recibir un comando, el cual puede ser de uno de los tres tipos (figura IV-5):

- Comando de "abort"
- Comando de función interna
- Comando de función externa

Cualquier otro tipo de secuencia de teclas, no definidas, se toma como un "ERROR". La lectura de un comando consiste en:

Inicialización del registro A con FFH

Habilitación de la interrupción asignada al teclado (INT0)

Mientras el valor del registro A no sea diferente de FFH

Hacer una lectura de la interface de teclas

Convertir el valor leído a un valor manipulable

Así, el valor de la tecla es mapeado en una tabla y obtenemos su valor funcional para el MEV.

a) El comando "ABORT": la secuencia es únicamente presionar la tecla marcada con su nombre, lo que nos permite detener un comando mal tecleado, una ejecución errónea y la obtención nuevamente del "prompt" del MEV (salto al proceso de desplegado del "prompt").

b) Un comando de función interna: es aquel que nos va a permitir manipular los valores de los registros internos del microcontrolador y de las localidades de la RAM del MEV (figura IV-6). Así dentro del proceso de funciones internas del MEV es posible llevar a cabo una de las siguientes tareas:

- Regresar al sistema -desplegado del prompt-
- Acceso a la RAM interna y externa del MEV
- Acceso a todos los registros de funciones especiales del microcontrolador
- Acceso a cualquier registro de usuario del banco actualmente activo.

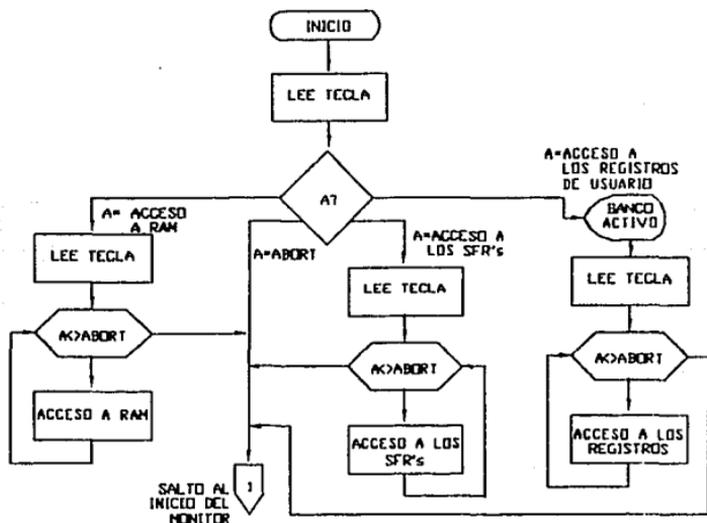


FIGURA IV-6: Diagrama del sistema de procesamiento de las funciones internas del MEV.

Para el regreso al sistema, basta con presionar en cualquier momento, la tecla marcada como "ABORT". Para iniciar la ejecución del proceso de comandos de funciones internas es necesario presionar la tecla "SHIFT". Si haremos acceso a la memoria del MEV se presiona la tecla "C" y cuando queremos acceder memoria interna del microcontrolador, proporcionamos una dirección de dos dígitos (un byte) o de cuatro dígitos (dos bytes) en el caso de la memoria RAM externa. Las direcciones son leídas de la parte hexadecimal del teclado y verificadas, contra accesos indebidos (los bancos superiores son utilizados por el MEV para almacenar su propia información, por ello solamente se encuentran tres bancos disponibles), así como un proceso de "regreso" para cuando se presionan teclas

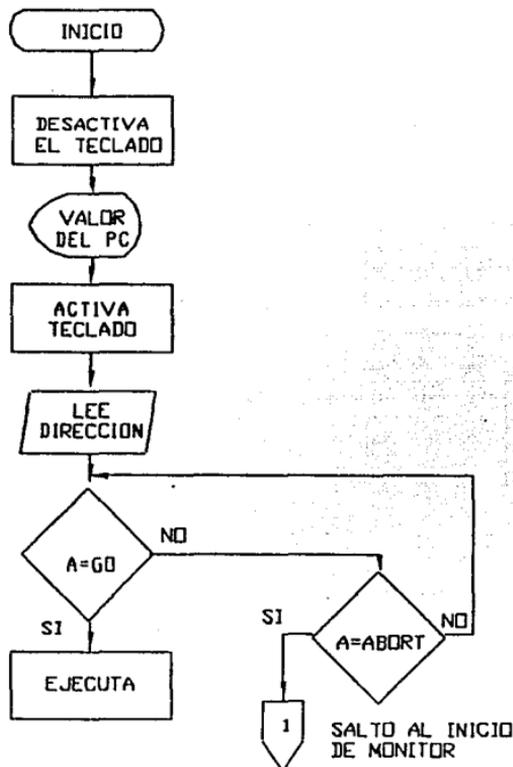


FIGURA IV-7: Diagrama de procesos de la ejecución libre de un programa de aplicación almacenado en la RAM externa del MEV.

equivocadas. En cambio, si queremos tener acceso a los SFR's, después de presionar "SHIFT", directamente continuamos con la lectura del registro deseado, marcado en las teclas como PC, SP, PO, etc.; es conveniente notar que al desplegarse la información contenida en los SFR's existen registros de 16 bits (PC, DPTR) y por lo tanto la información contenida es de cuatro dígitos, además ciertos registros fueron unidos, para reducir la cantidad de teclas (IEIP nos desplegará los valores del registro IE seguido de los

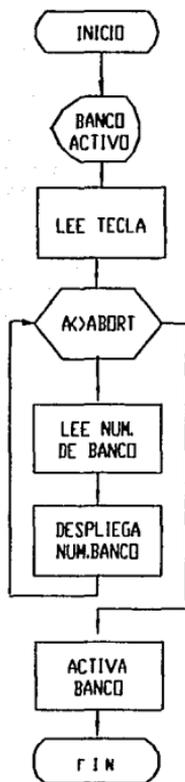


FIGURA IV-8: Proceso de activación de un banco específico de registros auxiliares.

valores del registro IP). Finalmente, para acceder un registro del banco activo, se presiona "Rx" y después se especifica el registro deseado (desde 0 hasta 7, que corresponden a R0 hasta R7) cuyos contenidos son de un byte (dos dígitos).

c) Un comando de función externa: nos proporciona funciones generales del MEV, como es la ejecución de un programa de usuario (figura IV-7), la depuración de los mismos mediante la corrida paso-a-paso, así como otras funciones con el entorno (impresiones, comunicación serial, almacenamiento de ciertos bloques de memoria, activación de uno de los tres bancos de registros de usuario como se observa en la figura IV-8).

Para la ejecución libre, es decir, dado un programa almacenado en la RAM externa del MEV, se prosigue a la corrida del mismo, dándole todo el control y quedando en manos del programador el regresar el control al monitor al finalizar su ejecución. Cuando se va a ejecutar un programa del usuario, se despliega el valor actual del PC, se solicita la dirección donde inicia el programa y se espera la señal de inicio ("GO"). La instrucción que da el mando al programa de usuario es: $JMP @A + DPTR$, donde tenemos que DPTR contiene la dirección de inicio del programa, el salto es ejecutado y a continuación todas las instrucciones almacenadas en la memoria externa.

El microcontrolador tiene cuatro bancos auxiliares de ocho registros cada uno, denominados como R0 hasta R7, para uso del MEV se utiliza uno de ellos, el superior, los otros tres quedan disponibles para el usuario. Como efecto de la inicialización el banco activo es el cero, sin embargo, si el usuario presiona la tecla marcada como "BCO", es desplegado el banco actualmente activo, que puede ser modificado por cualquiera de los otros dos bancos disponibles, proceso que es mostrado en la figura IV-8.

La corrida paso-a-paso de un programa de usuario se lleva a cabo mediante un método de interrupciones que son solicitadas al microcontrolador a través de la línea INT1 y algo de hardware adicional (mostrado en el capítulo anterior). Dentro de la depuración de un programa ejemplo se tiene la necesidad de ejecutar una sola instrucción a la vez, y poder tener el control del MEV para valorar el estado del controlador, por lo que al terminar la ejecución de una sola instrucción se despliega el contenido del acumulador, mediante la tecla "ABORT", que en este caso no ejecuta una reinicialización, sino solamente nos proporciona el "PROMPT", para acceder cualquier registro, si se presiona GO nuevamente se ejecuta la siguiente instrucción.

Las otras funciones externas consisten en almacenar o imprimir cierto bloque de información contenida en la memoria externa de RAM o EPROM, las direcciones son solicitadas después de presionar la solicitud de la función, marcadas ambas con "IMP" y "SAVE", respectivamente. EL formato de impresión es dado en caracteres ASCII para impresoras IBM PC GRAPHICS y el formato de almacenamiento es el utilizado ampliamente por los floppy's disk.

En el caso de presionar una secuencia indebida de teclas, el MEV responde con el despliegado de un mensaje de "Error", que se mantiene hasta presionar la tecla "ABORT".

IV.3 PROGRAMACION DEL SISTEMA MONITOR DEL MEV

Basados en el plan constructivo de la etapa de diseño del sistema de monitor del MEV y en los algoritmos para un desarrollo compatible con el microcontrolador, se lleva a cabo la

codificación de los diagramas y pseudocódigos, es decir, se traducen los módulos de programación a código de máquina.

El grado de abstracción más bajo lo posee el lenguaje ensamblador, que permite obtener tiempos más cortos de ejecución del programa porque se pueden aprovechar de manera óptima los registros internos del procesador, sin embargo, un programa ensamblador está confeccionado para un determinado procesador y por lo tanto no es portable. Además, se cuenta con un conjunto de programas que facilitan la programación de bajo nivel: *los ensambladores* y *los simuladores*; ambas herramientas son usadas de manera paralela y son altamente especializadas en el microcontrolador, sus características y su conjunto de instrucciones. Así el software de soporte utilizado consiste en el ASM51 como herramienta de ensamblado y el SIM8051 como simulador.

IV.3.1 PROGRAMACION DEL SISTEMA DE INICIALIZACION

Para la codificación de los diseños del sistema de inicialización se emplean técnicas de *programación estructurada*, a pesar de que la programación en lenguaje de bajo nivel no es afamado por sus facilidades para dicha técnica, se crean módulos estructurados a manera de dípolos (una sola entrada y una sola salida).

La ejecución del programa es automática y secuencial al momento de encendido, llamado comúnmente *arranque en frío*. Por la extensión del programa no se incluye completo el código fuente, sino únicamente los módulos más interesantes. Dentro del proceso de inicialización se tienen módulos de funciones específicas:

Asignación de nombres especiales a los registros apartados para el MEV (banco superior de registros)

Programación inicial de la interface 8279

Localización de las rutinas de atención a interrupciones

Programación de las interrupciones y sus prioridades

Inicialización del desplegado

En el caso del desarrollo de un prototipo completo, es necesario tener en cuenta siempre la disposición de los elementos electrónicos para hacer un acceso correcto a módulos de memoria, interfaces, etc. Las interrupciones no utilizadas por el

```

ORG 0000H
R00 EQU 001H
R01 EQU 011H
R02 EQU 021H
R03 EQU 031H
R04 EQU 041H
R05 EQU 051H
R06 EQU 061H
R07 EQU 071H
R10 EQU 081H
R11 EQU 091H
R12 EQU 0A1H
R13 EQU 0B1H
R14 EQU 0C1H
R15 EQU 0D1H
R16 EQU 0E1H
R17 EQU 0F1H
R20 EQU 101H
R21 EQU 111H
R22 EQU 121H
R23 EQU 131H
R24 EQU 141H
R25 EQU 151H
R26 EQU 161H
R27 EQU 171H
R30 EQU 181H,V0
R31 EQU 191H,V1
R32 EQU 1A1H,V2
R33 EQU 1B1H,V3
R34 EQU 1C1H,V4
R35 EQU 1D1H,V5
R36 EQU 1E1H,V6
R37 EQU 1F1H,V7
TIME0 EQU 2000H
TIME1 EQU 20081H
SERIAL EQU 20101H
PASO EQU 500H
INICIO EQU 400H
INICIAL EQU 100H
AJMP INICIAL
ORG 0003H
LJMP INT0
ORG 000BH
LJMP TIME0
ORG 0013H
LJMP PASO
ORG 001BH
LJMP TIME1
ORG 0023H
LJMP SERIAL

```

FIGURA IV-9: Listado de la primera página del programa fuente de inicialización.

prototipo que en nuestro caso son T0,T1 y R1T1 son deshabilitadas, pero se encuentran disponibles al usuario mediante conectores externos y sus rutinas de atención son ubicadas en las primeras ocho localidades de la RAM externa. Los registros apartados para el MEV son referidos como R3x donde $x=0, 1, \dots, 6$ ó 7 y que pertenecen al tercer banco de registros del usuario. Al igual que las rutinas de servicio para comunicación serial (R1T1) y para los temporizadores (T0 y T1), las variables son renombradas al inicio de la codificación del programa, donde también tenemos los saltos a las localidades físicas de las rutinas de servicio. Como ejemplo se muestra la primera página del programa fuente decodificado antes de ser ensamblado (figura IV-9).

IV.3.2 PROGRAMACION DEL SISTEMA DE AUTODIAGNOSTICO

Después de la ejecución exitosa de la inicialización del sistema, el prototipo realiza una autovaloración antes de ponerse al servicio del usuario, y es efectuado por el sistema de autodiagnóstico cuya decodificación se divide en tres módulos:

El autodiagnóstico del programa monitor

La valoración de los bancos de RAM

La revisión de las interfaces

Como resultado de los módulos se hace una valoración del prototipo, si es posible su funcionamiento o no. Para el primer caso se termina con un mensaje de "ACEPTACION", y se

continúa con el desplegado del "PROMPT". Para el segundo caso, simplemente se despliega un mensaje de "ERROR" y se pone en estado de "HALT" al procesador para que el usuario corrija el error, si es posible.

Los errores encontrados son almacenados en una estructura y los mensajes son mapeados a través de una tabla de conversión a siete segmentos.

IV.3.3 PROGRAMACION DEL SISTEMA DE COMANDOS DE LINEA

Cuando el prototipo se encuentra totalmente disponible al usuario presenta un "PROMPT", el cual no permite solicitar servicios de las diferentes funciones posibles de realizar por el MEV, mediante el sistema de comandos de línea, cuya decodificación es dividida en sus módulos funcionales:

- La habilitación del teclado
- El proceso de desplegado del "prompt"
- La lectura, revisión y definición de un comando
- La ejecución de una función interna
- La ejecución de una función externa
- La Invalidez de un comando y el desplegado de "ERROR"

IV.3.4 ENSAMBLADO DE LOS PROGRAMAS DEL MONITOR

El *lenguaje ensamblador* está orientado hacia el procesador específico utilizado. Cada línea de un programa contiene una instrucción de máquina, escrita en mnemónico. Los ensambladores son traductores de mnemónicos a código de máquina, lógicamente en valores hexadecimales.

Todos los programas son ensamblados mediante el programa ASM51 para PC IBM compatible. Uno a uno los programas realizados en los puntos anteriores, son procesados por el programa ensamblador. Una vez obtenidos los códigos de máquina sin errores, se procede al almacenamiento en una memoria no volátil.

```

0000          ORG 0000H
0000 R00 EQU 00H
0001 R01 EQU 01H
0002 R02 EQU 02H
0003 R03 EQU 03H
0004 R04 EQU 04H
0005 R05 EQU 05H
0006 R06 EQU 06H
0007 R07 EQU 07H
0008 R10 EQU 08H
0009 R11 EQU 09H
000A R12 EQU 0AH
000B R13 EQU 0BH
000C R14 EQU 0CH
000D R15 EQU 0DH
000E R16 EQU 0EH
000F R17 EQU 0FH
0010 R20 EQU 10H
0011 R21 EQU 11H
0012 R22 EQU 12H
0013 R23 EQU 13H
0014 R24 EQU 14H
0015 R25 EQU 15H
0016 R26 EQU 16H
0017 R27 EQU 17H
0018 R30 EQU 18H ; display V0
0019 R31 EQU 19H ; display V1
001A R32 EQU 1AH ; display V2
001B R33 EQU 1BH ; display V3
001C R34 EQU 1CH ; display V4
001D R35 EQU 1DH ; display V5
001E R36 EQU 1EH ; display V6
001F R37 EQU 1FH ; display V7
2000 TIME0 EQU 2000H
3008 TIME1 EQU 3008H
2010 SERIAL EQU 2010H
0500 PASO EQU 500H
0400 INT00 EQU 400H
0100 INICIAL EQU 100H
0000 2100 AJMP INICIAL
0003          ORG 0003H
0003 020400 LJMP INT0
000B          ORG 000BH
000B 022000 LJMP TIME0
0013          ORG 0013H
0013 020500 LJMP PASO
001B          ORG 001BH
001B 022008 LJMP TIME1
0023          ORG 0023H
0023 022010 LJMP SERIAL 0000
END

```

FIGURA IV-10: Listado de la primera página del sistema de inicialización ya ensamblado.

IV.3.4.1 CARACTERISTICAS DE LOS PROGRAMAS A ENSAMBLAR

El software de apoyo utilizado para el ensamblado de todos los programas del sistema monitor requiere de ciertas normas que

deben cumplir los programas fuente cuyas versiones deben ser *ASM*, para obtener como resultado dos programas cuyas versiones son *PRN* y *HEX*, el primero contiene el código de máquina del programa fuente y al mismo programa fuente con la tabla de símbolos y la documentación; el segundo contiene únicamente el código de máquina organizado en bloques de 16 bytes útil para el almacenamiento a memoria no volátil -ROM- o alguna variedad de ella.

El programa fuente a ensamblar debe cumplir con las siguientes normas:

- Contener la etiqueta de END y en el siguiente renglón la marca de fin de archivo "EOF"
- Las definiciones de localidades físicas "ORG" deben ser en mayúsculas
- La declaración de variables mediante la etiqueta EQU son en la primera columna
- Un dato inmediato es identificado por un # y un cero (consultar el Apéndice B)
- Las instrucciones no deben ir colocadas en la primera columna
- Las etiquetas de saltos y subrutinas deben estar en la primera columna

Un ejemplo del listado obtenido de un programa ensamblado se muestra en la figura IV-10.

IV.4 PRUEBAS DE FUNCIONAMIENTO

Las pruebas del software no pueden efectuarse al finalizar el diseño, desarrollo e implantación de los programas, sino que debe terminarse cada paso del desarrollo con adecuadas pruebas, para eliminar errores que no sólo consumen tiempo sino que resultan caros y frustrantes. Se sabe que por experiencia, hasta un 50% de los costos de desarrollo son absorbidos por las fases de pruebas.

Las ventajas de realizar pruebas a cada paso del desarrollo del software son:

- Los errores se reconocen a tiempo.

- Se les puede eliminar con un gasto menor.
- Evitamos la propagación de los errores a otros niveles.
- Se obtiene una relativa seguridad.

Los niveles de pruebas pueden ser tan refinados como se necesiten, algunos de ellos son los siguientes:

Fase de Desarrollo	Posibilidad de Prueba	Medio de Prueba
Diseño	Tablas de decisión, Diagramas de flujos	Prueba formal Revisión
Implantación	Lógica del desarrollo Estructura de datos	Traductor, Ensamblador, Depurador, Simulador,
Integración	Composición de Hw y Sw (prototipo)	Emuladores, Adaptadores de Emulación y prueba(ETA).

La modularización facilita la etapa de pruebas, porque el programa se compone de unidades menores. Las pruebas comprenden tanto los datos como la lógica de control.

IV.4.1 PLAN DE PRUEBAS DEL SOFTWARE

Para obtener la seguridad necesaria y suficiente de un sistema de software debe elaborarse un plan de pruebas que contemple los puntos siguientes:

- Lo que deben de mostrar las pruebas.
- Los resultados esperados.
- Hasta cuando queda demostrada una relativa funcionalidad.

Se deben analizar todas las posibilidades de ramificación, de un programa, pero por el tamaño la cantidad de caminos puede llegar a ser muy grande; al analizar las posibles bifurcaciones, determinamos la medida de pruebas para la lógica de programación.

Para las *estructuras de datos*, también se fijan los alcances de los valores, las pruebas consisten en observar cómo se comportan los módulos a los valores autorizados, después, con valores extremos y fuera de rango, así como sus combinaciones.

La *revisión* es un diálogo entre el diseñador y un equipo especializado para analizar al producto parcialmente y determinar sus puntos débiles y fallas. Las revisiones son reuniones, entonces, deberá elegirse un tema especial como un módulo, las interfaces o parte del código fuente. Las soluciones y respuestas son dadas por el diseñador.

El *procedimiento de verificación* debe demostrar la corrección de los programas, a través del código mnemónico obtenido del código objeto terminado y que se encuentra en memoria, mediante el desensamblado y comparado con el original.

La *depuración del software* nos auxilia en la prueba paso a paso de los programas. Podemos reorganizar la ejecución de un conjunto de instrucciones y luego detenerlo en un punto predefinido, para observar el estado del sistema, los valores de los registros, de las posiciones de memoria y las señales de E/S.

Los *simuladores* son programas que reproducen las leyes de un proceso en una computadora y permiten considerar a la instalación como un modelo. Los simuladores pueden reemplazar a otros módulos no disponibles, a un sistema objeto, proveer un sistema de control para ensayos, etc.

Los *emuladores* son unidades funcionales, realizados con módulos del programa que reproducen las características de una computadora tipo A en otra tipo B, para poder correr programas de A y B, donde los datos de una son aceptados por la otra con los mismos resultados. Los emuladores consisten en Sw y Hw que se interconectan al sistema de desarrollo y al Hw de aplicación. El procesador de emulación asume las funciones del procesador del Hw de aplicación. Toda la información que pasa por el sistema de buses es registrada en una memoria de trazo y es posible interrumpir cualquier operación en cualquier instante.

IV.4.2 ESTRATEGIAS DE LAS PRUEBAS

Existen dos métodos principales para llevar a cabo el plan de pruebas: el método de arriba-abajo y el método de abajo-arriba.

METODO DE ARRIBA-ABAJO (TOP-DOWN): El conjunto de pruebas se inicia en el módulo principal. Para los módulos inexistentes solamente se suministraran los valores pruebas o simplemente un retorno. Se van incorporando uno a uno los restantes módulos ensayados.

METODO DE ABAJO-ARRIBA (BOTTON-UP): El inicio de las pruebas se hace con el módulo de la posición más baja -el más cercano a la máquina básica-. Al módulo a probar se le suministran los valores prueba a través de un marco -programa preparado especialmente-. Se va incorporando en la prueba los módulos de encima, uno a uno.

CAPITULO V

APLICACIONES Y CONCLUSIONES

La importancia de un prototipo ingenieril consiste en la magnitud del campo de aplicaciones en la solución de diferentes problemas de la vida humana y su entorno con el fin de lograr los satisfactores y servicios necesarios.

Las *microcomputadoras* han penetrado en todos los campos de la técnica, por una parte se trata en muchos casos de reemplazar componentes electromecánicos o de simplemente hacer nuevos procesos que ahora ya son factibles de controlar por medio de la *electrónica de microprocesadores*, como en la técnica de mediciones, control y regulación que son los campos más amplios para las computadoras.

Un sistema controlado por una microcomputadoras consta de las partes funcionales que se observan en la *figura V-1* y es llamado como *sistema de aplicación*.

Con el apoyo de diferentes herramientas es posible llevar un programa desde el cuaderno de apuntes del diseñador hasta la ROM y lograr un objetivo, el prototipo construido es una de esas

SISTEMA CONTROLADO POR UNA MICROCOMPUTADORA

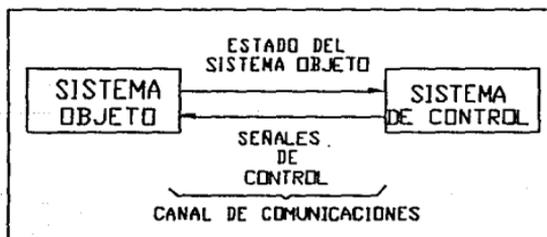


FIGURA V-1: Los sistemas controlados por una microcomputadora se interconectan a ella por medio de un canal de comunicaciones.

herramientas, ya que para controlar un sistema con un programa de hasta 64 kbytes basta con utilizar una microcomputadora experimental tal como el MEV (figura V-2).

Como ejemplos de aplicaciones de nuestra herramienta, se presenta un conjunto de prácticas sobre diferentes campos de la ciencia aplicada.

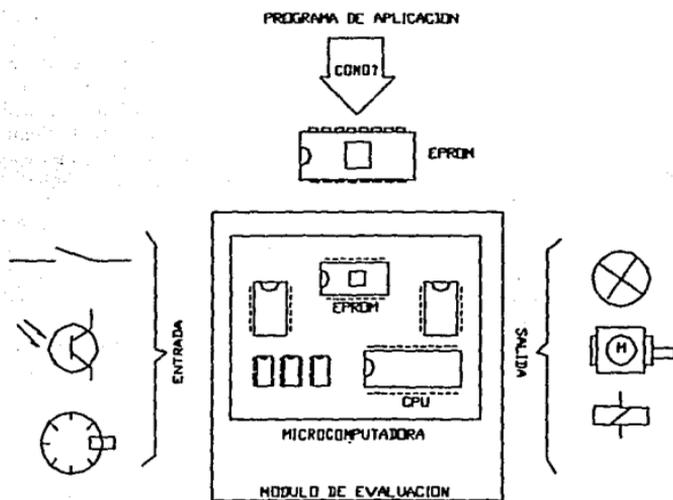


FIGURA V-2: El MEV como herramienta de control de un sistema mediante un programa de aplicación almacenado en una EPROM.

V.1 SIMULACION DEL MOVIMIENTO OSCILATORIO

Los *movimientos periódicos* ocurren frecuentemente en la naturaleza, algunos pueden ser descritos en función de una sola coordenada de distancia como el movimiento ascendente y descendente, también llamado *movimiento oscilatorio* (Vgr. el latido del corazón y el balanceo de un péndulo). Un *movimiento periódico* es aquel que después de transcurrir un cierto tiempo llamado *período del movimiento*, el sistema vuelve a su posición y velocidad inicial, después de lo cual el movimiento se repite una y otra vez.

V.1.1 CARACTERISTICAS DE UN PENDULO

Un *péndulo simple*, ilustrado en la figura V-3, es un sistema de movimiento armónico simple que consiste en una pequeña masa m suspendida de una cuerda de longitud l y de masa

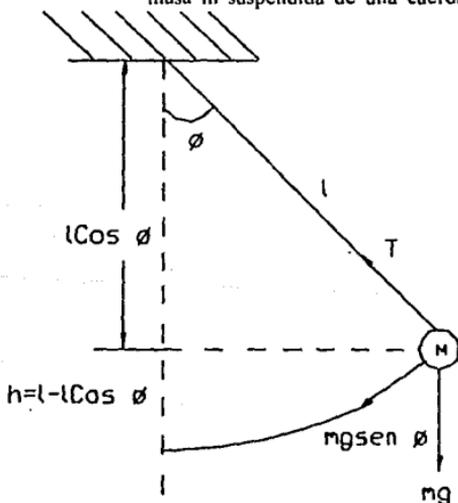
despreciable fija a un soporte superior estacionario.

El comportamiento de la masa suspendida es esencialmente el de una masa puntual.

La cuerda forma un ángulo con la posición de equilibrio vertical.

Las únicas fuerzas de acción sobre la masa son el peso (mg) que son indicadas por sus componentes paralela y normal a la cuerda y la tensión T .

Como la línea de acción de la tensión T pasa por el eje O , no da lugar a momento alguno con



EJE O

FIGURA V-3: Diagrama de un péndulo simple.

respecto a ese punto. Solamente la componente $mg\text{Sen } \phi$ genera momento y es paralela a la trayectoria del péndulo, cuya aceleración angular esta dada por la fórmula de la figura V-4 (a). Cuando el desplazamiento angular o es pequeño comparado con

$$a) \quad \frac{d^2 \phi}{dt^2} = \frac{-g \text{Sen} \phi}{l}$$

la unidad, el radián = 57.3° , entonces:

$$\text{Sen } x = x \text{ si } x < 1 \text{ radián}$$

$$b) \quad \frac{d^2 \phi}{dt^2} = -\omega^2 \phi$$

La fórmula quedaría como se muestra en la figura V-4 (b).

Finalmente si aplicamos la fórmula resultante de la figura V-4

$$c) \quad T = \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{l}{g}}$$

(c) tenemos que cuando $T=2$ segundos, a nivel de mar, la longitud de un péndulo de reloj sería de $l=0.993$ m.

$$l = \frac{gT^2}{4\pi^2}$$

FIGURA V-4: Fórmulas del movimiento de un péndulo simple.

V.1.2 PROGRAMA DE SIMULACION DEL MOVIMIENTO DE UN PENDULO

El programa de simulación es un proceso infinito donde vemos desplegado una masa representada en siete segmentos que nos muestra las características visuales en velocidad, tiempo e inercia del movimiento oscilatorio de un péndulo colgado de un soporte estacionario mediante una cuerda de longitud variable l , dado que la masa es despreciable y puntual no es leída, ni utilizada en los cálculos.

El diagrama de procesos del programa de simulación se muestra en la figura V-5, y fundamentalmente consiste en lo siguiente:

- El movimiento es periódico e infinito, por lo que el proceso no tiene final.
- Al inicio se lee la longitud de la cuerda que conforma al péndulo.
- Se hace un borrado de todos los despliegues denominados como V0 hasta V7.

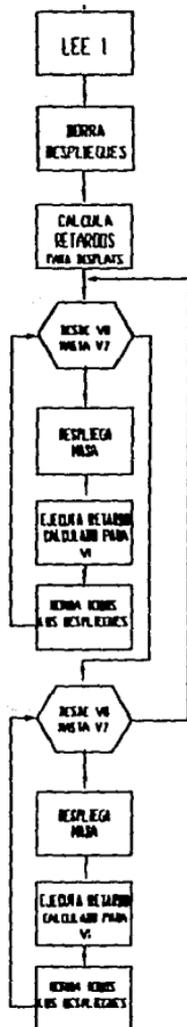


FIGURA V-5: Diagrama de procesos del programa de simulación del péndulo simple.

- A partir de las fórmulas anteriores y con el valor de l se calculan los retardos para cada despliegue.

Mediante un proceso repetitivo hasta que se despliega la masa en los ocho despliegues y que representa medio período de oscilación, se despliega con su retardo calculado y es borrado.

El proceso se repite en la segunda mitad del período, con las mismas características que la primera mitad.

Todo el programa se repite a partir de los desplegados.

La condición para aplicar las fórmula de la figura V-4 nos dicen que el ángulo de la cuerda con respecto a la vertical, no debe ser mayor a un radián (57.3°), que es el caso representativo de los despliegues. La ruta seguida por el péndulo de longitud l , es muestreada en ocho diferentes etapas, cuyas aceleraciones dependen de la longitud y del ángulo con respecto a la vertical, todo el proceso se encuentra dentro del cálculo de los retardos para cada despliegue.

V.2 ADQUISICION DE DATOS VIA RS-232C DE UNA COMPUTADORA PERSONAL

Una de las tareas más tediosas en el manejo del MEV como herramienta, consiste en introducir el código de máquina de un programa de aplicación, así como también lo es el revisado del código en la RAM antes de ejecutarlo y tener la certeza de que es el mismo del programa ensamblado, aún más cuando los programas tienen una extensión mayor o cuando se necesitan tablas de datos que necesitamos variar para observar el diferente comportamiento del sistema a prueba. Todas estas labores son demasiado mecánicas y pueden ser hechas completamente por otro medio sin regulación de nuestra parte como lo haría una computadora personal.

V.2.1 LA COMUNICACION SERIAL

Para la transferencia de datos se utilizan las interfaces y pueden ser de dos tipos: *serial* o *paralela*; la *interfaz paralela* acepta de manera simultánea 8 bits y los transfiere de la misma forma, por lo que necesitamos 8 conductores para datos y otro tanto para control que lo hace muy costoso en separaciones de varios metros, por lo tanto para grandes distancias es recomendable que el transmisor convierta los datos en un flujo serie y que el receptor invierta el proceso, reduciéndose el número de conductores en forma considerable.

En la *comunicación serial*, el emisor envía los bits uno tras otro por un sólo conductor hacia el receptor. La distancia entre el emisor y el receptor casi no es significativo actualmente. Se tiene dos modalidades de comunicación serial: la *semidúplex* -cuando en un sólo instante solo se puede transmitir o sólo recibir con un conductor- y la *dúplex* -que necesita una segunda vía de transmisión para poder transmitir y recibir de forma simultánea-. Las estipulaciones exactas que determinan las formalidades de la transmisión se encuentran en el *protocolo de comunicación* y son *la velocidad de transmisión, el formato externo de los datos, la sincronización y la seguridad de los datos*. La transmisión puede ser *asíncrona* (si se tiene una señal de inicio y de finalización) o *síncrona* (si se transmiten señales preestablecida de sincronía

como las fases por bloques). Además para la fidedigna recepción de la información, se puede agregar al dato un *bit de paridad*, es decir, complementar la cantidad de bits 1 a un número par o impar que al recibirlo se verifica automáticamente la palabra según la paridad establecida. Nuestro interés se basa en la *transmisión asíncrona* que necesita un generador de pulsos de reloj en el transmisor y en el receptor, cada uno a la misma frecuencia, para generar la tasa de transmisión de datos, dada por:

$$\text{Tasa de transmisión de datos} = \frac{\text{Frec. del reloj}}{\text{Factor de relación}}$$

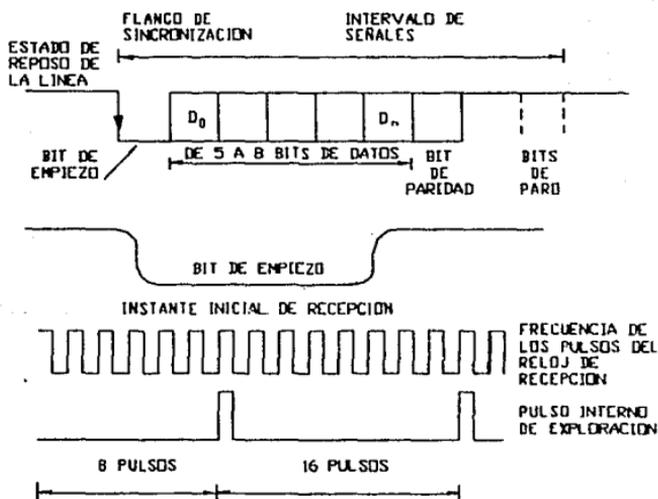


FIGURA V-6: Principio de sincronización en la transmisión serial asíncrona de datos.

La frecuencia del reloj es múltiplo de la tasa de transmisión en un factor de relación de 16 ó 64. El *principio de sincronización* (figura V-6) nos muestra como se reconoce la secuencia de bits enviados. La línea se encuentra inicialmente en un nivel lógico 1, cuando esta en reposo, al cambiar de nivel se inicia la transmisión ya que el bit de inicio es detectado, luego siguen de 5 a 8 bits de datos -según el protocolo establecido-, el bit de paridad -si existe- y los bits de paro.

Para garantizar que la exploración del bit siguiente se encuentra en "el medio del bit", el receptor cuenta 8 pulsos a partir de detectar el flanco descendente del bit inicial y luego explora el estado de la línea mediante un pulso interno de exploración, a continuación se explora cada 16 pulsos, por ello el factor de relación. Si se utiliza un factor de 1 la transmisión es *isosincrónica* y los relojes deberán estar absolutamente sincronizados.

V.2.2 CARACTERÍSTICAS DE LA INTERFACE SERIAL DE UNA COMPUTADORA PERSONAL

Una computadora personal tiene adaptadores de comunicación serial programables denominados como COM1 hasta COM4, contruidos basados en la norma internacional RS-232C. Para programar las características del protocolo de comunicación serial se pueden utilizar uno de los dos medios: el sistema operativo o un programa en cualquier lenguaje. La comunicación es asíncrona y con una gran variedad de protocolos que nos permiten tener muchas características de transmisión. La forma de programación desde el sistema operativo se basa en el comando MODE, cuyo formato es:

mode COMn:b,p,d,s,r

Todas las modalidades posibles se obtiene al sustituir las opciones por sus valores permitidos que aparecen en la figura V-7.

V.2.3 HARDWARE ADICIONAL PARA EL ESTANDAR RS232C

Según las características eléctricas del canal de comunicación son los aditamentos necesarios para poder adentrarnos a la modalidades del estandar de transmisión vía RS232C, los cuales

OPCION	FUNCION	POSIBLE VALORES	VALOR DE DEFAULT
n	Número del adaptador serial	1, 2, 3 o 4	1
b	Tasa de Transmisión	110, 150, 300, 600, 1200, 2400, 4800, 9600 o 19200	
p	Paridad	N (ninguna) O (par) E (impar) M (marca) S (espacio)	E
d	Núm. de bits de datos	5, 6, 7 o 8	7
s	Núm. de bits de paro	1, 1.5 o 2	Si b=110 entonces s=2 sino s=1
r	Reintentar	NONE (ninguna) E (reintentar)	

FIGURA V-7: Conjunto de opciones de comunicación asíncrona de los puertos seriales de un computadora personal.

nos indican que los voltajes para un 1 lógico es de -12 voltios y para un 0 es de 12 voltios.

Para adaptar nuestras señales TTL al canal de transmisión se utilizan convertidores como los CI's 1488 y 1489, ya que como nuestra comunicación será dúplex donde podemos transmitir y recibir de forma simultánea a través de dos conductores (figura V-8).

Dada la necesidad de obtener los voltajes de -12 y +12 voltios es necesario incluir una fuente de voltaje regulado.

V.2.4 PROGRAMA DE ADQUISICION DE DATOS VIA RS232C

El programa de adquisición de datos tiene dos funciones principales: cargar el código de máquina de un programa ensamblado y salvar el código existente en la RAM del MEV; para

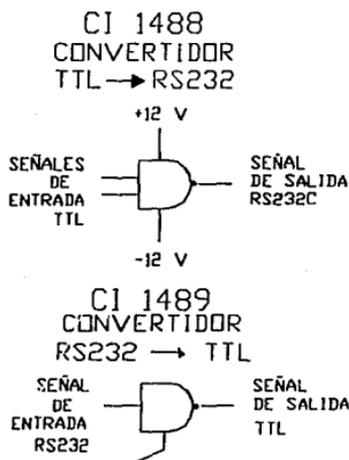


FIGURA V-8: Diagrama de los convertidores TTL a RS232C y viceversa.

ambas se tiene como fuente o como destino una computadora personal IBM compatible, así como se asignan a cierta teclas del MEV marcadas con "CARG" y "SALV", respectivamente. El protocolo de comunicación elegido es establecido en los valores asignados a los diferentes registro de microcontrolador, así con el registro SCON elegimos la cantidad de datos (8 bits), el bit de paridad, el número de bits de paro y el bit de inicio; con los registros TH1, TMOD y TCON se genera la tasa de transmisión tomando en cuenta el reloj interno del sistema. El diagrama de procesos que tiene el programa es mostrado en la figura V-9.

Al inicio se habilita el teclado para leer la función deseada, en todas las lectura de comandos, presionar la tecla "ABORT" nos hará reiniciar a un nuevo comando. Una vez definida la tarea a ejecutar, se programa el puerto, con todas las características del protocolo elegido.

Se lee la dirección inicial donde se cargarán los datos leídos o de donde se obtendrán los datos a enviar para su almacenamiento en discos flexibles y la cantidad de la información en bytes.

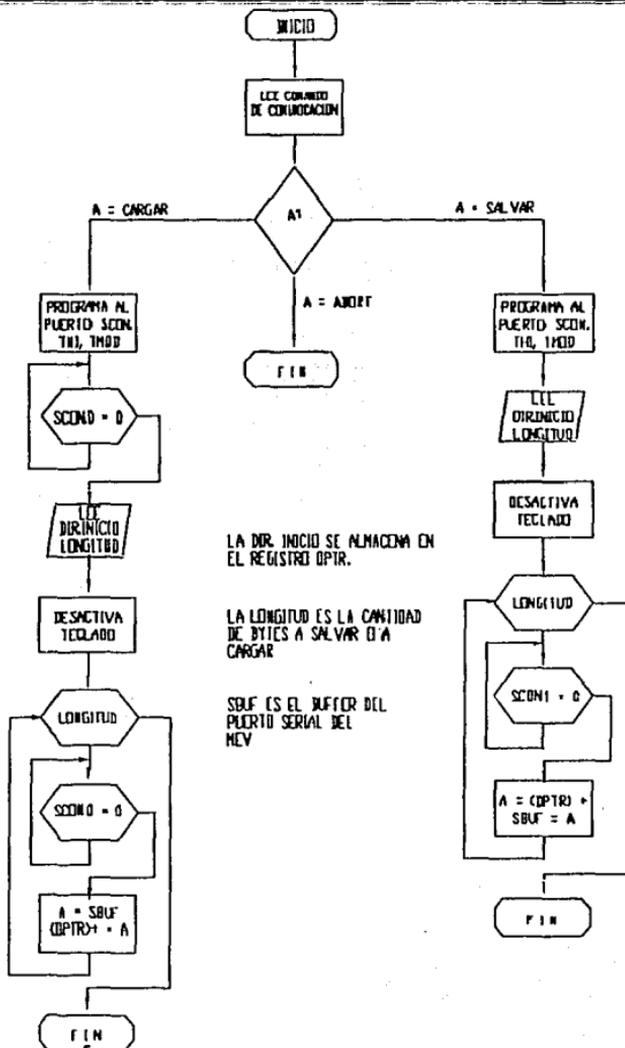


FIGURA V-9: Diagrama de procesos del programa de comunicación serial.

La dirección inicial es almacenada en el registro DPTR y la longitud en un registro de usuario.

A partir de entonces se espera la ocurrencia de una interrupción del puerto serial, en el caso de la carga, cuando ocurre, el dato es leído desde el SBUF (buffer del puerto serial) y es almacenado en la dirección del DPTR a través del acumulador. El proceso se repite hasta alcanzar la cantidad de bytes especificado anteriormente.

Para el caso de la transmisión, el dato se obtiene de la localidad apuntada por el registro DPTR y mediante el acumulador se envía al SBUF, el proceso se repite hasta hacer enviado todo el bloque marcado.

V.3 MONITOREO DE UNA VARIABLE FISICA

Nuestro entorno está caracterizado por *procesos analógicos* y en su gran mayoría, no son eléctricos. Por medio de *sensores* podemos captar magnitudes como temperaturas, presiones, flujos, etc., y convertirlas en *señales eléctricas analógicas*. Los sensores están formados por tres elementos: un **transductor** -transforma la magnitud a medir en una magnitud medible-, un **convertidor** -convierte una magnitud no eléctrica en una magnitud eléctrica evaluable- y un **amplificador** que conforma la señal, la estabiliza y la compensa, como lo vemos en la figura V-10.

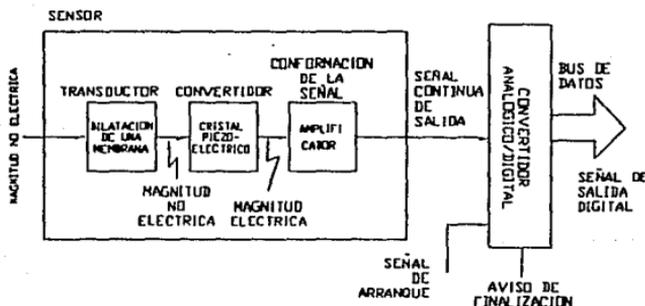


FIGURA V-10: Elementos funcionales de un sensor.

Finalmente, nuestra señal eléctrica analógica de salida requiere ser convertida en una *señal digital* para poder ser procesada por un sistema basado en un microprocesador o un microcontrolador, como es nuestro objetivo, el último paso es ejecutado por los *convertidores analógico/digitales (ADC)*.

V.3.1 CARACTERISTICAS DE LA CONVERSION A/D

La conversión de una señal analógica en digital no es continua, sino que requiere un tiempo de conversión, es decir, a determinados intervalos se toma una muestra de la señal de entrada y es convertida a un código binario. La máxima frecuencia de la señal analógica nos determina la *frecuencia de muestreo*, que según el *teorema de muestreo* se deben tomar por lo menos dos muestras en un período de una oscilación sinusoidal, además cuanto mayor es nuestra cantidad de muestras

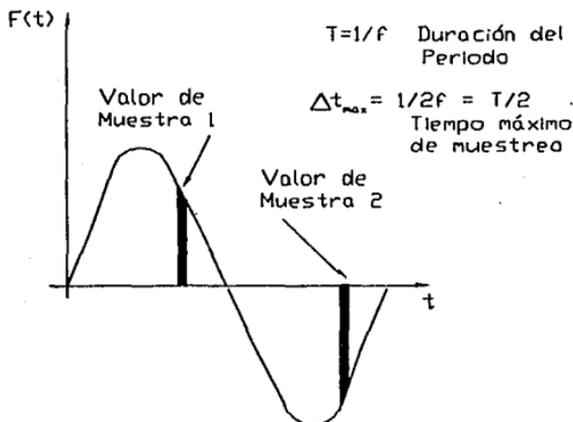


FIGURA V-11: Teorema de muestreo y el tiempo máximo de muestreo de una señal analógica.

en un período, mejor será la imagen digital resultante, empero el número de muestras no puede ser muy grande ya que el sistema digital necesita cierto tiempo para procesar la muestra tomada (figura V-11).

V.3.2 FRECUENCIA MAXIMA DE MUESTREO

Si un microcontrolador tiene un tiempo de ciclo de m microsegundos y para procesar una muestra ejecuta una rutina de atención de 100 instrucciones, la máxima frecuencia posible de la señal analógica esta dada por:

$$t = 100(nn) = XX \text{ microsegundos}$$

$$T = 2 (t) = 2YY' \text{ microsegundos, que es el periodo}$$

$$F = 1/T = 1/(2YY') \text{ microsegundos, que es la frecuencia máxima}$$

Sin embargo, los convertidores tienen un tiempo necesario de conversión, que no fue tomado en cuenta, que es de alrededor de 30 microsegundos.

V.3.4 PRINCIPIO GENERAL DE CONVERSION ANALOGICA/DIGITAL

La conversión analógica/digital se basa en divisores de tensión y de comparadores (amplificadores operacionales sin realimentación) cuya salida es uno si la entrada es igual o mayor que el voltaje de referencia, en otro caso es cero.

La resolución digital de la señal analógica (alcance) la define la tensión de referencia.

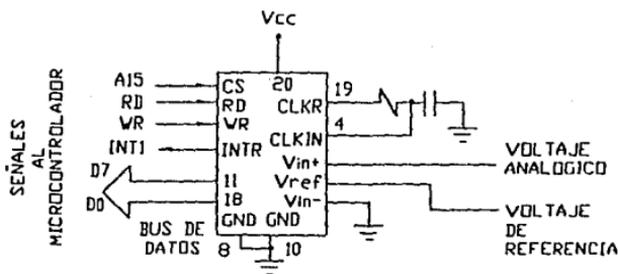


FIGURA V-12: Diagrama de conexión del ADC8003 al MEV.

V.3.5 CONEXION DEL ADC803 AL MEV

Mediante la conexión del convertidor A/D con el microcontrolador es posible procesar, según la rutina de atención, una señal analógica, como se muestra en la figura V-12.

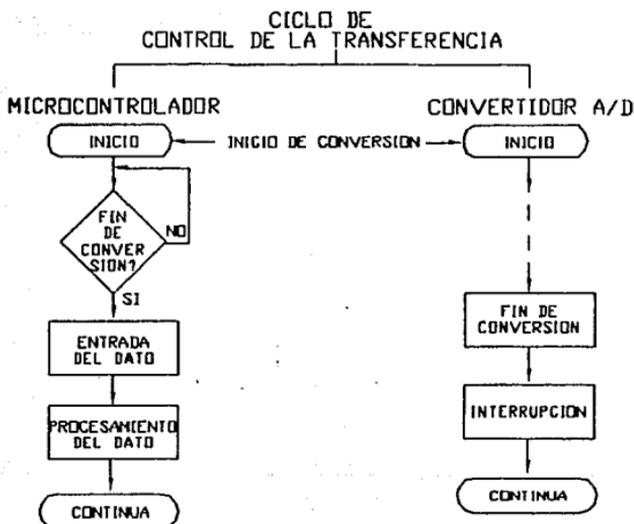


FIGURA V-13: Diagrama de procesos del programa de monitoreo de una variable física.

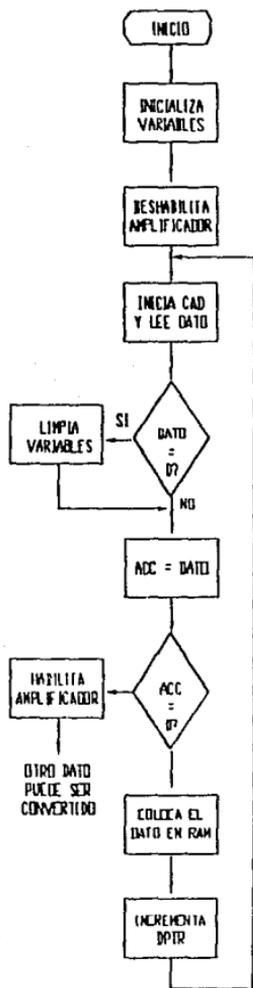


FIGURA V-14: Rutina de auto-cero para la corrección de datos leídos.

Con una instrucción de salida se habilita la selección del ADC, cuando el convertidor termina su tarea con la primera muestra le avisa al microcontrolador, a través de una petición de atención se hace una llamada a un subprograma que toma la muestra colocada en el bus de datos, este valor de medición convertido es procesado mientras se bloquea la entrada de interrupciones al microcontrolador.

V.3.6 HARDWARE ADICIONAL PARA EL MONITOREO DE LA VARIABLE FISICA

Para poder monitorear una variable física y utilizar al MEV para su análisis y procesamiento es necesario adherir nuevos componentes como el convertidor analógico/digital (ADC8003), una etapa de amplificación (un amplificador y un filtro paso-bajas) y el transductor.

V.3.7 PROGRAMACION PARA EL PROCESAMIENTO DE LA VARIABLE FISICA

El programa de monitoreo se componen de dos rutinas (figura V-13), cuyas funciones son:

- Control de transferencia de información
- Procesamiento de la variable

El control de transferencia de información, programa al puerto de lectura (P1), así como recibe el dato desde el ADC que solicita atención al microcontrolador por medio de una interrupción cuando ya tiene un dato convertido.

El procesamiento de la variable nos permite obtener el dato y procesarlo, para lo cual utilizamos una rutina de atención. Dentro del proceso se necesita llevar al ADC a un autocero a través de la rutina mostrada en la figura V-14.

V.4 CONCLUSIONES

Un sistema de desarrollo como el MEV es una herramienta muy útil para contruir sistemas de aplicación (figura V-13) lo que consiste en cuatro fases:

- En primer lugar y basándonos en un MEV, donde almacenamos un programa de aplicación que es depurado mediante el auxilio de un programa monitor y es probado hasta que finalmente obtenemos resultados.

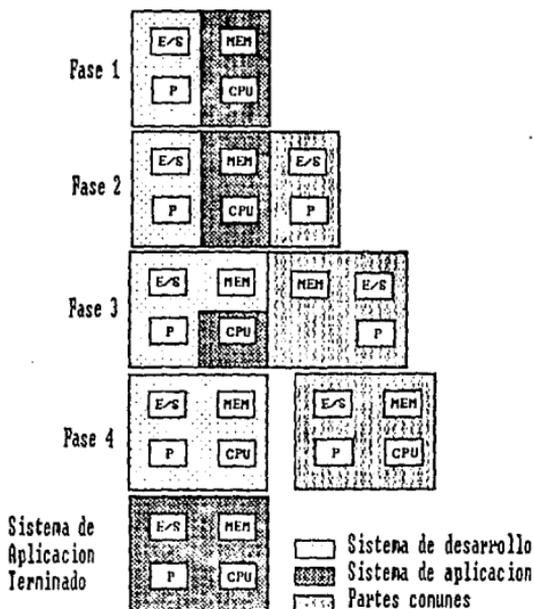


FIGURA V-15: Construcción de un sistema de aplicación utilizando al MEV como sistema de desarrollo.

- En la segunda fase, se dispone de algo de hardware, muchas veces incompleto y estrictamente lo necesario para nuestra aplicación. Mediante la emulación se comunican los sistemas: el MEV y el de aplicación. El programa todavía está en el MEV y puede ser corregido y adaptado.
- En la tercera fase se tienen ya las pruebas satisfactorias y el software terminal que es almacenado en una EPROM y se incorpora al sistema de aplicación, pero el microcontrolador del MEV sigue comunicado al sistema de desarrollo, como arbitro de cualquier conflicto, así como para proporcionar las señales de seguimiento (TRACE)
- Finalmente, en la cuarta etapa, se incorpora el microcontrolador del sistema de aplicación y se independiza del MEV, obteniéndose un nuevo producto o uno muy idéntico a la arquitectura del sistema de desarrollo.

En la aplicación de técnicas de construcción de prototipos basados en microprocesadores, se requiere:

- Una adecuada formación o perfeccionamiento personal.
- Una estrecha cooperación entre los fabricantes, los diseñadores de hardware y de software.
- Una dirección flexible que fomente innovaciones.
- Una cuidadosa planificación del producto y de las estrategias de mercado.

APENDICE A

DESCRIPCION DE PINES DEL MICROCONTROLADOR 8031

La descripción siguiente es compatible con todos los miembros de la familia MCS-51, como son el 8051, el 8052, el 80C31, el 8032, etc.

- Vcc (Pin núm. 40)

Alimentación de voltaje positivo. Valor típico: 5 volts.

- Vss(Pin núm. 20)

Tierra potencial de referencia. Valor típico: 0 volts.

- Puerto P0 (Pines 39 al 32)

Puerto bidireccional de 8 bits de drenaje abierto con niveles TTL, por lo tanto tiene 1's al estar sin conexión alguna. Contiene el byte bajo (LSB) del bus de direcciones y al bus de datos durante el acceso a memoria externa. Por lo tanto: $P0.x = Dx = ADx$.

- Puerto P1 (Pines 1 al 8)

Es un puerto bidireccional de 8 bits con resistencias internas conectada a Vcc, con niveles TTL, pueden leerse 1's mediante su resistencia interna y usarse como entradas. Sus pines no tienen funciones alternas y podemos hacer referencia a sus pines como P1.x.

- Puerto P2 (Pines 21 al 28)

Es un puerto de 8 bits con resistencias internas conectadas a Vcc, con niveles TTL y contiene el byte alto (MSB) del bus de direcciones durante el acceso a la memoria externa con 16 bits de dirección. Por lo tanto: $P2.x = A_{D(x+8)}$.

- Puerto P3 (Pines 10 al 17)

Es un puerto bidireccional de 8 bits con resistencias internas conectadas a Vcc, con funciones alternas listadas a continuación:

- P3.0 RXD Recepción serial
- P3.1 TXD Transmisión serial
- P3.2 INT0 Interrupción Externa 0
- P3.3 INT1 Interrupción Externa 1
- P3.4 T0 Temporizador 0
- P3.5 T1 Temporizador 1
- P3.6 WR Habilitador para escritura de memoria externa de datos.
- P3.7 RD Habilitador para lectura de memoria externa de datos.

- RST (Pin 9)

Entrada de la señal de reinicialización "RESET". Un nivel alto con una duración de 2 ciclos de máquina mientras el oscilador funciona reinicializa al dispositivo.

- ALE/PROG (Pin 30)

Pulso de salida, para habilitar al latch y almacenar el byte bajo de las direcciones durante un acceso de memoria externa de programas. Tiene una frecuencia de 1/6 del oscilador y es emitido en forma constante, aun cuando no se usa memoria externa de programas. También es usado como entrada del pulso de programación para dispositivos con memoria EPROM interna.

- PSEN (Pin 29)

Habilita la memoria externa de programas para lectura. Es habilitado 2 veces en cada ciclo de máquina cuando el dispositivo está ejecutando un programa desde memoria externa. La señal PSEN no es activada cuando el dispositivo ejecuta un programa en memoria interna.

- **EA/Vpp (Pin 31)**

Define el lugar físico del programa de inicialización, si EA=0 el programa es externo; si EA=1 el registro PC ejecuta el programa interno hasta exceder de la dirección 0FFFh para el 8051 ó 1FFFh en el 8052. Para dispositivos con EPROM, Vpp es la entrada del voltaje de programación.

- **XTAL1 (Pin 19)**

Entrada del circuito oscilador externo al oscilador-amplificador interno.

- **XTAL2 (Pin 18)**

Salida del oscilador-amplificador interno.

APENDICE B

SIMBOLOS CARACTERISTICOS DEL CONJUNTO DE INSTRUCCIONES DE LA FAM. MCS-51

SIMBOLO	DESCRIPCION
#	Valor Inmediato
@	Dirección indirecta
PC	Contador de Programa
/	Complemento del valor
23H	Valor hexadecimal 23
10B	Valor binario (10)B = 2H
rel	Localidad relativa
< dato >	Operando inmediato de 8 bits
< dato16 >	Operando inmediato de 16 bits
directo	Localidad de RAM de propósito general
Rn	Registro n del banco habilitado de registros de usuario (n {0,7})
ACC	Acumulador
B	Registro B o Acumulador auxiliar

APENDICE C

INTERFACE ESTANDAR RS-232C (CCITT V.24/ISO 2110)

Es una de las interconexiones más difundida para transmisión de datos seriales, llamada RS-232C según la nomenclatura norteamericana o también referida como CCITT V.24 en la nomenclatura internacional.

Consiste en 25 conexiones de intercambio con una función cada una, contenidos en una clavija de corte trapezoidal, para evitar el acoplamiento erróneo, asegurándose a través de dos tornillos al puerto de salida del dispositivo.

Permite una velocidad máxima de 20 kbps a una distancia máxima de 15 metros.

La descripción de las funciones de cada pin se puede observar en la figura C-1.

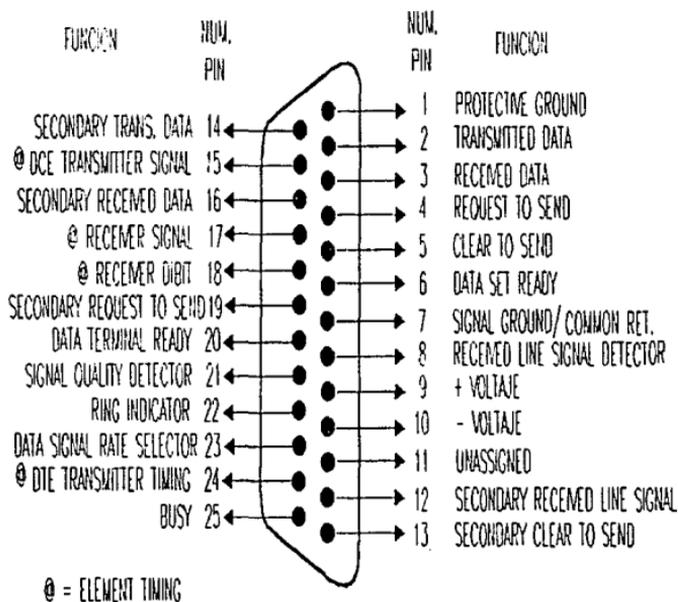


FIGURA C-1

BIBLIOGRAFIA

Autodesk Inc., *The autocad drafting package: Reference Manual*, U.S.A. 1986.

Hitachi, *IC Memories Data Book*, San Diego, CA., U. S. A., sept. 1985.

Holtz Matthew, *Mastering Ventura*, Macrobit Editores, S. A. de C. V., 2a. Edición, U.S.A. 1990.

Intel Corp., *Embedded controller handbook*, U.S.A., 1988.

Intel Corp., *Microprocessor and peripheral handbook*, U.S.A., 1988.

National Semiconductor Corporation, *CMOS Databook*, U. S. A. 1984.

National Semiconductor Corporation, *Linear Databook*, U. S. A., 1980.

Martínez Juan B., *Apuntes de microprocesadores y sistemas digitales*, D.I.M.E., Fac. de Ingeniería, U.N.A.M., México, 1984.

McKelvey John P., Grotch Howard, *Physics for science and engineering*, Ed. Harla, U.S.A., 1988.

Philips ECG, Inc., *ECG Semiconductors Master Replacement Guide, 14a. Edición, U. S. A., 1987.*

RCA SK Series, *Reliable Replacement Semiconductors, 2a. Edición, U. S. A., 1988.*

Schildt Herbert, C: *The complete reference*, Second Edition, Osborne McGraw-Hill, Berkeley, Cal., U.S.A., 1990

Siemens, *Microcomputadores*, Marcombo Boixareu Editores, Barcelona, Esp., 1988.

Texas Instruments, *Manual de Semiconductores de silicio*, Edición técnica 1984/1985.

Texas Instruments Incorporated, *The interface circuits databook for design engineers*, Second Edition, U. S. A., 1981.