

27
28



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

**DISEÑO DE UN CONTROLADOR DE
PROCESOS INDUSTRIALES UTILIZANDO
EL MICROCONTROLADOR 8031**

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A N :

ALEJANDRO E. CASTAREDA MONTES

DAVID GALVAN MARTINEZ

JORGE RAPHAEL KOPY

Director de Tesis: Ing. Ernesto Suárez Sport

MEXICO, D. F.

FALLA DE ORIGEN

1991



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.

DEDICATORIAS

A MIS PADRES:

Por todo el cariño y comprensión que me han brindado a lo largo de mi vida y sin los cuales no podría ser lo que ahora soy.

A MIS HERMANOS:

Por saberme entender y darme el ejemplo a seguir.

A ELOISA:

Por todo el cariño, apoyo y motivación que me ha brindado.

A FAMILIARES Y AMIGOS:

Por que han sido una parte fundamental a lo largo de toda mi vida, tanto como persona y como estudiante, y sobre todo por la gran amistad e interés mostrados durante todo este tiempo

ALEJANDRO

A MIS PADRES; Manuel y Josefina, por compartir conmigo su experiencia; enseñándome que con los errores se aprende más que con los aciertos, así como por toda la confianza que depositaron en mí.

A MIS HERMANOS; Guadalupe, Alejandra, Eduardo, Victor y Carlos; por su incondicional apoyo en los momentos difíciles.

Con gran admiración y respeto a la Memoria de mis Abuelos; y en especial a la de mi Tío "Rafa".

A MIS FAMILIARES Y AMIGOS: Por la ayuda desinteresada que siempre me han dado.

A LOS "IMPONDERABLES": Por que con ellos he retomado el verdadero valor de la amistad.

DAVID

Con especial cariño y admiración a mis Padres, Germán y Geraldine, por su gran apoyo, cariño y comprensión. Y principalmente porque gracias a ellos he llegado a ser quien soy.

A mi hermano Germán, por ser como es.

To my Grandpa, for being such a great pal, and a great human being.

A todas aquellas grandes personas que me han motivado a lo largo de mi vida.

JORGE

AGRADECIMIENTOS

Queremos hacer patente nuestro muy especial agradecimiento al Ing. Ernesto Suárez Sport, por su incondicional apoyo, dedicación y tiempo otorgados para la concepción, desarrollo y culminación del presente trabajo. Pero sobre todo por su amistad.

A la Ing. Lourdes Peña Lartigue, por sus valiosos conocimientos, tiempo y apoyo concedidos al desarrollo de la presente tesis.

INDICE

INDICE

INTRODUCCION	1
1.DESCRIPCION GENERAL DE UN PROCESO INDUSTRIAL	3
INTRODUCCION	3
DESCRIPCION DE LOS ELEMENTOS DE UN PROCESO INDUSTRIAL	4
ELEMENTOS QUE INTEGRAN UN PROCESO INDUSTRIAL	5
Almacenamiento de Materia Prima.	5
Procesamiento Inicial.	6
Procesamiento Intermedio.	6
Procesamiento Final.	7
Almacenamiento del Producto.	8
CLASIFICACION DE INDUSTRIAS	9
Por su Tamaño.	9
Por su Producto.	12
EL CONTROL EN LA INDUSTRIA	13
IDENTIFICACION DE PARAMETROS A CONTROLAR	14
Identificación de Variables en las Etapas Comunes a todo Proceso Industrial.	15
Area de almacenamiento de materia prima.	15
Areas de procesamiento.	16
Area de almacenamiento de producto final.	16

Transportación entre etapas del proceso.	16
TIPOS DE CONTROL DE VARIABLES	17
Control Manual.	17
Control Automático.	17
Control Semiautomático.	17
2.ELEMENTOS DE CONTROL	19
ESQUEMAS DE CONTROL	20
Sistemas de Control de Lazo Cerrado.	20
Sistemas de Control de Lazo Abierto.	21
Sistemas de Control Adaptivos.	22
Esquema Real de Control.	22
TIPOS DE CONTROLADORES	22
Controladores de Funcionamiento Autónomo.	23
Controladores Electrónicos.	23
ACCIONES DE CONTROL	23
Controles de Dos Posiciones.	23
Controles Proporcionales.	24
Controles Integrales.	24
Controles Proporcionales e Integrales.	25
Controles Proporcionales y Derivativos (PD).	25
Controladores Proporcionales Derivativos e Integrales (PID).	26
SISTEMAS DE TIEMPO REAL	27
TIPOS DE SISTEMAS DE CONTROL ELECTRONICO	29

CLASIFICACION Y CARACTERISTICAS DE SISTEMAS DIGITALES	30
Sistemas de Control Regulador.	31
Sistemas de Control Supervisor.	31
CONTROLADORES PROGRAMABLES	31
3.DESCRIPCION GENERAL DEL SISTEMA	35
CARACTERISTICAS DE UN ESQUEMA DE CONTROL AUTOMATICO	35
DESCRIPCION GENERAL DEL SISTEMA DE CONTROL	37
PLANTEAMIENTO DE LAS RESTRICCIONES DEL SISTEMA	41
4.DESCRIPCION DE LA UNIDAD DE PROCESO LOCAL (UPL)	43
INTRODUCCION	43
LA UNIDAD DE PROCESO LOCAL (UPL)	44
DESCRIPCION DEL HARDWARE DEL PROTOTIPO	45
CONSIDERACIONES DEL SOFTWARE PARA LA UPL	48
Conceptos importantes que debe considerar el programador de la UPL.	49
Consideraciones del software al nivel ensamblador.	51
5.UNIDAD DE COMUNICACION CENTRAL (UCC)	54
INTRODUCCION	54
DESCRIPCION DE LA FORMA EN QUE SE LLEVA A CABO LA COMUNICACION ENTRE LA UCC Y LAS DIFERENTES UPLs	54
OPERACION DEL CONTROL DE ENLACE DE DATOS (DLC)	56
Tipos de Protocolos de Comunicación.	57
Protocolo controlado por byte (BCP).	57

Protocolos orientados por bit (BOP).	59
Comparación entre los Protocolos.	63
DESCRIPCION DEL DISPOSITIVO DE INTERCONEXION ENTRE LA UCC Y LAS DIFERENTES UPLs	64
Descripción de la Arquitectura del MPSC-8274.	68
Descripción General de la Programación del MPSC-8274.	73
Transmisión en el modo de interrupción.	73
Recepción en el modo de interrupción.	74
6.LA UNIDAD DE PROCESAMIENTO CENTRAL	80
ARQUITECTURA DE LA COMPUTADORA PERSONAL IBM XT	80
Microprocesador 8088.	80
Circuitos de Reloj.	80
Bus del Sistema.	81
Memoria ROM (Read Only Memory).	82
Memoria RAM (Random Access Memory).	82
Contador de Tiempo.	83
DMA (Direct Memory Access).	83
Interrupciones.	83
Operaciones del Bus.	83
Ciclo de lectura a un puerto de entrada/salida.	84
Ciclo de escritura a un puerto de entrada/salida.	85
Comunicación síncrona y asíncrona.	85
TARJETA DE ADQUISICION DE DATOS	86
SOFTWARE PARA EL CONTROL DE SISTEMAS EN LINEA	89

Técnicas Básicas.	90
Sistemas en línea.	90
Definición de software en tiempo real.	90
Sistema de adquisición de datos.	91
Multiprogramación.	92
Programas residentes en memoria.	92
SOFTWARE EN LINEA PARA COMPUTADORAS PERSONALES	93
Sistema Operativo DOS (Disk Operating System).	93
Desarrollo del Software de la UPC.	94
Adquisición de datos.	95
Procesamiento de información.	96
Manejo y control de video.	96
Manejo y control de impresión.	96
7.PROCESAMIENTO ANALOGICO/DIGITAL CON	
MICROCONTROLADORES DE LA FAMILIA MCS-51 DE INTEL	97
INTRODUCCION	97
TECNICA DE CONVERSION POR APROXIMACIONES	
SUCESIVAS	97
CONVERTIDOR INTEGRADOR DE PENDIENTE DOBLE	104
USANDO EL 8031 CON TRANSDUCTORES RESONANTES	109
Introducción.	109
Conectando el Transductor Digital al 8031.	110
Estructura Temporizador/Contador en el 8031.	110
Criterio para Medir la Frecuencia o el Período.	111

Mediciones de la Frecuencia del Transductor.	112
Mediciones del Período del Transductor.	114
Mediciones para el Ancho del Pulso.	117
Exactitud y Resolución.	118
8.DISEÑO DE SISTEMAS CON MICROCONTROLADORES PARA AMBIENTES ELECTRICAMENTE RUIDOSOS	119
SINTOMAS DE LOS PROBLEMAS DE RUIDO	119
TIPOS Y FUENTES DE RUIDO ELECTRICO	120
Transitorios en la Línea de Suministro.	120
Pulsos Electromagnéticos (PEM) e Interferencia por Radio- Frecuencia (IRF).	120
Descarga Electrostática (DES).	121
Corriente de Tierra.	121
Ruido "Radiado" y Ruido "Conducido".	121
SIMULANDO EL AMBIENTE	121
TIPOS DE FALLAS Y MECANISMOS DE FALLA	122
METODOS PREVENTIVOS	123
Lazos de Corriente.	123
Blindaje.	123
Blindaje contra acoplamiento capacitivo.	124
Blindaje contra acoplamiento inductivo.	124
Blindaje contra IRF.	128
Tierras.	131
Tierra de seguridad.	131

Tierra de señal.	131
Aterrizaje práctico.	132
Cable trenzado.	134
Distribución y Desacople de la Fuente de Poder.	134
Como seleccionar el valor del capacitor de desacople.	135
El caso para regulación de voltaje en la misma tarjeta.	136
Recuperándose Adecuadamente de una Falla del Software.	136

CONCLUSIONES

BIBLIOGRAFIA

APENDICE A

LA FAMILIA DE MICROCONTROLADORES MCS-51 DE INTEL

APENDICE B

CARACTERISTICAS GENERALES DEL MPSC-8274

APENDICE C

DISPOSITIVOS EMPLEADOS

APENDICE D

RUTINAS DE INTERRUPCION: ROM BIOS Y DOS

APENDICE E

SISTEMAS DE RESPALDO CON BATERIA PARA EL 80C31BH

APENDICE F

EJEMPLO DE UN TRANSDUCTOR RESONANTE

INTRODUCCION

Al momento de hablar acerca de la interacción de una computadora con un ambiente productivo, inmediatamente surge la idea de automatizar dicho ambiente; es decir, emplear diversos elementos de cómputo, así como su hardware asociado, a fin de ejecutar ciertas acciones precisas en momentos críticos de un proceso. Al efectuar esto, se pretende simplificar, al máximo posible, la toma de decisiones por parte del usuario.

Considerando lo anterior, nos conduce a pensar que en un futuro cercano, competir industrialmente requerirá de una nueva concepción empresarial, misma que contemple la adopción de tecnologías vanguardistas tales como la automatización y sistematización de la producción.

En lo referente a los procesos productivos desarrollados en serie, es una realidad patente que la automatización es una fuerte tendencia que gana cada vez más terreno en todos los sectores industriales. A este respecto, la electrónica ha permitido reducir la intervención del ser humano en ciertas labores, primordialmente del tipo mecánico, en base a la utilización de sofisticados equipos, que si bien internamente son más complejos, ofrecen una mayor versatilidad para el usuario. De esta forma en las instalaciones industriales recientes, se combinan la automatización y la mano de obra sin alta especialización, a fin de reducir los costos de producción.

Como se desprende de lo anterior, el tratar de implantar un esquema productivo a nivel industrial que cuente con las características ya mencionadas, requiere de una inversión muy elevada. De tal forma, que en el presente trabajo se propondrá un **Sistema de automatización** que permita tener el control de ciertas etapas que conforman un proceso industrial, con una cierta reducción en cuanto al hardware asociado al mismo y en consecuencia lograr una reducción en cuanto a los costos del Sistema.

Es importante hacer notar, nuestro trabajo estará enfocado al estudio relacionado con la interacción que surge entre el usuario del Sistema y las diferentes etapas, susceptibles de controlar, que componen un proceso industrial determinado, es decir, que para el desarrollo del Sistema se considera que las señales provenientes del sensado y control de las mismas, son señales limpias (ideales) de tal manera que se tienen señales que están completamente disponibles para ser procesadas directamente por el Sistema. De igual forma, se pretende que el Sistema pueda ser adaptado de acuerdo a las características y requerimientos de la Industria bajo estudio, esto es, que con la misma estructura de hardware propuesta se pueda adaptar sin ningún problema a las características de la industria de tal forma que la única variante entre la adaptación entre cada industria en particular sea el desarrollo de un software especializado para cada una de las mismas. Estudios posteriores permitirán hacer un análisis más profundo de las

señales en forma real, en los cuales si se tienen que tomar en cuenta conceptos como el de tipo de señal, acoplamiento de la señal, etc, de tal manera que puedan ser elegidos los dispositivos de adquisición de datos (transductores, amplificadores de acoplamiento, etc) en forma rápida y de acuerdo a las características de cada una de las industrias bajo estudio.

En resumen se puede decir, que una de las limitantes del Sistema es la etapa de acondicionamiento de señales, ya que se espera que en el uso real, la industria contemple la definición de los transductores y actuadores, de acuerdo a sus propias necesidades, y apegándose a las características del Sistema de adquisición de datos diseñada.

Por lo que respecta al tratamiento temático desarrollado en esta memoria, el primer Capítulo estará destinado a establecer un marco teórico en el que se desenvuelven los procesos industriales en general; en el Capítulo segundo se hace una descripción de algunos conceptos relacionados con la teoría de control; el tercer Capítulo esta orientado a dar una descripción muy general del Sistema propuesto (Controlador de Procesos Industriales basado en el Microcontrolador 8031 de Intel); los Capítulos cuatro, cinco y seis están destinados al estudio y descripción en forma más detallada de los diferentes módulos o elementos que integran el Sistema; el Capítulo siete reseñan las estrategias a seguir para el diseño de Sistemas que operan en ambientes industriales, esto es, en ambientes muy ruidosos; finalmente, el Capítulo ocho está destinado a mencionar dispositivos (para la adquisición de datos provenientes de las variables o parámetros a controlar) que son utilizados actualmente en la industria y que pueden ser fácilmente adaptados a nuestro Sistema, tales como los Transductores así como ciertos convertidores analógico-digital (A/D).

CAPITULO 1

1. DESCRIPCION GENERAL DE UN PROCESO INDUSTRIAL

1.1. INTRODUCCION

En el agitado mundo en que vivimos, es común encontrar que gran parte de nuestros problemas son resueltos mediante computadoras; sin embargo, en contadas ocasiones nos detenemos a reflexionar que esa clase de equipos están diseñados para ejecutar funciones específicas, mismas que satisfacen ciertas necesidades muy particulares del ser humano. No debemos olvidar que una computadora es un sistema completo de procesamiento que opera por la eficiente interrelación de sus elementos constitutivos, los cuales están centrados en un microprocesador.

La historia nos muestra que el mundo industrial ha pasado por diversos períodos tecnológicamente revolucionarios, durante los cuales se ha dado lugar a una fuerte convulsión tanto económica como social.

Considerando lo anterior, hoy por hoy las ciencias de la Computación y de la Electrónica, nos sitúan ante una nueva revolución tecnológica, de ritmo pujante y con una civilización dinámica, propia de las sociedades altamente desarrolladas, (la sociedad postindustrial). Lo anteriormente expuesto nos lleva a pensar que para competir industrialmente, es necesario adoptar técnicas de automatización.

Para poder implantar un esquema productivo a nivel industrial, conviene desarrollar ciertas pruebas que tengan por objetivo facilitar el diseño de los equipos destinados a la automatización de un cierto proceso industrial real. Es interesante observar que dentro de las pruebas se destaca la realización de prototipos y la simulación de un proceso determinado. Cabe comentar que esa simulación involucra el análisis de los principales elementos que intervienen en las diferentes etapas en las que se desempeña el proceso en estudio.

Si se considera que la simulación es una parte trascendente del desarrollo tecnológico e industrial moderno, es pertinente aprovechar los recursos humanos y materiales, disponibles prácticamente en cualquier centro de trabajo convencional. Por tal motivo, un proyecto de gran interés para los ingenieros, tanto del área de electrónica como de sistemas, consiste en desarrollar un equipo tal, que se pueda convertir en una poderosa herramienta útil en el medio productivo automatizado. Con base en lo anterior, y explotando la revolución provocada por el surgimiento de las computadoras personales, el proyecto que aquí se desarrolla pretende generar un sistema que sea capaz de monitorear y controlar una serie de variables analógicas que estén involucradas directa o indirectamente en la secuencia de un proceso industrial dado. El sistema que aquí se plantea podemos conceptualizarlo de la siguiente forma.

Una unidad encargada de obtener y procesar en primera instancia la señal analógica generada en la parte correspondiente del proceso industrial en cuestión. Esta unidad se encarga de hacer la conversión A/D necesaria para llevar a cabo el análisis de la información del proceso y en base a ella determinar la acción a seguir, tomando como base un software de uso particular a cada parte del proceso industrial en general.

Así mismo, se cuenta con una unidad encargada de coordinar todo el flujo de información generado en cada una de las unidades locales a cada subproceso. Esta unidad pasará la información de cada unidad a una computadora personal a través de un cierto protocolo de comunicación.

Finalmente, una computadora personal del tipo PC, quien en coordinación con un software específico permitirá al usuario conocer a detalle la forma en que se ha ido generando el proceso en cuestión; todo ello por medio de análisis gráficos o estadísticos del proceso en general o de cada una de sus partes. Desde este equipo, el encargado de supervisar el sistema podrá modificar los parámetros principales sobre los cuales trabajará el sistema.

1.2. DESCRIPCION DE LOS ELEMENTOS DE UN PROCESO INDUSTRIAL

Acorde con la historia mundial, las industrias hacen su aparición en los siglos XVIII y XIX con la Revolución Industrial producida por la extensión del comercio y de la mecanización de la industria.

La aparición de la máquina de vapor en 1705, perfeccionada por Watt en 1763; de la lanzadera mecánica de Kay en 1773; y de la hiladora de algodón en 1741; entre otros inventos, motivaron el aumento de producción, la disminución de mano de obra y una profunda transformación en los sistemas de producción y de trabajo.

Con estos acontecimientos las industrias comenzaron a adquirir gran fuerza al realizar una producción en serie con el empleo de máquinas, dejando atrás la producción artesanal. Este cambio se dio no solo en Europa, sino que se hizo extensivo a todo el mundo. Al dar inicio este nuevo modo de producción, se dio lugar a una serie de transformaciones en la maquinaria y en los métodos de producción que se han vuelto un constante reto para el hombre.

Desde la antigüedad hasta nuestros días el hombre ha necesitado de medios más efectivos para la elaboración de bienes de consumo, con lo que la tecnología a niveles industriales ha tenido a bien actualizarse en todo momento hasta llegar a los métodos más sofisticados para el control de la industria, involucrando no solo máquinas de vapor, mecánicas, neumáticas e hidráulicas, sino computadoras que realizan rutinas de control preestablecidas por los ingenieros de control de la planta en cuestión.

El empleo de computadoras involucra el uso de dispositivos, generalmente electrónicos, para lograr el control automático que se ha convertido en parte importante e integral de los procesos de manufactura e industriales modernos.

En el presente capítulo se desarrollarán diversos conceptos básicos de los procesos industriales, así como una clasificación de los mismos.

1.3. ELEMENTOS QUE INTEGRAN UN PROCESO INDUSTRIAL

La figura 1.1 muestra de manera general los elementos que integran a cualquier proceso industrial; en tanto que los diagramas posteriores desglozan cada uno de sus componentes.

Como en todo proceso, se necesita de la materia prima para comenzar. Dicha materia prima puede ser un producto obtenido del sector primario (esto es, sin ninguna elaboración) o bien un producto elaborado, como por ejemplo papel para bolsas, empaques, etc.

La figura 1.1 presenta en conjunto, diferentes bloques que inician con el almacenamiento de la materia prima, el procesamiento inicial, el procesamiento final y por último el procesamiento del producto terminado.

Es importante mencionar que entre cada bloque de procesamiento se tiene la transportación de las diferentes etapas del producto, desde que es materia prima hasta que termina su transformación.

Como se mencionó al inicio del tema, se procederá a explicar los elementos que forman a cada bloque que integra el proceso general.

1.3.1. Almacenamiento de Materia Prima.

Una vez que la materia prima ha llegado a la industria, se requiere almacenarla, para ello debe someterse a ciertas etapas de control como son: el control de calidad, de inventario, y de pesos y medidas (figura 1.2).

Para que la materia prima pueda utilizarse debe pasar por un estricto control de calidad, ya que de no cubrir con las especificaciones necesarias, el producto no tendrá las características deseadas. Es por esto, que si la materia prima no satisface las normas de calidad, deberá ser rechazada. Sin embargo, cuando la materia prima pasa el control de calidad, se somete a otra etapa que es el control de inventario, puesto que se requiere saber continuamente la cantidad de materia prima disponible para la elaboración de los

diferentes tipos de producto, y en caso de no ser suficiente, solicitar la compra de más materia prima.

En cualquier tipo de proceso, la materia prima se somete a estas etapas, y en la mayoría de los casos, es aquí, donde se controlan los pesos y medidas de la misma para posteriormente comenzar con sus preparación inicial, que forma parte del siguiente bloque como puede verse en la figura 1.1.

Es importante hacer notar que las materias primas deben contar con ciertas condiciones ambientales en el lugar donde se almacenan. Estas condiciones son: temperatura, humedad, iluminación, grado de esterilización, entre otras.

1.3.2. Procesamiento Inicial.

En general, un producto requiere de más de una materia prima para su elaboración, por lo cual cada una de las materias primas que se requieran deben someterse a un cierto acondicionamiento individual, para cumplir con los requerimientos de elaboración. Por ejemplo, hablando de una industria alimenticia, en particular la panadera, se requiere que las materias primas para la elaboración del pan, reúnan ciertas condiciones, en las que se encuentren listas para efectuar el proceso de elaboración; como es: el huevo pasteurizado, la harina cernida y pesada para hacer la mezcla, la levadura tratada a la temperatura óptima, etc.

Cada una de estas materias primas ha sufrido una transformación individual, después de la cual serán sometidas a un nuevo control de calidad también individual. (Figura 1.3).

Una vez cubierto el control de calidad, se procede a integrar las materias primas adecuadamente para dar lugar a la siguiente etapa de su transformación, que de acuerdo con la figura 1.1 corresponde al procesamiento intermedio.

Con respecto a la integración de las materias primas debe quedar claro que ésta puede ser específica para cada producto, y para ello no habrá ningún esquema o formato general a seguir.

1.3.3. Procesamiento Intermedio.

Hasta el momento, no se puede hablar de producto, pero tampoco se puede continuar con el nombre de materia prima, puesto que ya ha sufrido una transformación. Para efectos de un manejo más adecuado del término, se llamará materia prima secundaria a la materia prima que ha sido transformada en su primera etapa.

Una vez realizada la primera transformación, se llevarán a cabo una serie de pasos que involucran el procedimiento específico que conformará el producto terminal. Al cumplirse cada uno de estos pasos se contará con un control de calidad, que permitirá

revisar continuamente, se tengan las condiciones necesarias para que el proceso sea óptimo.

Nuevamente debe aclararse que el número de pasos, la tecnología, tiempo y secuencia de los mismos, serán diferentes y específicos para cada producto, como puede verse en la figura 1.4. Retomando el ejemplo de la industria panadera, existen diversos productos que se elaboran en la misma planta y aunque en esencia las materias primas son comunes a los diferentes productos, éstos tienen una secuencia y procedimiento distintos entre sí. Se puede intuir que para producir pan blanco, se procede de manera diferente que para hacer bollos. Así como este ejemplo, existen muchos más dependiendo del tipo de industria de que se trate.

1.3.4. Procesamiento Final.

Al terminar el procesamiento intermedio ya se puede hacer mención de un producto terminal, puesto que ha sido sometido a su última transformación de materia secundaria

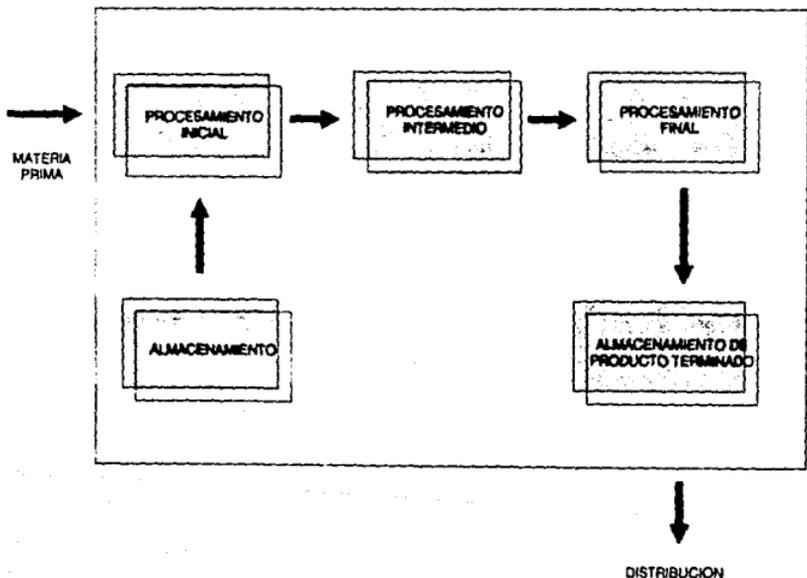


Fig.1.1 Elementos de un proceso industrial

a producto final, y con ésto solo resta envolverlo y/o envasarlo para poder almacenarlo y posteriormente distribuirlo para su venta.

Siguiendo estrictamente el diagrama de la figura 1.5, se puede ver que el producto terminado debe acondicionarse, esto es, darle la presentación adecuada para su venta al público. Esta presentación se refiere a su envoltura, envasado, etc.

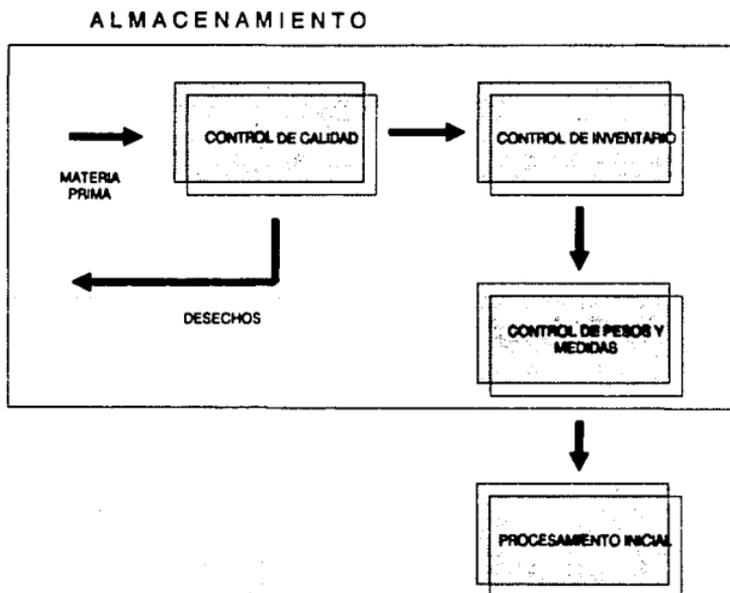
Ya que el producto fue acondicionado, pasa por el control de calidad que permite revisar si realmente es el producto que se esperaba obtener, bajo las condiciones impuestas al principio del proceso.

Si el producto fue el deseado, se manejará adecuadamente para su almacenamiento y posterior distribución.

1.3.5. Almacenamiento del Producto.

Así como la materia prima necesita almacenarse bajo ciertas condiciones ambientales y de control de inventario, el producto final también lo requiere.

Cuando el producto llega al almacén, significa que pasará por un control de inventario, en donde debe tenerse en cuenta la fecha de entrada del producto al almacén,



puesto que hay productos que tienen fecha de caducidad, o bien la empresa sigue ciertas políticas de conservación de sus productos durante un tiempo determinado, y en el caso que este plazo se venza, el producto no sale a la venta.

Ahora bien, las condiciones ambientales del almacén también son muy rigurosas, puesto que los productos necesitan estar en un medio óptimo para no perder sus cualidades. Para esto se mantiene un control de humedad, temperatura, luminosidad, grado de esterilización o pureza del aire, etc.; hasta que el producto se distribuya a los lugares destinados para su venta.

En la figura 1.6 pueden verse los elementos que integran el almacenamiento del producto terminado.

1.4. CLASIFICACION DE INDUSTRIAS

Debido a que las industrias poseen muchas características que las hacen diferentes unas de otras, su clasificación es realmente un serio problema, sin embargo, de manera general se puede enunciar la clasificación de industrias básicas, atendiendo a su tamaño y tipo de producto.

1.4.1. Por su Tamaño.

a) Industrias de Producción Masiva.

Se dice que se cae en la producción masiva, si se manufactura continuamente un elevado volumen de partes, durante un período de tiempo considerable; algunas autoridades hablan del orden de 100,000 piezas por año, aunque tal cifra puede resultar demasiado restrictiva.

En este tipo de industria, el volumen de ventas está bien prestablecido, de tal suerte que la cadencia de producción no depende de pedidos eventuales o particulares.

La maquinaria que genera estas partes se selecciona para conformar las piezas mediante varias operaciones, de tal forma que difícilmente estas máquinas pueden utilizarse para propósitos generales. Dentro de estas piezas que fácilmente pueden reconocerse como partes producidas en forma masiva, están: envases, lápices, automóviles, tuercas, tornillos, guantes, etc.

b) Industrias con Producción Moderada.

En la producción moderada, las piezas se hacen en cantidades relativamente grandes y tal vez en forma continua, pero el producto de la fabricación en cuanto a su

PROCESAMIENTO INICIAL

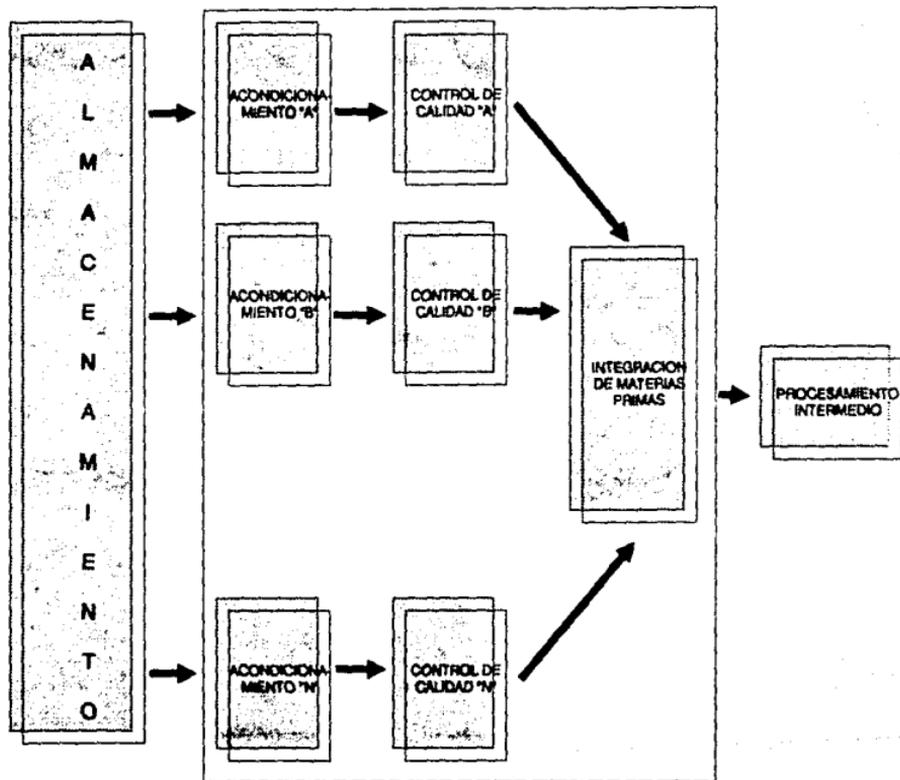


Fig.1.3 Procesamiento Inicial

cantidad, puede ser más variable y con frecuencia depende más de los pedidos eventuales. Aquí la cantidad de piezas que se fabrican puede variar de 25,000 a 100,000 por año dependiendo de su complejidad. Algunos ejemplos de industrias con volúmenes de producción moderada son: editoriales, equipos de dibujo, partes para aeronaves, y equipos transmisores de radio entre otros.

c) Industrias con Producción Limitada.

Las industrias con volúmenes de producción limitada, son más flexibles y su volumen de producción consiste en lotes limitados, dependientes de los pedidos y ventas previstas. El equipo que se utiliza es mucho más versátil, aunque requiere de operadores más competentes para llegar a conformar distintas tareas, dependiendo de la pieza o conjunto. El número de piezas por lote varía entre 10 y 100.

Estas empresas generalmente trabajan tres o mas productos, fabricándolos en cualquier orden y cantidad dependiendo de la demanda.

PROCESAMIENTO INTERMEDIO

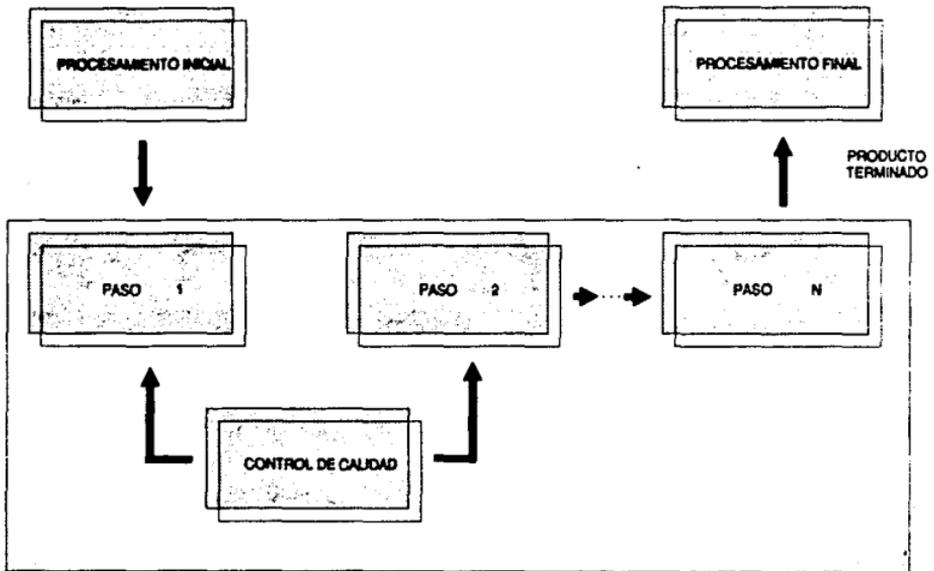


Fig. 1.4 Procesamiento Intermedio

Algunos ejemplos típicos de industrias con volúmenes de producción limitada son: la construcción de aeroplanos, autopartes para unidades antiguas, válvulas, manos artificiales para unidades automatizadas, entre otras.

1.4.2. Por su Producto.

Atendiendo al tipo de producto que se fabrica, las industrias pueden clasificarse como sigue:

- Alimentos y procesados de alimentos
- Manufacturas eléctricas y electrónicas
- Automotriz y transporte

PROCESAMIENTO FINAL

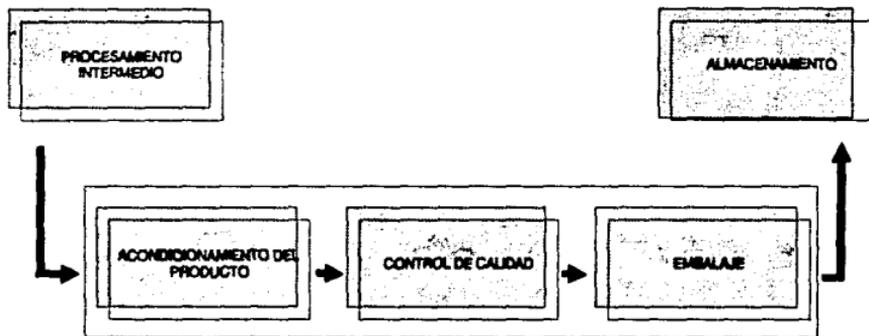


Fig. 1.5 Producto Final

- Accesorios metálicos, plomería, acondicionamiento y refrigeración
- Construcción
- Mobiliario y trabajo de la madera
- Productos de piedra, cristal y arcilla
- Industria extractiva, etc.

1.5. EL CONTROL EN LA INDUSTRIA

En términos generales, casi cualquier máquina o cualquier proceso puede ser automático, pero el uso de los medios de fabricación con tales características depende de los resultados que arroje un previo análisis económico.

Entre todos los avances registrados en los últimos 25 años, el más sobresaliente ha sido el uso de máquinas herramienta controladas numéricamente. El control por cinta

ALMACENAMIENTO

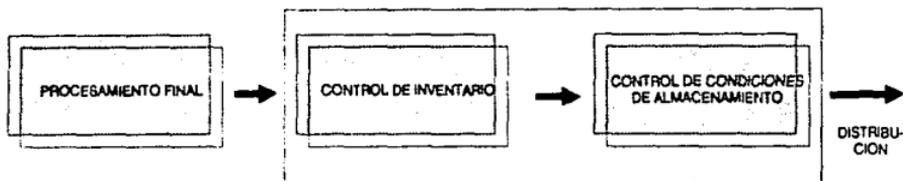


Fig.1.6 Almacenamiento del Producto

perforada, puede aplicarse desde pequeños tornos hasta máquinas fresadoras gigantes. Este avance representa el paso más significativo en el progreso de las máquinas herramienta, avance que durará algunos años; más aún con su perspectiva hacia la concepción y construcción de líneas de producción y máquinas controladas desde un escritorio.

Una empresa grande con un personal de ingeniería competente, mediante un juicio apropiado y con una ingeniería de proyecto creativa, puede comprar máquinas herramienta básicas, cabezales, mandos y otros accesorios para armar máquinas altamente eficientes.

El control numérico significó un símbolo de modernización, pero hoy día es una necesidad en la mayoría de los procesos industriales.

El control industrial involucra el término automatización que nació de la industria automotriz, al descubrir los métodos usados para controlar automáticamente diversas máquinas enlazadas entre sí, con los sistemas de transferencia de piezas. El control puede aplicarse a una máquina o a una serie de ellas, sea cual sea la complejidad de la secuencia de operaciones necesarias para fabricar un producto. Esta secuencia de operaciones puede incluir la alimentación de la carga a las máquinas, la transferencia del producto de

una máquina a otra, las operaciones de inspección y la expulsión final del producto terminado.

La automatización normalmente infiere un proceso restrictivo, que solo es aplicable a la pieza o a un producto dado, pero los controles y las máquinas disponen de cierta flexibilidad, para poder trabajar piezas que difieren ligeramente de aquellas para las cuales las máquinas o el proceso habían sido proyectados. La coordinación no es necesariamente automatización. La automatización implica autocorrección, tal como sucede en los dispositivos para controlar la velocidad de una máquina o en el termostato para controlar la temperatura de una casa.

En otras palabras, la automatización proporciona a la máquina cierta capacidad o habilidad, para seleccionar condiciones de operación aceptables y corregirse a sí misma, cuando ciertos límites preescritos se encuentran excedidos. Una derivación de la mecanización consiste en el control numérico completo del proceso.

Por otra parte, se designa como control numérico, el proceso en el cual las máquinas herramienta, se controlan por medio de instrucciones numéricas, que han sido almacenadas en una cinta perforada o en tarjetas.

La automatización es relativamente inflexible, porque se proyecta para un proceso o un producto individual. En cambio el control numérico es flexible, dado que tanto las máquinas herramienta básicas como las altamente especializadas, se controlan mediante información almacenada y disponible en cualquier momento, y que además, ello no implica la imposibilidad de operar estas máquinas con el elemento humano calificado.

En México, contamos con máquinas herramienta controladas numéricamente y también con procesos controlados por computadora; sin embargo, este tipo de tecnología no es de fabricación nacional y por ende su costo es elevado, de tal manera que para muchas industrias es inalcanzable su adquisición. En base a visitas realizadas a diferentes tipos de industrias en México, se ha podido constatar que la mayoría de éstas no están controladas, ni automatizadas en la totalidad del proceso que efectúan.

En contraposición a esto, existen industrias transnacionales establecidas en México, que sí cuentan con el respaldo económico para adquirir la tecnología moderna, que las hace contar con ciertas etapas automatizadas y en algunos casos excepcionales.

1.6. IDENTIFICACION DE PARAMETROS A CONTROLAR

Así como es posible identificar ciertas etapas comunes a distintos procesos industriales, también es factible determinar el tipo de variables que se pueden encontrar en cada una de estas etapas. En la mayoría de los procesos industriales, independientemente

de su tipo, se tienen que controlar parámetros tales como temperatura, presión, humedad, posición y nivel, entre otros. La diferencia en el manejo que se tenga de estos parámetros, en los diversos procesos, depende del rango en que se deban mantener las variables. En los siguientes párrafos se identificarán estos parámetros en las etapas que resultan ser comunes para todo proceso industrial, así como los distintos tipos de control utilizados.

1.6.1. Identificación de Variables en las Etapas Comunes a todo Proceso Industrial.

En el desarrollo del presente capítulo ha sido posible identificar áreas generales que son aplicables a todo proceso industrial, (figura 1.1). La investigación permitió detectar los parámetros que más comúnmente se presentan en cada una de las áreas generales, así como los métodos utilizados para controlarlos; por lo que en adelante nos avocaremos a describir las variables encontradas en cada etapa.

1.6.1.1. Área de almacenamiento de materia prima.

Generalmente, las materias primas utilizadas en la elaboración de ciertos productos, especialmente en la industria alimenticia y farmacéutica, requieren de un gran cuidado de las condiciones ambientales, para asegurar una calidad óptima en el momento de su utilización. Dentro de este rubro, se pueden encontrar las siguientes variables:

- i) Temperatura.
- ii) Humedad.
- iii) Presión.
- iv) Intensidad luminosa.

En esta área, es donde se da inicio al proceso de control de calidad dentro de la industria. Toda materia prima es sometida a un reconocimiento de calidad, tanto al entrar al almacén, como al momento de pasar a la etapa de procesamiento inicial. Dependiendo del tipo de materia prima, y del producto a elaborar, las variables involucradas en este punto pueden ser:

- Calidad físico-química
- Concentración de acidez (PH)
- Nivel de fluidos
- Peso
- Volumen

1.6.1.2. Áreas de procesamiento.

De acuerdo a la industria de que se trate, y del producto que se vaya a elaborar, se tendrá la delimitación de las áreas de procesamiento inicial, intermedio y final. Es en esta etapa en donde se puede encontrar la mayor variedad de parámetros a controlar, dependiendo del proceso, como son:

- i) Velocidad
- ii) Posición
- iii) Temperatura
- iv) Presión
- v) Fuerza
- vi) Viscosidad
- vii) Humedad
- viii) Nivel de flúidos

1.6.1.3. Área de almacenamiento de producto final.

Al igual que ocurre con la materia prima, muchos productos finales requieren de condiciones controladas en el ambiente existente en el área de almacenamiento, para evitar su descomposición o alteración. Por lo que es recomendable verificar las condiciones de las siguientes variables:

- i) Temperatura.
- ii) Presión.
- iii) Humedad.
- iv) Intensidad luminosa.

1.6.1.4. Transportación entre etapas del proceso.

En toda industria, se tiene una infraestructura, (en algunos casos simple y en otros más compleja), para transportar el producto generado en cada área, hasta hacerlo llegar a la etapa siguiente. Dependiendo de la industria, puede ser que esta transportación sea realizada por una persona, o bien a través de sofisticados mecanismos de bandas transportadoras, donde inclusive se puede encontrar la intervención de brazos

mecánicos. Cualquiera que sea el método utilizado, las variables involucradas en este caso son:

- i) Posición.
- ii) Velocidad.

1.7. TIPOS DE CONTROL DE VARIABLES

Ya se ha tratado el punto de los distintos tipos de variables físicas, y su posible existencia en todo proceso industrial. Toca ahora abordar la forma en que estos parámetros pueden ser controlados. Existen básicamente tres formas de control: manual, semiautomático y automático; cuya elección dependerá de los recursos de cada empresa y, en muchos casos, de sus políticas.

1.7.1. Control Manual.

En la actualidad en muchas de las industrias, gran parte de las variables físicas mencionadas, si no es que todas, son controladas por operadores o supervisores. Este personal se encuentra encargado de revisar continuamente que los parámetros, tanto ambientales como de funcionamiento de los equipos, se encuentren dentro de los rangos de operación y en caso de no ser así, tomar una acción para regularlos. El recurso humano, sin embargo, no siempre reacciona, ante un estado de alarma, con la rapidez necesaria para controlar el proceso, o en su caso detenerlo, sin una considerable pérdida de materia prima.

1.7.2. Control Automático.

Así como existen industrias que controlan manualmente los parámetros que interviene en sus procesos, también las hay en las que la mayoría de sus procesos son controlados automáticamente. Los medios de control automático pueden variar, dependiendo de la variable a controlar y de los recursos para hacerlo. Los dispositivos más utilizados en el control industrial son los Controladores Programables, así como algunas versiones especiales de computadoras personales.

1.7.3. Control Semiautomático.

Un tercer tipo de control es aquél que se encuentra en un punto intermedio entre los dos anteriores, donde se cuenta con dispositivos de detección o sensado, que accionan algún tipo de alarma. Esta clase de sistemas son sólo de detección, ya que no son capaces de tomar ninguna acción correctiva, misma que debe ser tomada por un operador.

CAPITULO 2

2.

ELEMENTOS DE CONTROL

Como analizamos en el Capítulo I, en el área de procesos industriales resulta esencial llevar a cabo operaciones de control de ciertas variables (presión, temperatura, humedad, etc.), cuya magnitud debe mantenerse cercana a un valor fijado previamente, llamado punto de calibración.

El objetivo primordial de un sistema de control es mantener controlada una o más variables, comparando la señal medida con el valor deseado, y si existe alguna diferencia entre éstos, se debe actuar sobre el proceso hasta lograr minimizar dicha diferencia. La función de control descrita anteriormente, puede realizarse de varias formas, tales como acciones mecánicas, neumáticas, hidráulicas, eléctricas y electromecánicas, ya sean analógicas o digitales.

En relación a esto, el control automático de un proceso disminuye en gran medida los errores que pueden ser cometidos por un humano al tomar decisiones inmediatas, y provee la continua respuesta estable en determinado proceso susceptible de pequeños cambios; por lo que el **control automático** puede definirse como un estado de balance de las condiciones propias del proceso que tienden a sufrir cambios, dependiendo del estado en que éste se encuentre.

La figura 2.1 ilustra un diagrama de bloques en el que se muestran los componentes básicos de un sistema de control.

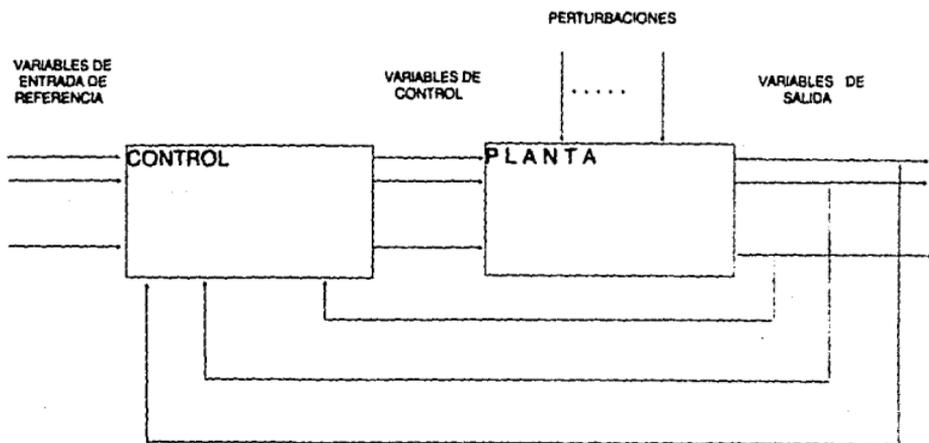


Fig. 2.1 Diagrama de Bloques de un Sistema de Control

Dichos componentes se clasifican en dispositivos de medición y dispositivos de control; ambos intervienen en el control del sistema, ya que interactúan con las variables y los mecanismos del proceso.

El control automático está acompañado por un ciclo de eventos que comienzan con un cambio en la variable controlada y termina con regresar la variable a su nivel de estabilidad. El cambio de estado en la variable, es inevitable porque en esto se basa la acción de control y es la forma en la cual el controlador responde a los cambios de la variable, quedando la selección del modo de control supeditada a una extensa variedad de condiciones, así como la calidad de control deseada.

Los conceptos anteriores, servirán de base para que en el presente capítulo se analicen los esquemas básicos de control, algunas clasificaciones de los sistemas de control y por último las ventajas que ofrecen las computadoras personales y los controladores programables para el control de procesos industriales.

2.1. ESQUEMAS DE CONTROL

Debido a que existen numerosos y muy variados esquemas de control, en esta sección se definirán los de mayor interés para el desarrollo de este trabajo, como son:

- Sistemas de Control de Lazo Cerrado
- Sistemas de Control de Lazo Abierto
- Sistemas de Control Adaptivo
- Esquema Real de Control

2.1.1. **Sistemas de Control de Lazo Cerrado.**

Un sistema de control de **lazo cerrado o malla cerrada**, es aquel en el que la señal de salida tiene efecto directo sobre la acción de control; esto es, los sistemas de control de lazo cerrado son sistemas de control realimentado. La señal de error actuante, que es la diferencia entre la señal de entrada y la de realimentación (que puede ser la señal de salida o una función de ésta), entra al detector o control a manera de reducir el error y llevar la salida del sistema a valor deseado. En otras palabras, el término lazo cerrado implica el uso de acción de realimentación para reducir el error del sistema.

La figura 2.2 ilustra un diagrama de bloques de un sistema típico de control de lazo cerrado.

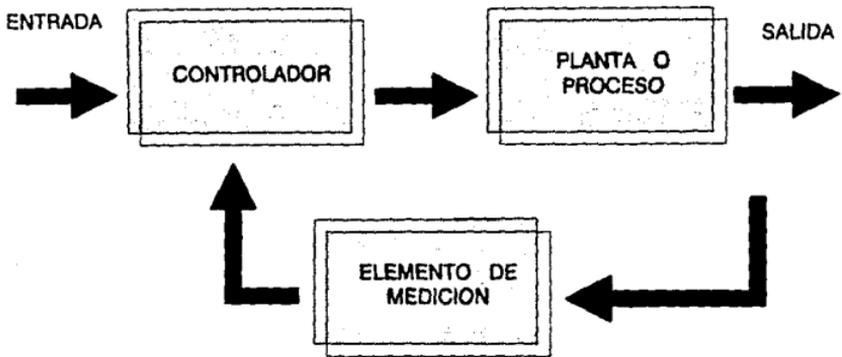


Fig. 2.2 Sistema de Control de Lazo Cerrado

2.1.2. Sistemas de Control de Lazo Abierto.

Los sistemas de control de **lazo abierto** también llamados de **mallo abierta**, son sistemas de control en los que la salida no tiene efecto sobre la acción de control. Es decir, en un sistema de control de lazo abierto la salida ni se mide ni se realimenta para ser comparada con la entrada. La figura 2.3 muestra la relación entrada-salida de tal sistema.

Es conveniente tomar en cuenta que en presencia de perturbaciones, en un sistema de control de lazo abierto no se satisfacen las condiciones de la función que asigna una entrada con una salida en forma determinística. En la práctica, sólo se puede usar el control de lazo abierto, si la relación entre la entrada y la salida es conocida y si no hay perturbaciones internas o externas.

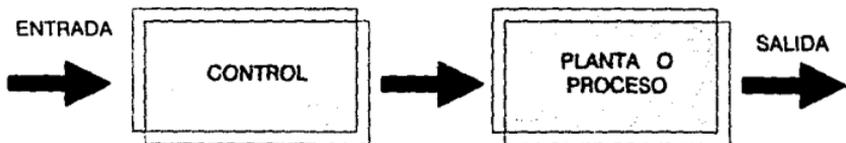


Fig. 2.3 Sistema de Control de Lazo Abierto

2.1.3. Sistemas de Control Adaptivos.

Las características dinámicas de la mayoría de los sistemas de control no son constantes por diversas razones, como el deterioro de los componentes al transcurrir el tiempo o las modificaciones en parámetros o en el medio ambiente. En un sistema de control realimentado se atenúan los efectos de pequeños cambios en las características dinámicas, pero si las modificaciones en los parámetros del sistema y el medio son significativas, el sistema para ser satisfactorio ha de tener la capacidad de adaptación. La adaptación implica la capacidad de autoajustarse o automodificarse de acuerdo con modificaciones imprevisibles del medio o de la estructura en que trabaja; por lo que a los sistemas que poseen algún nivel de adaptación se les denomina sistemas de control adaptados o adaptivos.

2.1.4. Esquema Real de Control.

En todo proceso existen condiciones o variables fundamentales (temperaturas, presiones, flujos, etc.) que rigen y determinan en gran medida la operación global del proceso y del sistema de producción. Los sistemas de control se implementan con el fin de manejar dichas variables para mantenerlas el mayor tiempo y tan cerca como sea posible de sus valores especificados, logrando así los objetivos antes mencionados.

En un esquema general de control se observa que el lazo de control comienza sensando continuamente la variable clave, mediante el elemento primario de medición (transductor), éste detecta la variable y a partir de ella induce un efecto de tipo mecánico o eléctrico principalmente, que es tomado por el transmisor/indicador para producir la indicación y la señal transmisible, correspondiente a la magnitud de la variable medida. En el controlador, esta variable se compara con el valor especificado o de referencia, y ante una diferencia, el controlador reconoce la magnitud y signo de la desviación, con tal de definir las acciones destinadas a corregir el valor de la variable de proceso para ajustarlo al valor de referencia. El actuador y el elemento final de control se encargan de efectuar las acciones dictadas por el controlador; el actuador activa el elemento final de control para que maneje adecuadamente aquella energía o material de proceso, que, interactuando con el medio, modifica y/o mantiene la magnitud de la variable fundamental.

2.2. TIPOS DE CONTROLADORES

Los sistemas de control que en la actualidad presentan grandes perspectivas de desarrollo son los Sistemas de Control Eléctricos y Electrónicos, ya que presentan características más deseables para el control de procesos, algunas de las cuales son su alta velocidad de respuesta, gran precisión y sensibilidad, y la disponibilidad de la fuente de suministro de energía necesaria. Los controladores eléctricos y electrónicos son muy versátiles, y en la actualidad muy económicos, debido al desarrollo que han alcanzado

en los últimos años, por lo cual su aplicación se ha extendido. Los controladores de esta clase manejan señales de entrada eléctricas, las que provienen de elementos sensores, de donde se acondicionan para que estén de acuerdo a las características del controlador. Las salidas del controlador también son eléctricas, las cuales excitarán actuadores eléctricos, electroneumáticos o electrohidráulicos.

Los controladores eléctricos y electrónicos se pueden clasificar en base a dos aspectos: por el tipo de controlador y por el modo de control.

Atendiendo a la clasificación anterior, los controladores eléctricos y electrónicos pueden ser de dos tipos: controladores de funcionamiento autónomo y controladores electrónicos.

2.2.1. Controladores de Funcionamiento Autónomo.

Esta clase de dispositivos obtienen la energía que necesitan para la operación del elemento final de control, así como la de su detector de error, del mismo proceso que están regulando. Lo anterior se logra mediante el uso de un elemento sensor como un diafragma, fuelle, flotador, etc. Típicamente, estos controles son aparatos muy simples y muy empleados en industrias químicas para el control de temperatura, presión, nivel, etc.

2.2.2. Controladores Electrónicos.

El gran desarrollo que han tenido los sistemas electrónicos digitales en las últimas décadas, los ha llevado a ser los sistemas de mayor uso en la actualidad.

Con los sistemas digitales se pueden implementar algoritmos de control muy sofisticados, además de los utilizados por los clásicos controles analógicos y por los sistemas neumáticos e hidráulicos.

2.3. ACCIONES DE CONTROL

A continuación se explicarán con mayor detalle las acciones de control analógicas más comunes.

2.3.1. Controles de Dos Posiciones.

En un sistema de control de dos posiciones, también llamado ON/OFF, el elemento accionador tiene solamente dos posiciones fijas, que en muchos casos son simplemente conectado y desconectado. El control de dos posiciones es relativamente simple y económico y, por esta razón, ampliamente utilizado en sistemas de control tanto industriales como domésticos.

Sea la señal de salida de control $m(t)$ y la señal de error actuante $e(t)$, en un control de dos posiciones, la señal $m(t)$ permanece en un valor máximo o mínimo, según sea la señal de error actuante, positiva o negativa, de modo que:

$$m(t) = M_1 \quad \text{para } e(t) > 0$$

$$m(t) = M_2 \quad \text{para } e(t) < 0$$

donde M_1 y M_2 son constantes. Generalmente el valor mínimo M_2 es o bien cero, o $-M_1$. Los controles de dos posiciones son generalmente dispositivos eléctricos, donde habitualmente hay una válvula accionada por un solenoide eléctrico.

2.3.2. Controles Proporcionales.

Para un control de acción proporcional, la relación entre la salida del controlador $m(t)$ y la señal de error actuante $e(t)$ es:

$$m(t) = K_p e(t)$$

y su función de transferencia es:

$$\frac{M(s)}{E(s)} = K_p$$

donde K_p se denomina **sensibilidad proporcional** o **ganancia**.

Cualquiera que sea el mecanismo en sí, y sea cual fuere la potencia que lo alimente, el control proporcional es esencialmente un amplificador de ganancia ajustable.

2.3.3. Controles Integrales.

En un control con acción integral, el valor de la salida del controlador $m(t)$ varía proporcionalmente a la señal de error actuante $e(t)$. Es decir:

$$\frac{dm(t)}{dt} = K_i e(t)$$

o bien

$$m(t) = K_i e(t) dt$$

donde K_i es una constante regulable. La función de transferencia del control integral es:

$$\frac{M(s)}{E(s)} = \frac{K_i}{s}$$

Si se duplica el valor de $e(t)$, el valor de $m(t)$ varía dos veces más rápido.

Para un error actuante igual a cero, el valor de $m(t)$ se mantiene estacionario. La acción de control integral recibe también el nombre **control de reposición**.

2.3.4. Controles Proporcionales e Integrales.

La acción de control proporcional e integral queda definida con la siguiente ecuación:

$$m(t) = K_p e(t) + \frac{K_p}{T_i} e(t) dt$$

o la función de transferencia de control es:

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right)$$

donde K_p representa la sensibilidad proporcional o ganancia, y T_i el tiempo integral. Tanto K_p como T_i son regulables. El tiempo integral regula la acción de control integral, mientras una modificación en K_p afecta tanto a la parte integral como a la proporcional de la acción de control. Al inverso del tiempo integral se le denomina **frecuencia de reposición**; dicha frecuencia está definida como el número de veces por minuto que se duplica la parte proporcional de la acción de control. Esta frecuencia se mide en términos de repeticiones por minuto.

2.3.5. Controles Proporcionales y Derivativos (PD).

La acción de control proporcional y derivativa queda definida por la siguiente ecuación:

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

y la función de transferencia es:

$$\frac{M(s)}{E(s)} = K_p(1 + T_d s)$$

donde K_p es la sensibilidad proporcional, y T_d el tiempo derivativo. Tanto K_p como T_d son regulables. La acción de control derivativa, ocasionalmente denominada **control de velocidad**, está dada cuando el valor de salida del control es proporcional a la velocidad de variación de la señal de error actuante. El tiempo derivativo T_d es el intervalo de tiempo en el que la acción de velocidad se adelanta al efecto de acción proporcional.

Mientras que la acción de control derivativo posee la ventaja de ser anticipadora, también tiene las desventajas de que amplifica las señales de ruido y puede producir efecto de saturación en el actuador. Hay que hacer notar que nunca se puede tener una acción de control derivativo sola, porque este control es efectivo únicamente durante períodos transitorios.

2.3.6. Controladores Proporcionales Derivativos e Integrales (PID).

La combinación de los efectos de acción proporcional, acción de control derivativa y acción de control integral, se llama **acción de control proporcional derivativa e integral**. Esta acción combinada tiene las ventajas de cada una de las tres acciones de control individuales. La ecuación de un control con esta acción combinada está dada por:

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} + \frac{K_p}{T_i} \int e(t) dt$$

o la función de transferencia es:

$$\frac{M(s)}{E(s)} = K_p \left(1 + T_d s + \frac{1}{T_i s} \right)$$

donde K_p representa la sensibilidad proporcional, T_d el tiempo derivativo y T_i el tiempo integral.

La mayor parte de los controles automáticos industriales usan como fuente de potencia la electricidad o un fluido a presión, que puede ser aceite o aire. En base a esto, también pueden clasificarse en controles neumáticos, hidráulicos o electrónicos.

El tipo de control a usar, depende de la naturaleza de la planta, y sus condiciones de funcionamiento, incluso condiciones de seguridad, costo, disponibilidad, confiabilidad, precisión, peso y tamaño.

2.4. SISTEMAS DE TIEMPO REAL

Los sistemas de tiempo real han adquirido una gran importancia en los últimos años. El dramático descenso en los costos de los circuitos integrados los han hecho accesibles para aplicaciones industriales, comerciales, educativas, domésticas e incluso para diversión. Como ejemplos se pueden mencionar: el sistema de control computarizado de una fábrica de pan, el sistema de control de un motor de pasos, sistemas de seguridad por medio de microcomputadoras, el sistema de ignición de un coche, etc.

Cada uno de los sistemas mencionados requieren de un respaldo de programas de acuerdo al tipo de aplicación, de tal manera que es necesario utilizar técnicas de diseño de software que sean útiles en un amplio rango de aplicaciones. Además, también se hace necesario tener conocimientos sobre circuitos integrados, ciertas interfaces de comunicación, etc.

Una definición sencilla de un sistema en tiempo real sería la de aquel sistema que responde a la necesidad de actuar en un período de tiempo proporcional a la urgencia de la necesidad. Específicamente el sistema se basa en proporcionar la información necesaria para actuar sobre el mundo real.

El tiempo real describe el procesamiento de información de una manera suficientemente 'rápida', en la que el resultado del procesamiento está inmediatamente disponible para controlar el proceso que se está monitoreando. Básicamente cualquier computadora puede configurarse para ejecutar algunas operaciones de tiempo real, el criterio primordial es que la computadora sea capaz de ejecutar una acción específica en un período de tiempo particular.

Algunas de las propiedades con las que deben contar los sistemas en tiempo real son:

a) Tiempo de Respuesta Adecuado.

Un sistema que no reaccione lo suficientemente 'rápido' en el medio en el que actúa, no puede considerarse un sistema de tiempo real. Sin embargo se debe considerar que 'rápido' depende del sistema que se trate. Por ejemplo, para controlar el curso de un proyectil el sistema de tiempo real debe reaccionar en unos cuantos milisegundos; una respuesta de dos segundos después, sería crítica. Sin embargo, en un sistema bancario en línea, la respuesta de saldo de cuenta puede tardarse unos cuantos segundos y seguiría siendo 'rápido'.

b) El Sistema debe ser lo más Completo y Preciso Posible.

En el sistema se deben de preveer todas las situaciones normales y anormales que pudiesen suceder en cierta parte de un proceso, de tal forma que se les pueda dar la solución más adecuada, aunque en ocasiones se hace muy difícil especificar los requerimientos de un sistema diseñado para controlar una cierta porción del mundo real, debido al amplio rango de situaciones y condiciones que ocurren en él.

Sin embargo, todas aquellas situaciones que se tengan contempladas en el proceso deben de ser tratadas en forma adecuada y precisa, pero además aquellas situaciones que no se tengan contempladas deben ser tratadas rápidamente de tal forma que se les pueda dar la solución más adecuada.

c) Confiabilidad.

Se debe pretender que el sistema sea lo más confiable posible, es decir, que en el caso de que se presente una falla en éste, él mismo tenga la capacidad de solucionar de inmediato la falla y que no se presente ninguna alteración en el proceso en curso. Generalmente, este tipo de situaciones se solucionan con el diseño de un sistema redundante, en el cual, se pretende que funcionen en forma paralela ciertos elementos o dispositivos iguales a los que se presenten en el diseño original, esto es, que en el momento que se presente una falla en alguno de los dispositivos que integran el sistema, el dispositivo que se encuentra funcionando en forma paralela entra automáticamente a formar parte del proceso original y es entonces cuando se elimina la posibilidad de paro del proceso y en consecuencia se elimina la posibilidad de alteración del mismo.

d) Economía.

En este punto se debe de hacer una evaluación del costo-beneficio del sistema a diseñar. Para poder competir con el diseño de sistemas de control ya hechos, es necesario tomar muy en cuenta este punto, ya que lo que se pretende es hacer alguna acción del proceso con la menor cantidad de dispositivos o elementos, sin dejar a un lado la confiabilidad del sistema, en otras palabras, que sea lo más eficiente posible con la menor cantidad de elementos a interactuar.

e) Disponibilidad de los Elementos del Sistema.

En el diseño se debe contemplar la disponibilidad de los elementos en el mercado, en específico, en el mercado de México, para que de esta forma se pueda dar soporte o mantenimiento, lo más rápidamente posible a los componentes que integran el sistema.

2.5. TIPOS DE SISTEMAS DE CONTROL ELECTRONICO

En época reciente los sistemas electrónicos de control han tenido un gran auge, ya que sobre otros tipos de sistemas, presentan ventajas tales como: mayor disponibilidad, servicio de mantenimiento fácil, menor tamaño físico, costos de instalación menores, etc.

Otra ventaja que presentan los sistemas electrónicos en especial sobre los sistemas neumáticos, es el reemplazo de ductos de transmisión por cables eléctricos, lo cual reduce substancialmente, o bien elimina por completo las pérdidas neumáticas que degradan el rendimiento de las mallas de control. Por otra parte los controles electrónicos pueden ser sintonizados con mayor precisión que otros sistemas.

En general, los sistemas de control electrónicos pueden agruparse en dos categorías: controles continuos y controles discretos. Fundamentalmente los controles continuos se han desarrollado para cubrir las necesidades de la industria de proceso y transformación, y el control discreto en las industrias de manufactura.

Los sistemas de control continuo o analógico pueden realizar funciones tales como: control de flujo, temperatura, nivel, presión, etc. Este tipo de controles manipula las variables del proceso para mantenerlas dentro de un rango predeterminado.

Por otra parte, los controles discretos o digitales se emplean ampliamente en el control de procesos industriales. Una de las primeras funciones que realizaron este tipo de dispositivos fue la adquisición de datos. En esta clase de aplicaciones se capturan los datos directamente del proceso para ser analizados posteriormente y tabulados.

Más tarde los sistemas de control digital tuvieron una aplicación más sofisticada consistente en una acción previsor, en la cual los datos del proceso son empleados para realizar y optimizar los cálculos de las expresiones del algoritmo de control. Estos cálculos proveen los puntos de calibración adecuados para los controladores analógicos que son los que realizan la primera intervención de control dentro del proceso.

Por último, el uso más poderoso que representan este tipo de sistemas es el **control digital directo (DDC)**. En este caso, el sistema digital de control captura los datos directamente del proceso, empleando dicha información para resolver ecuaciones que son equivalentes a las funciones analógicas de control; una vez hecho esto, el sistema ajusta los actuadores para efectuar un control regulatorio apropiado. Los puntos de calibración de este tipo de mallas pueden, además, ser suministrados ya sea por el

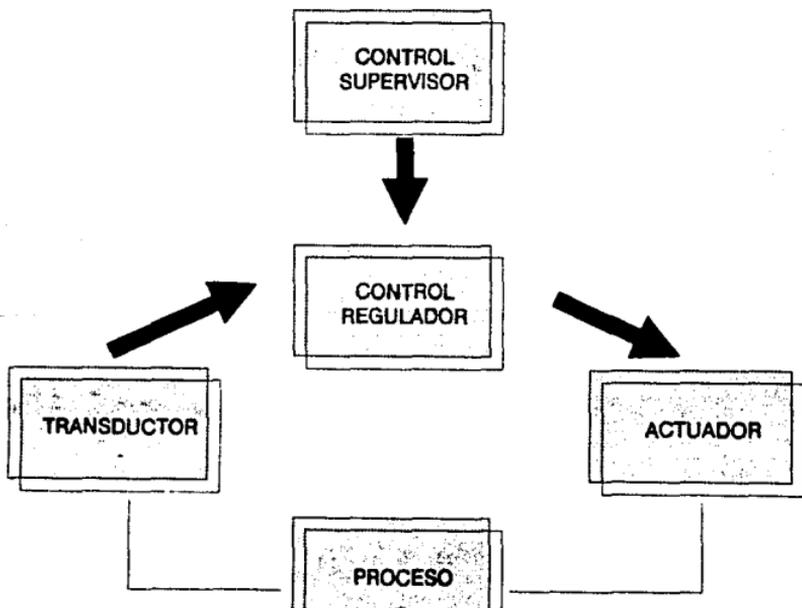
operario o bien por el mismo sistema al realizar una acción supervisora. Las ventajas que presentan los sistemas DDC sobre los controles analógicos reside en la flexibilidad que existe para implementar algoritmos de control más sofisticados que el Proporcional Integral Derivativo.

Una desventaja importante en los sistemas DDC es que cuando se presenta una falla, se afectan todas las mallas de control del proceso. Por esta razón, los sistemas de control digital generalmente cuentan con un respaldo analógico para los puntos más críticos del proceso bajo control.

2.6. CLASIFICACION Y CARACTERISTICAS DE SISTEMAS DIGITALES

Los sistemas digitales pueden realizar diferentes acciones de control dependiendo de su complejidad, sin embargo, si tomamos en cuenta el tipo de funciones que ejecutan, podemos clasificarlos en dos categorías: sistemas de control regulador y sistemas de control supervisor.

La figura 2.4 muestra un diagrama de bloques que incluye a estos dos sistemas de control involucrados en un proceso.



2.6.1. Sistemas de Control Regulator.

Este tipo de sistemas, conocidos también como controladores reguladores, aceptan una o varias señales provenientes de uno o varios sensores, con el fin de determinar los ajustes a realizar, si es que hay que efectuar alguno, para mantener los valores críticos del proceso en un entorno cercano a los puntos de calibración, previamente determinados. Esencialmente el propósito de este tipo de controladores, es mantener el proceso dentro de los límites especificados de operación.

Las características que se encuentran en estos sistemas, son: requerir de un gran número de entradas, muestreo rápido de dichas entradas, computación relativamente veloz, gran número de salidas, equipo de respaldo para fallas, etc. Dentro de esta categoría se incluyen los controladores DDC.

2.6.2. Sistemas de Control Supervisor.

El propósito fundamental de los sistemas de control supervisor, es determinar las condiciones óptimas de operación del proceso en un instante dado. Por otra parte, podemos considerar que estos sistemas generalmente son los responsables del procesamiento numérico de la información de acuerdo al algoritmo de control que se esté aplicando, con el fin de obtener el mejor proceso posible.

Entre las características que presentan este tipo de sistemas se encuentran: requerir de un menor número de entradas respecto a los controladores reguladores, menor equipo de respaldo, computación y muestreo relativamente lento.

Al analizar las características, es fácil suponer que estos controladores permiten la existencia de jerarquías dentro de los sistemas de control, ya que se puede considerar que un sistema regulador puede ser esclavo de un sistema supervisor, siendo éste a su vez esclavo de otro sistema supervisor y así sucesivamente.

2.7. CONTROLADORES PROGRAMABLES

Podemos definir un controlador programable como "un aparato electrónico que opera digitalmente, el cual usa una memoria programable para almacenamiento interno de instrucciones que representan funciones lógicas, secuenciales, temporizadas, y de conteo a través de módulos analógicos o digitales de E/S, para varios tipos de máquinas o procesos.

Los controladores programables comprenden básicamente los siguientes componentes: interfaces de E/S, procesador, memoria, dispositivos de programación, fuentes de poder, etc.

Las interfaces de E/S conectan al controlador con el equipo industrial que será controlado. La interface de entrada recibe el proceso y las señales de la máquina, las convierte a la forma adecuada en que el controlador puede aceptarlas.

El procesador tiene tres funciones principales. Ejecuta varias rutinas lógicas, realiza operaciones en las señales de entrada y determina las salidas apropiadas; los programas usados por el procesador para realizar estas funciones son almacenados en la memoria del controlador.

Algunos dispositivos de programación son usados para cargar el programa en memoria. Hay dos tipos de éstos que son muy comunes: una terminal CRT o un teclado lógico. El uso de una terminal CRT permite la aplicación de lenguajes de programación para introducir datos a través de un teclado lógico con botones que tienen asignada una función especial.

Existen tres tipos básicos de lenguajes de programación usados en controladores programables: diagramas de relevadores de escalera, lenguajes basados en funciones booleanas y lenguajes mnemónicos que son similares a los lenguajes ensambladores.

Los diagramas de relevadores de escalera consisten en símbolos que representan contactos abiertos, o cerrados y otros componentes de control de equipo eléctrico.

Los lenguajes basados en funciones booleanas, establecen relaciones entre las entradas y salidas del controlador. Estas relaciones incluyen funciones AND, OR e IGUAL.

Los lenguajes mnemónicos incluyen funciones como LOAD, AND, OR y STORE.

Estos lenguajes pueden ser usados para programar una variedad de funciones en el controlador incluyendo:

a) Funciones de control en relevadores. Estas involucran la generación de una señal de salida, a partir de una o más entradas de acuerdo a una lógica particular, contenida en el contador de programa de la memoria.

b) Funciones de tiempo. Usadas para generar por ejemplo, una señal de salida con un retraso específico, después de haber sido recibida una señal de entrada.

c) Funciones de conteo. Las cuales son similares a las funciones de tiempo, en las que el contador suma el número de entradas en contacto, y genera una cierta salida programada cuando la suma ha alcanzado una cuenta definida.

d) Funciones aritméticas. Como adición, sustracción, multiplicación y división.

e) Funciones de control analógico. Con las cuales se habilita el controlador programable directamente de dispositivos analógicos, particularmente de funciones de control proporcional, integral y derivativo.

Los controladores programables pueden clasificarse de diversas maneras, una de ellas es en cuanto a su capacidad de entradas y/o salidas: pequeños, medianos y grandes.

Aún cuando los controladores programables han avanzado mucho en cuanto a su capacidad, otras áreas comienzan a abrirse paso en este campo de control: las computadoras personales de tipo industrial y los dispositivos inteligentes de entrada/salida.

En los últimos cinco años, las computadoras personales han proliferado enormemente por la diversidad de usos, y han tenido un efecto significativo en las oficinas, las escuelas y el hogar. Estas han sido utilizadas en aplicaciones de laboratorio, pero sólo recientemente han sido consideradas para aplicaciones industriales. Esto ha sido posible después de éxitos significativos con las computadoras personales.

Ahora bien, es sabido que las computadoras industriales están cambiando, y adquiriendo tales características que permiten prescindir de los controladores programables dedicados a controlar los parámetros indispensables en cualquier tipo de industria.

Cuando se tiene la necesidad de efectuar algún tipo de control, se deben considerar dos requerimientos indispensables: la seguridad y la respuesta del sistema, por lo cual es importante distinguir entre computadoras y controladores programables.

Generalmente, una computadora está diseñada para procesar grandes volúmenes de datos rápidamente, mientras que un controlador programable está diseñado para controlar una planta con un sistema de respuesta rápida.

En un controlador programable podemos distinguir dos elementos: el controlador y las interfaces de entrada/salida. La parte central del sistema es el controlador, el cual es programado para examinar el estado de la planta, decide qué cambios se necesitan hacer y da las órdenes para que la planta cambie su estado. Aunado a esto, los controladores programables modernos pueden almacenar información, y además comunicarla a otros controladores, computadoras o impresoras.

Las interfaces de entrada/salida permiten al controlador programable comunicarse con la planta, convertir las señales recibidas en información que el controlador pueda entender y traducir instrucciones de éste a señales de control actuantes. Hasta el momento el controlador es programado en base a diagramas lógicos de relevadores eléctricos. En la actualidad los controladores programables avanzados cuentan con capacidad matemática, permitiéndoles ser usados para procesamiento de señales numéricas. Estos propósitos de diseño indican que un controlador programable es un dispositivo capaz de controlar maquinaria con mínimo esfuerzo y utilizando los componentes necesarios para el usuario.

La velocidad de procesamiento es razonablemente rápida, pero no tanto como una computadora, porque realiza continuamente una verificación de error para lograr óptima seguridad.

Una computadora está diseñada para operar en un ambiente limpio y tiene mayor capacidad matemática que un controlador programable sin necesidad de verificar errores, por lo tanto su velocidad de procesamiento es mayor.

La computadora es utilizada para control de maquinaria de tal forma que sirve de guía a los sistemas, donde los controladores programables no tienen suficiente velocidad. Si la computadora es requerida para algoritmos complejos, entonces las entradas a la planta, pasan los datos a la computadora para procesarlos.

Para que una computadora pueda usarse en una planta, el usuario requiere de elementos de hardware, además de protegerla ampliamente del ambiente de la industria, a diferencia de un controlador programable, que es un dispositivo más robusto.

Muchas fábricas se avocan al uso de las computadoras personales, como dispositivos de programación, pero en el área de control está restringido su uso a laboratorios, adquisición de datos, etc., manteniéndose en un ambiente limpio y con tiempos de respuesta que no son críticos.

En este momento, la tendencia es incrementar la inclusión de los controladores programables en sistemas de control de procesos y control industrial. Esto involucra la comunicación no solo entre PC y dispositivos directamente involucrados en el control, sino también entre controladores programables de manera individual y con computadoras y equipo periférico tal como impresoras.

CAPITULO 3

3.

DESCRIPCION GENERAL DEL SISTEMA**3.1. CARACTERISTICAS DE UN ESQUEMA DE CONTROL AUTOMATICO**

Dadas las condiciones de ciertas industrias especializadas en el área de control, ubicadas en nuestro país, se detectaron tres formas básicas de control:

- Manuales
- Semiautomáticas
- Automáticas

De acuerdo a lo expuesto en los Capítulos 1 y 2, en donde se hacía referencia a la identificación de parámetros susceptibles de controlar en un proceso industrial, así como de los esquemas típicos de control utilizado en las diferentes industrias, se puede inferir que la industria en México cuenta con sistemas de control manuales con tendencia a ser semiautomáticos; sin embargo, los equipos que se utilizan en muchos casos son antiguos y, en ocasiones, obsoletos.

Lo anterior hace surgir la necesidad de automatizar, al máximo posible, la mayoría de los esquemas de control a utilizar en una planta industrial. En referencia a este particular, surgió la idea de estudiar a fondo ese problema que se vive en la industria mexicana, ya que en la actualidad el país podría contar con los elementos necesarios para llevar a cabo esta tarea, es decir, que se puedan implementar sistemas de control industrial altamente automatizados.

Todo **sistema de control automático** puede concebirse básicamente como se muestra en la figura 3.1 .

Resulta claro que para controlar cualquier variable en tal esquema de realimentación, es necesario medirla primero. Por tanto, todos los sistemas de control deben contener al menos un instrumento de medición. A su vez, el bloque correspondiente al instrumento de medición se encuentra formado por dos elementos fundamentales, un sensor para la adquisición de la variable sujeta a control y, un transductor para convertir la variable sensada en una señal eléctrica, a fin de facilitar su manejo.

El controlador interpreta el cambio en la variable controlada, suministrada por el dispositivo de medición, y realiza la acción controladora necesaria para mantener el balance deseado dentro del proceso.

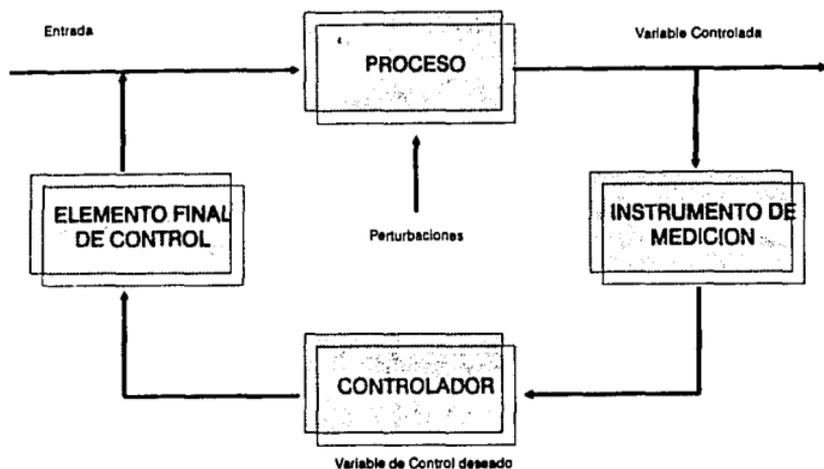


Fig. 3.1 Sistema de Control Automático

En muchas aplicaciones, es un microprocesador el que controla variables mecánicas tales como la posición o la orientación de un objeto. Para esto se requiere de un dispositivo de salida, generalmente llamado actuador, capaz de traducir señales de control en movimientos mecánicos. El elemento final de control, es el mecanismo que varía las condiciones de la variable a controlar en respuesta a una señal del controlador. Este elemento puede ser una válvula, un apagador, una bomba o un motor. El elemento final de control cierra la malla, proporcionando al proceso la variable ya regulada para ser sensada nuevamente, comenzando un nuevo ciclo.

Las **variables o parámetros** físicas que pueden encontrarse en los diversos procesos industriales suelen ser incompatibles entre sí, con respecto al medio de control o debido a su naturaleza, estas variables pueden ser hidráulicas, mecánicas, neumáticas o eléctricas. La forma más sencilla de interpretar y manejar una variable física, consiste en representarla como una señal eléctrica. Así, una variación en una variable física (cambio de posición, de presión, de temperatura, flujo etc.) puede traducirse en un cambio de voltaje. El dispositivo encargado de realizar este cambio de señales es conocido como transductor eléctrico, precisamente por convertir una señal de naturaleza diversa, en una señal eléctrica.

En general, un transductor se define como un dispositivo alimentado por la energía y que dona energía, usualmente en otra forma, a un segundo sistema. Dentro de los transductores se pueden considerar dos tipos de dispositivos: **Sensores y Actuadores**.

Los sensores, como se explicó anteriormente, se encargan de realizar funciones de monitoreo, es decir, que se encargan de adquirir información sobre las condiciones en las que se encuentra la variable bajo control, expresándola generalmente a través de algún cambio en su composición física. Cabe hacer notar que, estos instrumentos simplemente indican la condición del ambiente y sus lecturas no ofrecen ninguna función de control.

Por otra parte, el actuador es el elemento del esquema de control (figura 3.1) que recibe la señal del controlador para regular la potencia suministrada al elemento final de control. Es decir, es el elemento (o elementos) del esquema de control que se encarga de ejecutar las acciones mandadas por el controlador.

Es necesario aclarar que debido a que las señales que provienen del controlador son de baja potencia, no pueden activar a los elementos finales de control, es por esto que el actuador interviene como interface entre estos y el controlador.

3.2. DESCRIPCION GENERAL DEL SISTEMA DE CONTROL

La concepción inicial del Sistema surgió de la necesidad de lograr la automatización de diferentes esquemas que permitan tener el control sobre ciertas variables que conforman un proceso industrial determinado.

Para lograr este propósito hemos concebido el desarrollo de un sistema conformado por una **Unidad de Procesamiento Central (UPC)** que a través de una **Unidad de Comunicación Central (UCC)** podrá comunicarse con diversos módulos de control de variables independientes entre sí, llamados **Unidades de Procesamiento Local (UPL)**.

Por lo que se refiere a la definición de la topología del sistema a usar, se planteó en un principio la posibilidad de emplear una topología de red, tipo punto a punto, sin embargo por la necesidad de abatir costos en el cableado y el hardware asociado, así como otros factores como el ruido, se optó por la topología tipo multipunto.

La primer opción, como se muestra en la figura 3.2, presenta la desventaja de poseer un protocolo de comunicación muy sencillo, así como mayor rapidez entre otros. En cambio, la desventaja que presenta es de tener mayor susceptibilidad al ruido, mayor costo en el cableado y mayor hardware de interconexión.

La segunda opción presenta la ventaja de tener un menor costo en el cableado, menor susceptibilidad al ruido, entre otras. La desventaja que se tiene es el de emplear un protocolo de comunicación mas sofisticado. (Ver figura 3.3.)

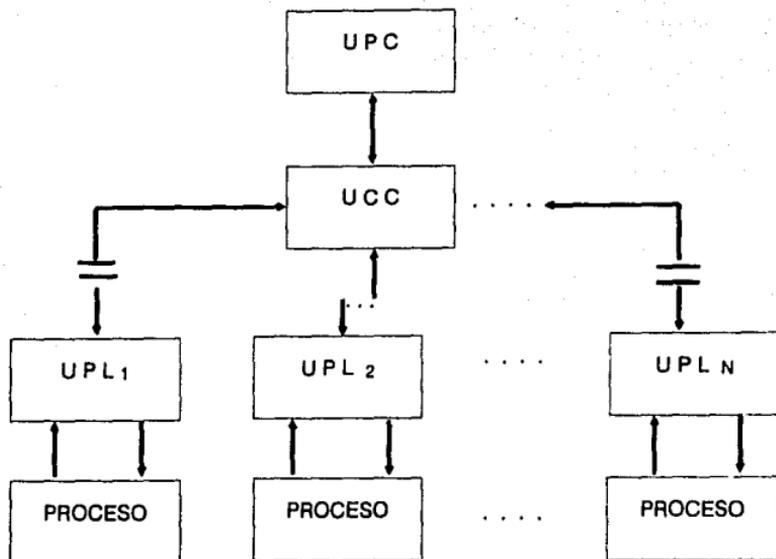


Fig. 3.2 Configuración Punto a Punto

Cada una de las Unidades de Procesamiento Local (UPL), se encargará del monitoreo y control de los diferentes parámetros o variables que intervienen intrínsecamente en un proceso industrial determinado.

Estas Unidades semejan el esquema de control automático descrito en los párrafos anteriores (figura 3.1), de tal forma que en estas unidades, por medio de la acción de ciertos transductores (instrumentos de medición) se obtiene la información de las condiciones que guarda la variable en un momento determinado, con el fin de determinar el tipo de acción que debe llevarse a cabo. A su vez, por medio de un dispositivo controlador (generalmente se utiliza un microprocesador), se interpretan los cambios sucedidos en las condiciones del estado de la variable o parámetro bajo estudio, de tal modo que este pueda tomar las decisiones sobre los ajustes necesarios que deben llevarse a cabo para que el valor del parámetro se mantenga en un entorno cercano a los puntos de calibración previamente establecidos y, de esta manera sostener el balance dentro del proceso industrial.

El controlador mediante un dispositivo de salida conocido como actuador, realiza las acciones de control en el proceso. Este dispositivo es capaz de traducir señales de control en movimientos mecánicos.

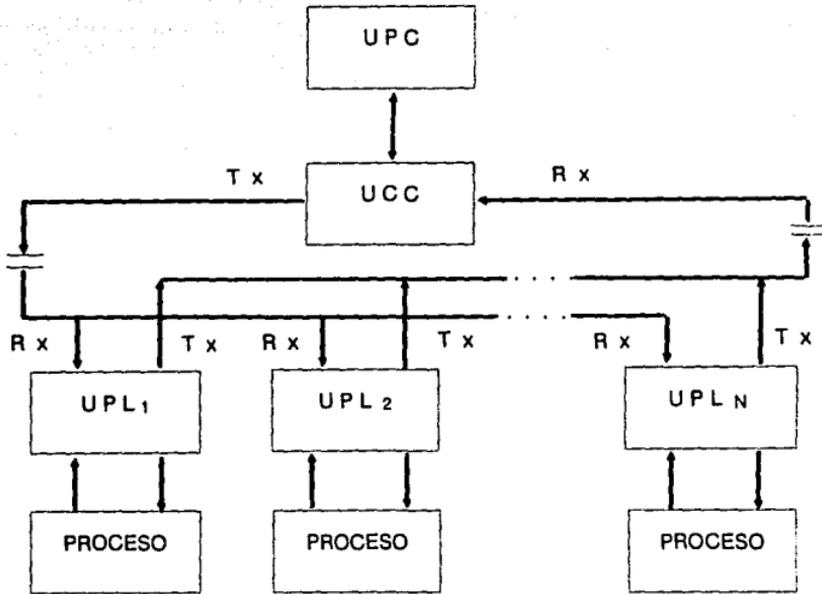


Fig. 3.3 Configuración Multipunto

Esencialmente, lo que se pretende con las Unidades de Procesamiento Local es de mantener las variables o parámetros que intervienen en un proceso industrial determinado, dentro del rango de operación especificado por la industria misma.

Es evidente que la selección de cada uno de los transductores y actuadores empleados en las Unidades de Procesamiento Local están en función directa del tipo de variables o parámetros que se requieran controlar.

Las variables que generalmente intervienen en el esquema de un proceso industrial, como se mencionó en el Capítulo I, son: presión, temperatura, flujo, niveles, velocidad de fluidos y movimientos de almacén, aunque pueden intervenir otro tipo de variables de acuerdo al tipo de industria o proceso que se esté manejando. Nuestro sistema se pretende que pueda manejar cualquier tipo de variable, esto es, que con la misma estructura de hardware (con el transductor y actuador adecuado para cada variable) en cada una de las Unidades de Procesamiento Local se pueda controlar o monitorear cualquier tipo de parámetro o variable. Esto se puede lograr teniendo el respaldo de programas de software de acuerdo al tipo de aplicación o variable que se este manejando en ese momento, es decir, utilizar un software especializado para cada una de las UPLs.

La Unidad de Procesamiento Central (UPC) estará constituida en general por una computadora personal (PC) compatible con IBM, por medio de la cual se permite tener un contacto directo entre el sistema de control y el usuario del mismo, de tal manera que este último pueda observar el comportamiento del proceso y sus diferentes etapas en un instante o período de tiempo determinados.

Por medio de la Unidad de Procesamiento Central (UPC) se podrán realizar o manejar reportes, ya sean gráficos o en forma numerica (análisis estadístico), a cerca de las condiciones que guardan en un período de tiempo determinado, los parámetros o variables que se encuentran bajo el control de las diferentes Unidades de Procesamiento Locales, de tal forma que permitirán analizar en detalle cada una de las etapas que constituyen el proceso industrial.

A través de la computadora personal el usuario del sistema, podrá determinar y fijar el período de adquisición de datos, de tal forma que se puedan obtener las condiciones (análisis estadístico, gráficas o reportes) que guarda el sistema en un cierto período de tiempo previamente establecido por el usuario. Cabe hacer mención de que como en la misma UPC, por medio de la computadora personal, se dará el almacenamiento masivo de la información proveniente de las diferentes Unidades de Procesamiento Local, se podrá obtener la información a cerca de las condiciones en las que se encontraba el sistema en un período de tiempo anterior al que se está trabajando actualmente.

Así mismo, por medio de esta misma unidad se podrán manipular y modificar en forma arbitraria los niveles máximos y mínimos de los diversos parámetros que intervienen en el proceso de control.

Para llevar a cabo la elaboración de los reportes numericos o gráficos para cada una de las UPLs en un periodo de tiempo determinado, es necesario establecer un esquema de comunicación, en el cual se de la transferencia de información entre la UPC y las diferentes UPLs a través de una Unidad de Comunicación Central, para lo cual se necesita un esquema en el que se pueda dar la comunicación o transferencia de información de dispositivo a dispositivo; además, que se puedan manejar aplicaciones en tiempo real, tales como aquellas que suceden en plantas industriales entre robots en varias estaciones de trabajo.

Cabe hacer notar que para llevar a cabo la sincronización entre todos los dispositivos que integran el sistema, es necesario que por medio de la Unidad de Procesamiento Local a través de la computadora personal (PC) se de un reseteo maestro "master reset", de tal manera que en el momento que se encienda el sistema, la primer acción del mismo sea dar este reseteo y de esta forma inicializar las funciones de cada uno de los dispositivos y en consecuencia del sistema en general. Es claro que, el usuario del sistema tiene el control de esta función, es decir, que en el momento que se quiera o necesite se pueda dar inicio a las funciones del sistema.

La forma en la que operará la Unidad de Comunicación Central (UCC), como se observa en la figura 3.3, puede definirse bajo dos formas distintas: La primera de ellas se

refiere a la forma de interconexión y función con la Unidad de Procesamiento Central (UPC). En este caso, como se definió en la topología, la Unidad de Comunicación Central se encontrará a muy poca distancia de la Unidad de Procesamiento Central, de tal manera que la forma de comunicación entre estas unidades será a corta distancia, de lo cual se deduce que el modo de comunicación a emplear entre ellas será en **paralelo**. En este modo de comunicación se presenta la ventaja de poder manejar datos en forma rápida y precisa, esto es, que en este tipo de comunicación se presenta menor inmunidad al ruido lo que permite reducir los posibles errores que surgieran durante la transmisión.

La segunda forma de operación de la UCC está determinada por su interconexión con las diferentes Unidades de Procesamiento Local (UPL). Para determinar la forma en la que se lleva el intercambio de información entre estas unidades, es necesario definir de acuerdo a la topología adoptada la forma de interconexión.

Debido a las características del sistema, la UCC se encontrará a una distancia considerable de las distintas Unidades de Procesamiento Local, por lo cual, para su interconexión se requiere de un cableado extenso y en consecuencia muy costoso. Por lo cual, se decidió emplear una comunicación del tipo **serial en configuración multipunto** (observar la figura 3.3), lo cual nos permite simplificar el cableado del sistema y en consecuencia nos permite tener una reducción muy significativa en cuanto al costo del mismo. Sin embargo, al adoptar esta forma de comunicación nos lleva a la elaboración de un software muy sofisticado.

En resumen se puede decir que el reto que representa nuestro diseño es el de lograr establecer un sistema de comunicación serial que potencialmente pueda enlazar información ya sea que provenga desde cualquiera de las diferentes Unidades Locales así como desde la misma Unidad de Procesamiento Central, es decir, lograr establecer un sistema de comunicación en el que se de la transferencia de información se realice en ambos sentidos pero además en forma simultánea. Por otro lado, se debe considerar que este sistema debe ser capaz de trabajar dentro de los ambientes de una oficina, industria y/o laboratorio, lo que significa que se está potencialmente expuesto al ruido.

3.3. PLANTEAMIENTO DE LAS RESTRICCIONES DEL SISTEMA

Las restricciones que fueron impuestas al sistema son las siguientes:

a) Debido a que la distancia entre la UCC y cada una de las UPLs será de alguna manera considerable (en cada industria depende de las dimensiones de su planta), se debe considerar un estándar o interface de comunicación adecuado para que no se presenten errores en la transmisión ya sea por ruido o por algún otro factor externo.

- b) Debido a que el sistema operará en un ambiente industrial estará sometido a una gran cantidad de ruido electromagnético, por lo que el canal de comunicación será conformado por cable coaxial.
- c) El número de UPLs estará limitado por el protocolo o sistema de enlace de comunicación a emplear, es decir, que depende de la cantidad de unidades que pueda soportar o manejar el esquema de comunicación serial. De la misma manera, el número de Unidades de Procesamiento Local a usar depende de las necesidades y características específicas de cada industria, esto es, que este número de unidades es determinado por el tipo y la cantidad de variables o parámetros físicos a controlar en un proceso industrial determinado.
- d) Se considera que la velocidad de transmisión de información no debe ser muy alta, debido a que la relación entre el porcentaje de error y la velocidad es directamente proporcional.
- e) La etapa de acondicionamiento de señales no es contemplada en el diseño del Sistema, es decir, que se consideran las señales de entrada a los transductores y las señales de salida de los actuadores como limpias (ideales) y en consecuencia directamente disponibles para ser procesadas.

CAPITULO 4

4. DESCRIPCION DE LA UNIDAD DE PROCESO LOCAL (UPL)

4.1. INTRODUCCION

Existen algunas necesidades comunes en los sistemas computarizados orientados al control:

- Llevar la cuenta del tiempo real elapsado
- Mantener una cuenta de las transiciones en las señales
- Medir el ancho preciso de los pulsos de entrada
- Comunicarse con otros sistemas o gente
- Monitoreo cercano de eventos externos asíncronos

Hasta ahora, los sistemas basados en microprocesador necesitaban dispositivos periféricos como temporizador/contadores, USARTs o controladores de interrupción para dar cabida a estas necesidades. Gracias a los microcontroladores, es decir a las microcomputadoras de una sola pastilla destinadas a sistemas de control industrial, las necesidades de dispositivos periféricos externos pueden ser encontradas en un sólo dispositivo

Los microcontroladores se están convirtiendo rápidamente en la columna vertebral de los sistemas de cómputo en base al silicio. Desde un punto de vista técnico, el atributo más significativo que diferencia un microcontrolador de un microprocesador, además de incluir RAM y ROM, es el manejo de entradas y salidas (E/S). En general el manejo de E/S es una parte íntima de la arquitectura de los microcontroladores. El set de instrucciones y la arquitectura de un microcontrolador permite que el CPU controle directamente las facultades de E/S en el dispositivo. Esto contrasta directamente con el microprocesador donde las E/S son en esencia un "mar" de direcciones y depende del diseñador del hardware incluir algún tipo de hardware de E/S en este "mar" de E/S. Debe ser obvio que simplemente agregando ROM y RAM a un microprocesador no creará un microcontrolador.

Este íntimo contacto con las E/S le dan al microcontrolador una notable ventaja sobre el microprocesador, en aplicaciones donde son intensivas las E/S. Los microcontroladores pueden probar, fijar, complementar o borrar terminales de E/S con mayor rapidez que un microprocesador, y también pueden realizar decisiones basándose en el estado de otras facultades del hardware, tales como temporizador/contadores, con

velocidad similar. Esta integración de E/S, en tanto el hardware como el software, hace del microcontrolador un dispositivo "ideal" para muchos tipos de instrumentación inteligente.

Acerca de la problemática a la que se enfrenta el control digital en tiempo real y en función del sistema, se puede mencionar lo siguiente:

a) La elección de los sensores.

Se debe considerar el tipo de variable a medir, la precisión y linealidad del sensor.

b) La elección de los convertidores.

Se deben tomar en cuenta factores como velocidad de conversión, precisión o número de dígitos y la linealidad.

c) La elección de la computadora.

Ya que de ésta dependen factores tan importantes como la velocidad de procesamiento del algoritmo de control. Aunado a esto están la capacidad de memoria para almacenar por un lado el programa en sí y las subrutinas de servicio; y por otro, las variables involucradas en el algoritmo y su evolución, si es necesario.

d) La elección del período de muestreo.

El cual depende de la planta y de los objetivos del control; y afecta en el cálculo de la señal de control.

4.2. LA UNIDAD DE PROCESO LOCAL (UPL)

La UPL se diseñó alrededor del 8031. Microcomputadora de una sola pastilla, de la familia de microcontroladores MCS-51 de Intel (ver apéndice A). Este dispositivo no sólo cumple con las necesidades comunes de los sistemas computarizados orientados al control, sino que además cuenta con un procesador booleano. Este procesador booleano es el aspecto más importante de la arquitectura de la familia MCS-51, ya que provee un soporte de hardware completo para el manejo de un bit individual. Hay que recordar que los controladores por lo general tienen que ver sobre el estado de un bit y no un byte: en el mundo práctico, los contactos de un interruptor sólo se pueden abrir o cerrar, los indicadores sólo se encienden o se apagan, los motores se encienden o se apagan, etc.

¿Qué se logra con un procesador booleano?. Cualitativamente nada. Las mismas facultades se pueden (y por lo general se han implementado) implementar en otras

máquinas usando complicadas secuencias de otras operaciones básicas. Cualquier CPU puede resolver cualquier problema, dándole el tiempo suficiente.

Cuantitativamente las diferencias entre una solución dada con el 8031 y aquellas requeridas por las arquitecturas anteriores, son numerosas. Lo que se logra con el 8031 son soluciones más rápidas, limpias y más económicas a aplicaciones en base a un microcontrolador.

El combinar instrucciones para el procesamiento de bytes y de bits individuales, produce gran poderío. Una aplicación basada en MCS-51 será:

- más sencilla de escribir, ya que la arquitectura se correlaciona más íntimamente con el problema a resolver
- más fácil de corregir ya que más instrucciones individuales no tienen efectos secundarios no deseados o inesperados
- más eficiencia en el manejo de bytes, debido al direccionamiento directo de bits y un ruteo relativo al contador del programa
- una corrida más rápida ya que se tienen que buscar menos bytes por instrucción y se procesan menos saltos condicionados
- menor costo por el alto nivel de integración de un sólo componente

4.3. DESCRIPCION DEL HARDWARE DEL PROTOTIPO

Es necesario aclarar que, siendo el objetivo del presente trabajo mostrar un sistema de control general, únicamente presentamos la parte de computadora de un sistema de control, por lo que al referirnos a la UPL nos referimos a dicha parte y su interface para el manejo del protocolo HDLC/SDLC.

El hardware de la UPL prototipo presenta sencillez, versatilidad, flexibilidad y gran poderío. Debido a la característica general de la UPL se probaron 4 diferentes prototipos de una configuración computadora/controlador basada en el 8031 (ver la figura 4.1).

Es necesario recordar que el hardware interno de un microcontrolador de una sola pastilla, reemplaza toda la lógica externa que se emplea para control, temporización, interrupciones, etc. En estas cuatro configuraciones sólo es necesario agregar teclado, display, relevadores, interruptores y las E/S definidas por el usuario (como convertidores D/A y A/D) para obtener el producto final.

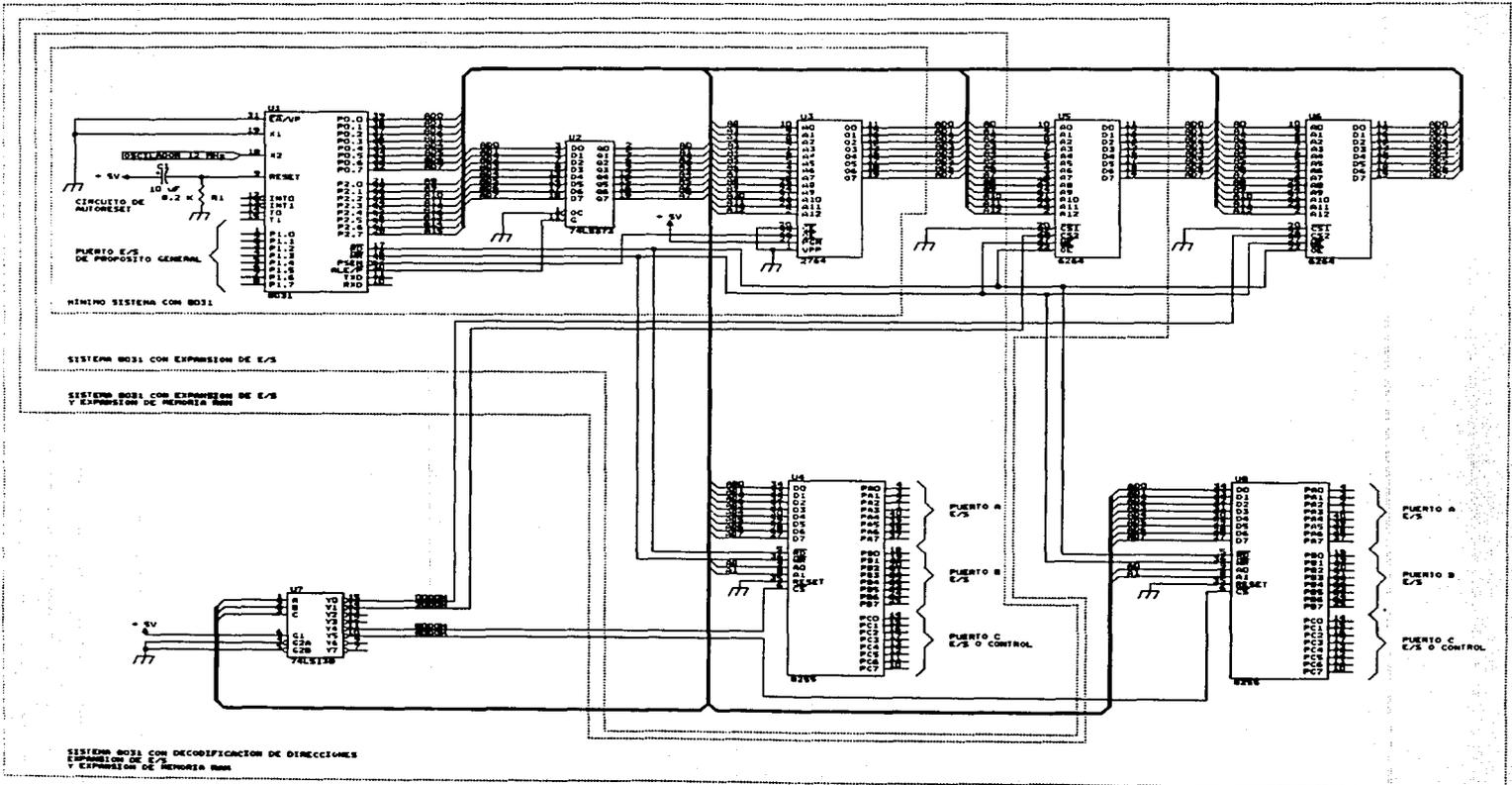


Fig.4.1 Alternativas para el Hardware de la UPL

La figura 4.1a muestra el sistema básico con 8031: un 8031, una EPROM 2764 para guardar 8K bytes de programa y un latch 74LS373 para demultiplexar el bus de direcciones/datos. El sistema cuenta con 128 bytes de RAM, un puerto paralelo de E/S bidireccional, un puerto serial bidireccional, dos temporizador/contador y dos entradas para interrupciones externas. Con un oscilador de 12 MHz, llega a ejecutar la mayoría de las instrucciones en 1 μ s (o bien 1 millón de instrucciones por segundo).

Un sólo puerto de E/S puede examinar un teclado matricial de 16 teclas. Con un bit de salida adicional, puede manejar un display de LCD inteligente de 2 rengiones y 20 columnas. Los bits de E/S restantes pueden manejar triacs o FETs de potencia para control de C.D. o C.A. Con un programa en la EPROM se tiene un poderoso controlador de tiempo real. Conectando el puerto serial a un convertidor de nivel RS-232C, se puede tener un puerto serial para control remoto o monitoreo en la computadora principal.

Si no se utilizarán el puerto serial, los temporizador/contador y las interrupciones externas, el 8031 puede emplear esos bits especiales como un segundo puerto paralelo de E/S (Puerto 3), por lo que el sistema básico del 8031 puede tener hasta 16 bits de E/S. Cada bit se puede probar, fijar y borrar individualmente bajo el control del software.

Si para la aplicación los dos puertos paralelos no son suficientes, la figura 4.1b muestra lo que se puede hacer para obtener tres más. Agregar un sólo PPI 8255. El puerto C del 8255 puede ser dispuesto para manejo de protocolo, con lo que se tiene la base para un convertidor de formato serial a paralelo o bien 24 bits de E/S adicionales para un controlador más grande. Notese que no se necesitan dispositivos intermedios entre el 8031 y el 8255.

Como el 8255 necesita de las líneas RD' y WR', el puerto 3 del 8031 no está del todo libre. Los 6 bits restantes pueden aún manejar E/S general o sus funciones específicas.

Si la aplicación requiere de más de 128 bytes de RAM, la figura 4.1c muestra como obtener 8K bytes de RAM, al agregar una SRAM 6264. Con ella se puede tener un convertidor de formato con "buffer" o un poderoso controlador serial-programable. Una RAM 62256 puede dar hasta 32K bytes sin problema alguno y aún sin dispositivos intermedios.

Finalmente si se necesita de RAM y E/S adicionales, la figura 4.1d muestra como conectar múltiples integrados de E/S. El decodificador 74LS138 genera las señales de "chip select" para los dispositivos contemplados en el mapa de memoria del 8031, con cada "select" cubriendo el rango de los 8K bytes. El sistema mostrado tiene 16K bytes de RAM y siete puertos bidireccionales de E/S.

El objetivo de presentar estas alternativas radica en que la parte de "computadora" del sistema de control no requiere de un hardware más elaborado. Para una unidad de un tamaño razonable, se invertirá más el tiempo de diseño del hardware en los dispositivos de E/S (v.gr.: convertidores A/D y D/A) que en los circuitos del 8031. Que es exactamente lo que se busca.

El mayor beneficio al diseñar con microcontroladores es la facilidad de agregar nuevas características al sistema, sólo con cambiar el programa, no cambiando las conexiones del circuito. Una nueva EPROM le puede dar una nueva personalidad al circuito.

La parte restante para implementar el sistema de control completo, necesita de la elección de los dispositivos adicionales que ayuden en esta tarea, y la elección de estos depende de las señales que se van a manipular; en general serán necesarios los sensores, los convertidores y los actuadores que ejecuten las órdenes del controlador para modificar alguna variable de la etapa del proceso al cual está destinada la UPL y conseguir así, los objetivos del control. Se debe señalar que algunos dispositivos requieren de otros para que funcionen como sea necesario; así un convertidor A/D necesitará de un dispositivo de S/H ("sample and hold") que le ayude a efectuar la conversión, y un sensor requerirá de otros dispositivos para que traten (acondicionen) la señal entregada por él si así se requiere. El modelo de la etapa del proceso de la planta y las señales de E/S, son la información básica y necesaria para el diseño de un esquema de control digital en una UPL.

Cada etapa del proceso impone el esquema de control de la UPL, y éste los dispositivos que lo constituyen. No puede hablarse de un esquema ideal a una etapa cualquiera, sin haber estudiado suficientemente dicha etapa y sus señales. Para diseñar las UPLs se deben estudiar profundamente, aparte de la etapa y sus señales, su ubicación (medio ambiente) para saber si existen señales que interfieran y destruyan las órdenes de control (ver el capítulo 8, que trata el tema de ruido).

4.4. CONSIDERACIONES DEL SOFTWARE PARA LA UPL

La computadora del sistema de control, contiene parte del esquema de control. Y es un programa almacenado en memoria, que es quien va a dirigir las acciones de control. Dicho programa consta de las etapas necesarias para cumplir con los objetivos de control; y de una manera sencilla, este procedimiento o algoritmo de control contempla los siguientes pasos:

- a) Inicializar todas las variables necesarias del algoritmo.
- b) Leer la señal de referencia, ya sea de memoria o de algún puerto de entrada.
- c) Medir la señal de salida de algún puerto de entrada. Lo cual implica considerar el tiempo de sensado y conversión, y acondicionamiento si se requiere.
- d) Calcular la señal de error con algún criterio adecuado.
- e) Calcular la señal de control mediante el algoritmo del controlador.

f) Convertir a analógica la señal de control.

g) Esperar una señal (período de muestreo) para tomar otra muestra de las señales de referencia y la salida de algún puerto de entrada, para procesarlas nuevamente.

h) Volver a b).

La manera en que se lleve a cabo la secuencia anterior dependerá de la etapa del proceso, de sus señales, de los objetivos del control y de los dispositivos que componen el esquema, así como del medio ambiente.

4.4.1. Conceptos importantes que debe considerar el programador de la UPL.

Para aplicaciones de tareas en tiempo real, el programador debe considerar tomar control de dispositivos de E/S, memoria física, prioridades de tareas, procesamiento de excepciones e integridad de datos.

a) Control sobre dispositivos de E/S.

El programador debe controlar los dispositivos de E/S para monitorear y controlar los periféricos de la UPL. Para ello se debe contemplar el manejo de interrupciones.

b) Control sobre memoria física.

El control sobre la memoria física es necesaria para evitar "swapping" (intercambios automáticos de información entre la memoria y dispositivos de almacenamiento masivos) y controlar el flujo de datos. El programador debe inhibir el swapping e identificar y controlar las áreas de memoria física que sirven para almacenar datos (buffer) de los dispositivos de E/S. El uso de memoria compartida y operaciones asíncronas son necesarias para un trabajo eficiente.

c) Control sobre prioridades.

Por otro lado, el programador debe tener control de las prioridades de las tareas, para asegurar que las tareas de tiempo real no serán detenidas por causa de otras tareas de menor importancia. Los mecanismos de control de prioridades permiten al programador decidir y controlar cuál tarea "tomará" el procesador. En un sistema de tiempo real, a cada tarea se le garantiza que en el peor de los retardos, tendrá tiempo suficiente para completar su trabajo.

d) Programación de excepciones.

Debido a que la naturaleza de los sistemas en tiempo real no están a salvo de condiciones de error, es necesario contar con un nivel adicional de control asíncrono por el programador (esto se conoce como procesamiento de excepciones).

En tareas de control interdependientes, la pérdida de control puede resultar fatal y además que la aplicación no concluya con éxito. Esto significa que el programador debe verificar todas las posibles condiciones de error relacionadas con la operación del programa, tales como errores de E/S y errores de "overflow/underflow" aritméticos.

Además, es necesario "atrapar" otro tipo de errores no concernientes a las operaciones del programa, como son errores en la línea de comunicaciones, errores en la memoria y fallas de energía.

e) Integridad de datos.

La integridad de datos es otro factor sobre el cual el programador debe mantener un riguroso control. Una falla de energía es el único ejemplo de un evento externo que provoca un impacto en la integridad de datos. Por lo tanto, para procesos en tiempo real, es necesario un mecanismo para determinar que datos han sido perdidos y cuales permanecen aún válidos.

f) Despachador y colas.

Dos aspectos críticos en un medio de tiempo real son, la naturaleza aleatoria de los eventos físicos y la ocurrencia simultánea de procesos físicos. Debido a esto el manejo de interrupciones y propiedades de multitareas son los principales atributos del software para sistemas de tiempo real. Es decir, el corazón del software debe ser el manejo de multitareas: el despachador.

El resto del software, se establece alrededor de este núcleo y sirve a las demandas específicas del medio de aplicación.

En particular, las listas o colas y sus manejadores que están alrededor del despachador, son construidos para comunicarse con los diferentes recursos físicos soportados por el software del sistema. Es decir, una cola puede contener aquellas tareas que están listas para ser ejecutadas en el procesador (v.gr.: procesos o programas), otra cola puede tener tareas que esperan accesos al hardware de E/S, y otra cola puede contener tareas que esperan por algún evento específico a ocurrir.

En cualquier software de un sistema multitareas, el despachador usa las colas como entrada. Su salida, por otro lado, es una simple tarea que ha sido activada y permitida para ejecutarse en la UPL. El algoritmo de despachador define en gran parte al software del sistema.

En un sistema, el despachador simplemente selecciona la primera tarea a ser ejecutada, permitiéndole que corra hasta que termine o hasta que consuma un período específico de tiempo.

En un sistema más sofisticado, el despachador puede seleccionar tareas en el esquema Round-Robin (ver figura 4.2), y permite a cada una un tiempo específico de

proceso. Una vez que la tarea excede el tiempo permitido, es colocada al final de la cola y forzada a esperar la ejecución de todas las tareas que vayan adelante de ella.

El esquema Round-Robin con periodos de ejecución iguales, es adecuado si cada tarea no es más importante que cualquiera de las otras. Sin embargo, si se requiere que alguna tarea tenga mayor prioridad, es necesario un algoritmo de despachador más sofisticado, es decir, un algoritmo que reconozca que unas tareas son más importantes pero, que tampoco excluya de la UPL las tareas de baja prioridad.

Una solución es el uso de varias colas, donde el lapso de tiempo de ejecución este relacionado a la prioridad de los elementos de la cola.

En este caso, el despachador deberá permitir a todas las tareas de cada cola de prioridad diferente, el acceso a ejecución, siempre que a las de menor prioridad les asigne un tiempo de ejecución menor.

Un refinamiento adicional permitirá que las tareas de mayor prioridad suspendan a la tarea de ejecución. Esta técnica conocida como pre-vaaciado, es una importante característica para sistemas de tiempo real (ver figura 4.3).

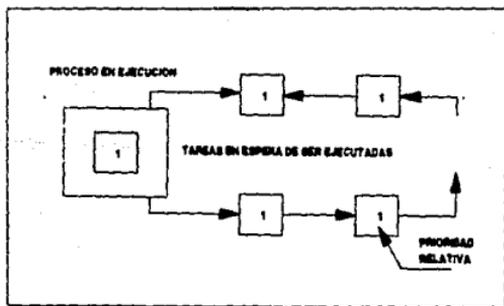


Fig. 4.2 Esquema Round-Robin

4.4.2. Consideraciones del software al nivel ensamblador.

El mayor problema en el diseño del software para la UPL, es armonizar una gran variedad de dispositivos de E/S asíncronos.

El primer análisis importante por hacer, es determinar cuales tareas deben completarse a tiempo, cuanto llevarán (en hardware y/o software), y las prioridades relativas de cada una.

Es decir, no hay que perder de vista que en diseños en máquinas que no cuentan con multitareas, cada tarea se irá ejecutando secuencialmente y es responsabilidad del software desarrollado (o del programador), asignarle a cada una un tiempo específico de procesador, el cual garantice el éxito de cada tarea.

Para esto es importante determinar por un lado, la frecuencia con que se debe activar

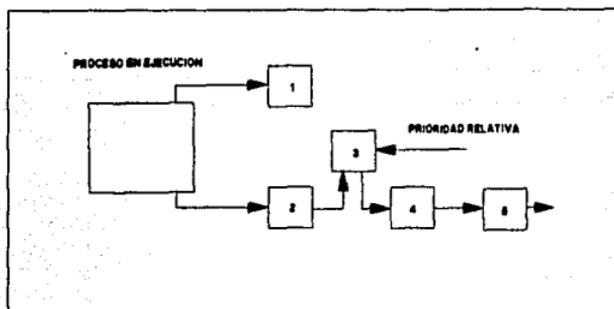


Fig. 4.3 Técnica de Pre-Vaciado

cada rutina de servicio y por otro, el tiempo que debe tomar cada rutina en ejecutarse. Una proposición útil, considera el uso de un programa "supervisor"; el cual tiene la responsabilidad de decidir cual función se debe activar. Aquí, cada función es implementada como una o más subrutinas; el supervisor procesa una tabla que especifica cual subrutina debe activar, sin embargo, es importante notar que cada subrutina no debe tardar un tiempo mayor que el período más corto en el sistema.

También existen otros problemas, por ejemplo, todas las subrutinas son independientes, aún en sus interfases de E/S, es por esto que deben ser capaces de comunicarse entre ellas, es decir, cada subrutina puede ser productor de datos de un tipo y un consumidor de datos de otro tipo. Entre el productor y el consumidor debe existir un acuerdo necesario del formato de datos, valores disponibles y localidades. Todas estas convenciones de liga deben ser definidas para cada subrutina de tal manera que cada una de ellas pueda ser escrita para conformar los acuerdos comunes.

Otro análisis importante radica en el empleo de interrupciones o poleos, para permitir la E/S de eventos múltiples asíncronos, los cuales ocurren con aparente simultaneidad.

Al emplear poleos cada tarea se diseña para ejecutarse en un período de tiempo máximo, el cual es menor que el tiempo requerido por la frecuencia más alta de una subrutina de E/S. Al final de cada ejecución de subrutina, el control es regresado al supervisor, en este punto el supervisor decide en que momento activar una de las subrutinas dependientes del tiempo o permite que se active otra de las subrutinas independientes.

En aplicaciones simples en las que la UPL no está saturada de trabajo, este esquema trabaja muy bien. Sin embargo, en caso contrario, las únicas alternativas posibles son las interrupciones.

Cuando las interrupciones son usadas para transferir datos ya sea de entrada o de salida, deben manejarse las siguientes tareas por el programa de interrupción:

- Salvar el contenido de todos los registros que se usarán durante la interrupción.
- Si existe más de una interrupción, identificar cuál necesita el servicio e invocar el software apropiado.
- Reconocer la interrupción de tal manera que el dispositivo de interrupción libere la terminal de interrupción.
- Efectuar las transferencias de datos.
- Verificar si ésta es la última serie de interrupciones. Si este es el caso, informar al programa principal, por ejemplo, que todo el registro ha sido leído.
- Restaurar los registros y las banderas a su estado original (antes de la interrupción).
- Habilitar interrupciones. Rehabilitar las interrupciones que fueron detectadas y detenidas para prevenir interferencias y luego regresar al programa interrumpido.

CAPITULO 5

5. UNIDAD DE COMUNICACION CENTRAL (UCC)

5.1. INTRODUCCION

Como se mencionó anteriormente, para poder llevar a cabo la transferencia o enlace de información entre la Unidad de Procesamiento Central (UPC) y las diferentes Unidades Locales (UPL), es necesario definir una Unidad adicional, esta unidad es llamada Unidad de Comunicación Central (UCC).

La Unidad de Comunicación Central (UCC), funciona como una interface entra la UPC y las diferentes UPLs, es decir, que se encarga de controlar el intercambio de información entre cada una de las Unidades, como se puede observar en la figura 3.3 del capítulo 3, de tal manera que la forma en que opera la UCC puede ser dividida en dos partes: La primera de ellas está destinada a describir el esquema de comunicación para poder llevar a cabo el intercambio de información entre la UPC y UCC, para lo cual se destinará el siguiente Capítulo.

La segunda parte está destinada a definir el esquema de comunicación a emplear para llevar a cabo el intercambio de información entre la UCC con las diferentes UPLs, para lo cual se destinará el presente Capítulo.

5.2. DESCRIPCION DE LA FORMA EN QUE SE LLEVA A CABO LA COMUNICACION ENTRE LA UCC Y LAS DIFERENTES UPLs

Para poder describir la forma en la cual se realiza la comunicación o el enlace de información entre la UCC y las distintas UPLs, es necesario tratar de definir las características y requerimientos de nuestro sistema.

Como se mencionó anteriormente, el tipo de comunicación a emplear en el sistema es la serial, debido a que nos trae consigo una reducción en cuanto al hardware del sistema, sin embargo nos lleva a la elaboración de un software muy sofisticado, por lo cual, es necesario definir el tipo de norma o protocolo de comunicación a utilizar para llevar a cabo la transferencia de información hacia o desde las diferentes unidades locales.

Los protocolos diseñados para los microprocesadores de 8 bits funcionan correctamente con la CPU de 16 bits, pero los requisitos del sistema pueden exigir elementos

con un mayor rendimiento. En ningún campo de las interfaces se hace esto tan evidente como en las comunicaciones serie. Cuando se desarrolló el microprocesador de 8 bits, las comunicaciones eran relativamente sencillas, la CPU daba soporte sólo a un número reducido de terminales, con señales asíncronas serie con un estándar RS-232C. Este tipo de comunicaciones tan sencillas adolece de una serie de inconvenientes:

- a) El porcentaje de bits de control del formato es alto (un bit de arranque y uno, o un bit y medio de parada para cada 5, 6, 7 u 8 bits de datos).
- b) No existe control de enlace de datos (DLC), de forma que se puede crear, por ejemplo, un sistema de bus serie con una jerarquía sencilla (es más fácil construir un sistema jerárquico que uno no jerárquico "tipo red").
- c) La detección de errores es mínima (restringida a un único bit de paridad opcional por palabra de datos) y la corrección no existe.
- d) Las velocidades estándar son bajas (normalmente, hasta 9 600 baudios).

Aunque se contaban con las comunicaciones síncronas, que permiten obtener mayores velocidades y menos porcentajes de bits de control en el formato, este tipo de comunicaciones, soportadas por interfaces serie ofrecidos por la mayoría de los fabricantes (8251A, 6851 ACIA, o Z-80 SIO), sólo soportaban protocolos muy simples.

La situación de los grandes ordenadores centrales era de alguna forma similar a los dispositivos descritos con anterioridad, hasta la aparición de los grandes sistemas de IBM, lo cual trae consigo, el desarrollo de las normas de control de enlace de datos (DLC). A su vez, el desarrollo de estas normas dieron como resultado unos protocolos fijos para un sistema jerárquico de comunicaciones serie. Este tipo de interfaces DLC permiten configurar una red simple de microordenadores, con un único ordenador amo, y una serie de estaciones esclavas, comunicándose a través de un bus serie.

Generalmente, una estación maestra o de procesamiento central sitúa los controles de transmisión y de recepción de las estaciones remotas o esclavas a través de la adición de caracteres no imprimibles a los datos para ser transmitidos.

Un control de enlace de datos (DLC), algunas veces llamado protocolos de segundo nivel, es un conjunto de reglas que son seguidas por computadoras o terminales interfaceadas para ordenar la transferencia de información. Las funciones básicas de un DLC son :

- a) Establecer y terminar una conexión entre dos estaciones.
- b) Asegurar la integridad del mensaje a través de la detección de los errores, requerimiento para retransmisión, y reconocimiento positivo o negativo.
- c) Para identificar al transmisor y al receptor a través de puleo o de selección.

d) Sostener funciones especiales de control como el requerimiento de estado, reseteo de estación, reconocimiento de reseteo, comienzo, conocimiento de comienzo, y desconexión.

Una de las funciones clave de cualquier procedimiento de control de enlace de datos es la de proveer la transparencia de datos, lo cual trae consigo el poder distinguir bits de datos de los bits de control o el de poder decir cuando un bloque particular de datos empieza y cuando termina sin restricción del contenido del bloque de datos. Dos métodos generales para llevar a cabo la transparencia de datos son ampliamente usados. El primero, conocido como inserción de byte, y es usado en las comunicaciones síncronas binarias. El segundo es conocido como inserción de bit y es usado en el control de enlace de datos síncrono. En el caso de inserción de byte estos son un par de caracteres; en el caso de inserción de bit son una serie de 8 bits. Para asegurar la transparencia de datos el receptor elimina estos bytes o bits especiales, conservando el patrón de datos original.

5.3. OPERACION DEL CONTROL DE ENLACE DE DATOS (DLC)

En la mayoría de los sistemas de comunicación de datos, así como en nuestro sistema, un cierto número de terminales interactúan con uno o mas CPUs. Para que esto suceda, los dispositivos de transmisión y de recepción deben ser capaces de decir cada uno cosas como :

- Esta es la estación A transmitiendo
- Tengo un mensaje para la estación B
- ¿ Estas listo para recibir ?
- He recibido tu último mensaje
- He finalizado, ahora es tu turno para transmitir
- ¿ Tienes algo que mandarme ?

En la figura 5.1 se muestra un ejemplo típico en el cual se requiere de un protocolo de comunicación de tal manera que una computadora central pueda seleccionar una terminal remota, así que esta computadora central podría mandar o recibir datos de la terminal remota. La estación de control de comunicación entonces recibe toda la información digital desde su modem, la decodifica, y la manda esta a su terminal propia. En este sentido, la estación de control puede recibir información en grandes cantidades desde un diverso número de terminales a la vez.

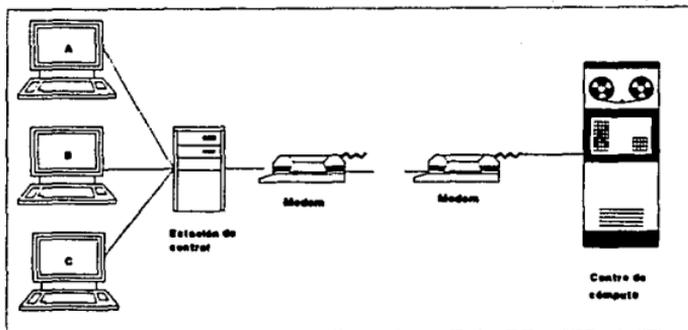


Fig. 5.1 Interacción en un Sistema de Cómputo

5.3.1. Tipos de Protocolos de Comunicación.

Se tienen muchos caminos para poder configurar un sistema de comunicación de datos y cercanamente a esto se tienen muchos tipos de protocolos. Básicamente, sin embargo, debido a los requerimientos y características de nuestro Sistema nos interesan ciertos tipos de protocolos o normas que pueden ser divididos en dos grupos:

- a) Protocolos controlados por byte (BCP).
- b) Protocolos orientados por bit (BOP).

Algunos ejemplos de Protocolos orientados por bit (BOP) son el Protocolo de enlace de datos síncrono (SDLC) desarrollado por IBM; el procedimiento de control de comunicación de datos avanzado (ADCCP) desarrollado por ANSI; el control de enlace de datos de alto nivel (HDLC) de ISO; y el control de enlace de datos de Burroughs (BDLC).

Por otro lado, algunos ejemplos de protocolos controlados por byte, incluyen el protocolo de mensajes digitales de comunicación de datos (DDCMP) y el protocolo de comunicaciones síncronas binarias (BISYNC) desarrollado por IBM.

5.3.1.1. Protocolo controlado por byte (BCP).

Los mensajes con el protocolo controlado por byte son transmitidos en unidades llamadas bloques. La muestra de protocolo que puede ser usado para mostrar como se lleva a cabo el enlace de datos es el desarrollado por la IBM y es llamado protocolo de comunicaciones síncronas binarias (BSC O BISYNC.)

En BCP, los caracteres especiales o secuencia de caracteres que controlan la operación de enlace consisten de tres diferentes tipos: caracteres de comunicación, caracteres de control de periféricos y caracteres gráficos.

Los caracteres de comunicación proveen la sincronización para determinar el comienzo de un campo particular en un bloque. Estos caracteres incluyen :

SOH Comienzo del guíador "Header".

STX Comienzo de texto.

ETX Fin de texto.

EOT Fin de transmisión.

ACK Conocimiento de mensaje.

NAK Desconocimiento de mensaje.

EOH Fin de guíador.

ITB Bloque de transmisión intermedio.

ETB Fin de bloque de transmisión.

DLE Salida del enlace de datos.

Los caracteres de control de periféricos se encargan de controlar la operación de dispositivos periféricos, mientras que los caracteres gráficos son generalmente caracteres de información que pueden ser vistos en una pantalla de tubo de rayos catódicos.

El formato de bloque para el protocolo de comunicaciones síncronas binarias se muestra en la figura 5.2 .

El protocolo BISYNC puede requerir la inserción de caracteres SYN a distintos intervalos de tiempo. Dos caracteres SYN son insertados para asegurar la sincronización propia para el comienzo del formato. Cada caracter SYN consiste de 8 bits. SOH es un caracter de control de comunicación que consiste de 8 bits, usado al comienzo de la secuencia de caracteres que constituyen información de rutina. El guíador (header) es el control de información prefijado en el texto del mensaje, por ejemplo, código de fuente o destino (dirección), prioridad o tipo de mensaje, conocimiento.

El Texto es el mensaje para ser transmitido; este es de longitud variable y podrían tener transparencia de datos. La porción del texto en un bloque de mensaje es identificado después de la identificación del carácter STX, ETX indica el final del mensaje.

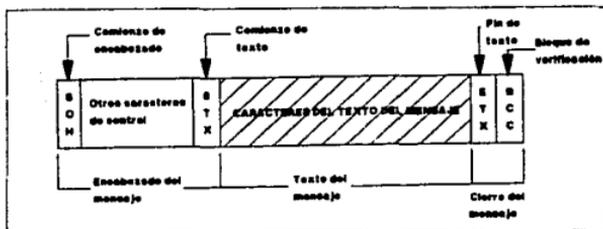


Fig. 5.2 Formato BYSYNC

Por último, el campo de chequeo de error se encarga de desarrollar la verificación del bloque de transmisión y es normalmente colocado al final del bloque. La longitud de los mensajes son divididos en una serie de bloques para transmisión. Cada bloque de texto, excepto el último, es seguido por el carácter ETB (fin de bloque de transmisión) o un carácter ITB (fin de bloque de transmisión intermedio). Si un error es detectado en cualquier bloque intermedio, ninguna acción puede ser tomada hasta que el carácter ETB es recibido; entonces todos los bloques intermedios deben ser retransmitidos.

5.3.1.2. Protocolos orientados por bit (BOP).

En general, los protocolos orientados por bit son menos dependientes de los caracteres de control y en lugar de esto se fían de la posición de los bits sin campos específicos o bloques de bits. Hay un formato de campo estándar de fijación de la longitud de los bits. Quizá el más ampliamente conocido de los protocolos orientados por bit es el desarrollado por IBM y es el llamado Protocolo de enlace de datos síncrono (SDLC).

Las normas de control de enlace de datos síncrono definen, con algún grado de flexibilidad, diferentes niveles de comunicación en un sistema que posea una CPU amo o estación primaria y una serie de dispositivos esclavos o estaciones secundarias, comunicándose a través de un bus, punto a punto, o en una configuración multipunto. El modo de transmisión puede ser semidúplex (transferencia de información se realiza en ambos sentidos, pero primero una estación transmite y después la otra estación responde) o dúplex integral (la transferencia de información se realiza en ambos sentidos y en forma simultánea) como se muestra en la figura 5.3.

El protocolo SDLC esta orientado al bit, lo que significa que no se impone al usuario ninguna restricción sobre el formato de transmisión de los datos (por ejemplo en caracteres de 8 bits). La información transmitida utilizando SDLC puede usar cualquier código o formato, y se trata simplemente como un conjunto de bits.

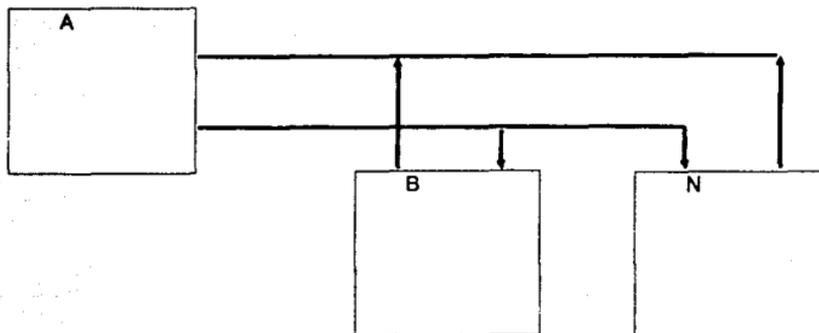


Fig. 5.3 Configuración Multipunto en Full-Duplex

En SDLC la información es mandada por tramas o "frames". Cada una de estas tramas contiene una serie de campos los cuales tienen funciones muy específicas. En la figura 5.4 se muestra el formato que deben tener las tramas SDLC.

BANDERA DE INICIO	CAMPO DIRECCION	CAMPO CONTROL	CAMPO INFORMACION	VERIFICACION DE MARCO	BANDERA DE TERMINACION
01111110	BITS 8	BITS 8	LONGITUD VARIABLE	BITS 16	01111110

Fig. 5.4 Tramas de SDLC

Los campos de indicadores de principio y final de la trama sirven para delimitar a la misma, y obviamente, deben ser únicos, de forma que no haya posibilidad de que una secuencia aleatoria de datos coincida casualmente con una secuencia de bits de los indicadores, causando entonces una indicación de "final de trama" fantasma. Hay dos formas de asegurar la unicidad de los indicadores, o pueden restringirse los datos para que las únicas secuencias de bits de datos no puedan imitar la secuencia de los indicadores; o bien debe modificarse el flujo de datos por parte de la circuitería de

transmisión, reestableciéndose su aspecto original por los circuitos de recepción, para asegurar la unicidad del indicador. Esta última técnica hará posible que el sistema transporte datos en cualquier formato, sin restricción, es decir, que sea transparente. La ventaja de la transparencia de código es que puede utilizarse el mismo SDLC orientado al bit para transmitir datos en caracteres ASCII, programas en código de máquina binario, o en cualquier otro formato.

La forma en que funciona la circuitería de transmisión para preservar la unicidad del indicador es la siguiente :

El indicador de SDLC es la secuencia de bits 01111110, es decir, seis unos consecutivos delimitados por dos ceros. Siempre que la circuitería de transmisión encuentra cinco unos consecutivos en una secuencia de bits de datos, direcciones o control, automáticamente inserta un cero después del quinto uno de la secuencia. Este cero adicional asegura que la información de datos, control o dirección no puede tener nunca los seis unos consecutivos del indicador. Dado que siempre se inserta un cero tras cinco unos, sin tener en cuenta el nivel lógico del sexto bit, el receptor reconocerá los seis primeros bits de un indicador 011111 y después examinará el siguiente bit : si el siguiente bit es un "1" la secuencia de bits será un indicador, siguiendo los campos de dirección y control al último cero. Si el siguiente bit es un cero, es que se ha insertado en la secuencia de datos, direcciones o control por el transmisor, y debe quitarse antes de pasar los datos. La acción del transmisor se llama inserción de bit (bit stuffing) y la del receptor se llama eliminación de bit (bit stripping). Entre tramas, una estación SDLC transmitirá indicadores o caracteres de reposo (todos unos).

El campo de direcciones define al transmisor y al receptor. Además, los campos de dirección se definen para adaptarse a la necesidad de los protocolos SDLC amo-esclavo y se transmiten con el bit menos significativo en primer lugar. Así, una trama enviada por la CPU maestra o estación primaria a una estación esclava o estación secundaria tendrá la dirección de destino (la de la estación secundaria) en su campo de dirección, y un código de orden en su campo de control mientras que una mandada por una esclava contendrá la dirección de origen (la de la estación secundaria).

Además de posibilitar el acceso a estaciones secundarias independientes, el campo de dirección debe contener una dirección de "difusión", que permite a la estación primaria mandar datos a todas las secundarias de forma simultánea.

Por último, como se muestra en la figura 5.4, este campo está compuesto de 8 bits por lo que se deduce que se pueden tener 256 combinaciones diferentes de direcciones, de lo cual se infiere, que una misma estación primaria puede estar manejando o monitoreando 256 estaciones secundarias diferentes, sin embargo, por medio de varias herramientas de software se pueden llegar a direccionar un mayor número de estaciones que las establecidas por el campo propio de direcciones.

El campo de control consiste de 8 bits y determina la naturaleza del mensaje. El campo de control es usado por la estación primaria (la cual mantiene el control del enlace

de datos en todo momento) para decir a la estación direccionada que es lo que esta haciendo, como polear dispositivos, transferencia de datos, retransmisión de datos, entre otras cosas. La estación direccionada puede usar el campo de control para responder a la estación primaria, para decir que tramas ha recibido o cuales ha mandado. El campo de control es también usado tanto por la estación primaria como por la secundaria para la llevar la cuenta del número de tramas transmitidas o recibidas en cualquier momento.

El campo de información puede ser de cualquier longitud y puede estar compuesto por cualquier estructura de código (EBCDIC o ASCII por ejemplo) y lleva el texto del mensaje. Puede llevar mensajes de supervisión (requerimiento para retrasmisión, etc.) y computación de chequeo de trama.

La secuencia de comprobación de trama (FCS), definida como los últimos dieciséis bits de la trama que preceden inmediatamente a la secuencia del indicador de terminación, es una secuencia de comprobación de bloque estándar que abarca todos los bits de la trama entre los indicadores. El FCS es similar a una palabra de comprobación redundante cíclica (CRC).

El sistema de chequeo de redundancia cíclica (CRC), es el sistema de detección de errores más efectivo en los sistemas de comunicación con una mínima cantidad de hardware.

El cálculo de este CRC, se llevan a cabo por medio de secciones múltiples de registros de corrimiento los cuales alimentan a unas compuertas OR - exclusivas como se muestra en la figura 5.5. En cuanto a la lógica de las compuertas OR - exclusivas se tiene un cero a la salida, si y sólo si, ambas entradas son cero o ambas uno. Si las entradas son diferentes, la salida de esta compuerta es un uno. Por otra parte, de acuerdo a la colocación y cantidad de las compuertas OR - exclusivas varía por CRC-12, CRC-16, CRC-CCITT los cuales son los CRC más conocidos.

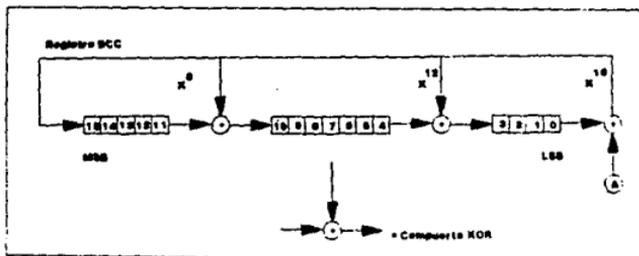


Fig. 5.5 Registro de Corrimiento del CRC

De acuerdo al protocolo que se pretende utilizar (SDLC), el CRC que se utiliza en este protocolo es el CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$), el cual se muestra en la figura 5.5. Como se puede observar, los términos de MSB y LSB usados en la figura se refieren a los bits más significativos y menos significativos del registro, el bit menos significativo (LSB) al final del registro es mandado primero, pero corresponde al término X^{16} del CRC.

Cuando se empieza a transmitir la información, el circuito de transmisión de CRC es inicializado a unos. Cuando cada uno de los bits de información es transmitido, estos mismos bits son aplicados al punto marcado "A" en la figura 5.5, es decir, que cada uno de los bits de información son colocados en el registro de corrimiento del CRC y se propagan a lo largo de las diferentes secciones de los registros de corrimiento de tal forma que al final de la transmisión de la información se genera un bloque de chequeo de caracteres (BCC). Este mismo procedimiento se implanta en el receptor, de tal manera que a medida que se capta la información se va generando un CRC de recepción, por lo que al final de la recepción de la información se tiene que chequear el BCC generado en la recepción y el BCC generado en la transmisión de tal manera que se pueda llegar a localizar un error en la información misma.

Como se mencionó anteriormente, el CRC utilizado en el protocolo SDLC es el CCITT. Este tipo de CRC cuando esta operando con caracteres de 8 bits, la acumulación de BCC es de 16 bits, por lo que la longitud de campo destinado para el chequeo de secuencia de trama es precisamente 16 bits.

Además, este CRC provee una detección de error del 99% en el caso de manejo de más de 12 bits.

5.3.2. Comparación entre los Protocolos.

Generalmente, el BCP se caracteriza por operar en comunicación semidúplex. Después que un formato BCP es mandado, este debe ser reconocido por el receptor. Esto reduce la velocidad de la comunicación.

Por otra parte, en el protocolo controlado por byte (BCP-BISYNC) hay muchos caracteres de control para la configuración de la línea, control de equipo periférico, y otros gráficos. Además, la transparencia de datos se lleva a cabo sólo a través del uso de mecanismos de escape.

En el protocolo orientado por bit (BOP-SDLC) hay dos o tres caracteres de control; por ejemplo, el control de equipo periféricos esta funcionando entre la computadora maestra y las terminales remotas, por lo que esto no es parte de la función del protocolo. Por otra parte, existe un formato estándar para la información, supervisión y tipos de tramas no secuenciales.

La transparencia de datos es provista a través de la inserción de un cero y su eliminación. Además, en BOP, el receptor no tiene que tener conocimiento de recepción de cada trama.

Otra de las características importantes del protocolo orientado al bit (BOP-SDLC) es que no hay restricciones en cuanto al código de datos, es decir, se puede transmitir información en cualquier tipo de formato o código. También el chequeo y la corrección de error, es disponible en una trama completa en el BOP.

5.4. DESCRIPCION DEL DISPOSITIVO DE INTERCONEXION ENTRE LA UCC Y LAS DIFERENTES UPLs

Una vez analizadas las características de los protocolos de comunicación, así como los requerimientos de nuestro Sistema, se llegó a la conclusión de que para llevar cabo el enlace o transferencia de información entre la Unidad de Comunicación Central (UCC) y las Unidades Locales (UPL), el Protocolo orientado por bit (BOP-SDLC) es el que más se ajusta a las necesidades del Sistema, debido a que este maneja la comunicación serial, que como ya se mencionó, este tipo de comunicación nos permite tener una cierta reducción en cuanto al cableado de interconexión entre las unidades y en consecuencia a esto una reducción en lo que se refiere al costo del Sistema. Además, útiliza la comunicación dúplex integral (transmisión en ambos sentidos en forma simultánea) lo cual trae consigo un aumento en cuanto a la velocidad de transmisión.

Finalmente, no existen restricciones en cuanto al código o formato de los datos a transmitir (código ASCII o EBCDIC), ni en cuanto a la cantidad de información a transmitir.

En resumen se puede concluir que el protocolo orientado por bit SDLC (control de enlace de datos síncrono) ha sido diseñado para poder manejar y controlar el flujo de información en un enlace de comunicación de datos.

Cabe hacer mención que el hardware de la UCC está determinada en forma semejante al hardware de las distintas UPLs, el cual fué propuesto y descrito en el Capítulo anterior, y el cual se observa en la figura 5.6. La única adición que se presenta es que debe ser definido el dispositivo de interconexión con las diferentes UPLs de tal manera que permita ser programado para soportar el protocolo de comunicación SDLC, como se puede observar en la figura 5.7 .

Hay dos restantes consideraciones en cuanto al hardware del sistema: La velocidad de transmisión de los datos y el nivel de las señales de transmisión.

La velocidad de transmisión de los datos puede ser expresada como bits por segundo y se refiere a la velocidad de transmisión a lo largo de la línea del enlace de datos.

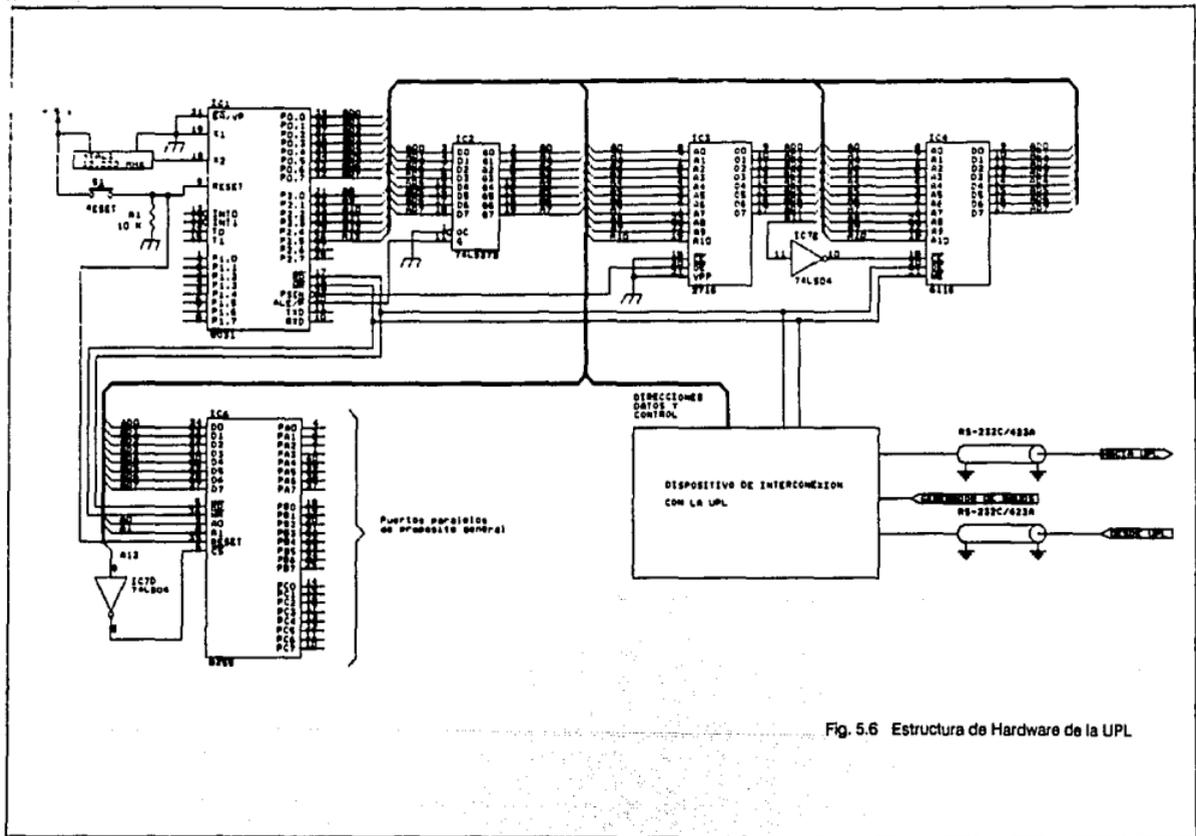


Fig. 5.6 Estructura de Hardware de la UPL

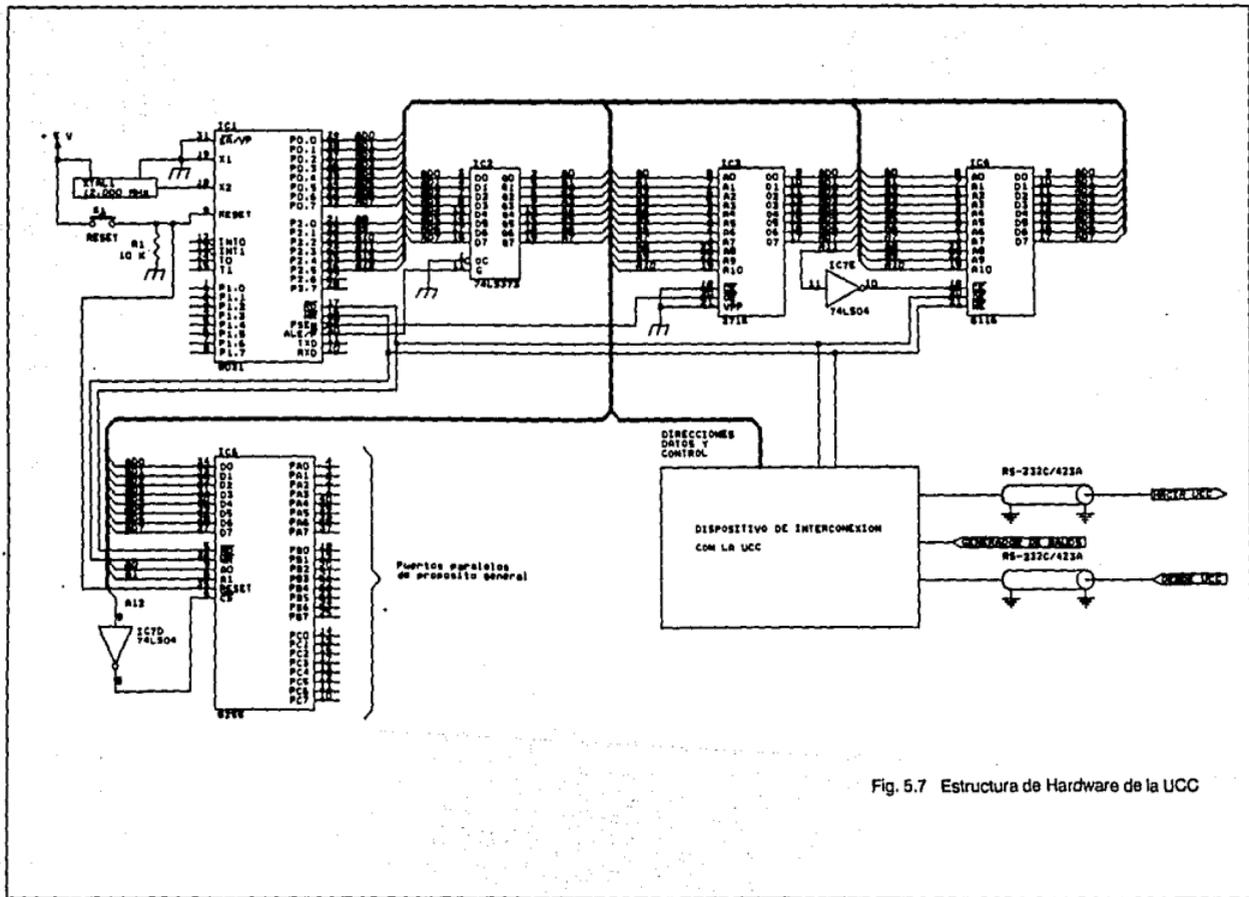


Fig. 5.7 Estructura de Hardware de la UCC

Por lo que se refiere a este punto, se utiliza un dispositivo especial de generador de velocidad de transferencia de datos (Motorola 14411), el cual cuenta con una gran variedad de frecuencias de trabajo por lo que de acuerdo a las necesidades y requerimientos del sistema se escogió el trabajar cuando menos con 9600 bits por segundo. (Ver figura 5.6).

Por otro lado, se deben tomar en cuenta los niveles de las señales de transmisión, que como se debe recordar, aunque los niveles TTL podrían ser usados para la transmisión no son apropiados para transportar señales a más de 3 a 6 metros (en nuestro esquema como se recordará, la UCC se encontrará a una distancia considerable de las diferentes UPLs). El problema se origina por el hecho de que solamente 2V separa un nivel lógico "0" o "1". Con solo 2V de inmunidad para los ruidos, la comunicación podría ser susceptible a interferencias, por ejemplo, de motores o de interruptores.

Un comité industrial acordó una interfase normalizada para resolver este problema, así como para sugerir normas para la industria, esta norma es llamada EIA RS-232C.

Los niveles de señal del RS-232C son bipolares y utilizan una tensión negativa entre -3 y -15V para representar un nivel lógico "1" y una tensión positiva de 3 a 15V para representar un nivel lógico "0". La región entre -3 y 3V ayuda a la inmunidad contra los ruidos y es una región "muerta". Generalmente este tipo de norma se aplica a sistemas en los cuales se requiere interconectar dos dispositivos que se encuentran a una distancia máxima aproximada de 15 metros de longitud entre ellos. Es claro, que existen sistemas en los cuales los dispositivos a interconectar se encuentran a distancia mayor de la descrita anteriormente, por lo que existen otras normas que pueden ser aplicadas para este propósito, por ejemplo, la norma EIA RS-423A la cual maneja distancias de hasta 600 metros de longitud, o como la EIA RS-422A que maneja distancias de hasta 1200 metros de longitud.

El canal de comunicación empleado es un cable coaxial de 2 hilos ya que el modo de operación es el de dúplex integral "full duplex", en el que la transmisión o enlace de información se realiza en ambos sentidos y en forma simultánea.

Para poder implantar físicamente este tipo de norma o protocolo de comunicación en nuestro Sistema se hizo necesario utilizar un dispositivo conocido como Controlador Serial de Multiprotocolo (MPSC- 8274), el cual fué desarrollado por INTEL, fabricado con tecnología HMOS, además de ser un dispositivo de 40 terminales. (Ver Apéndice B).

El dispositivo esta diseñado para poder interfacear líneas de comunicación de gran velocidad, usando protocolos como los asíncronos, el IBM BISYNC y el SDLC, a sistemas de microcomputadoras de la familia de INTEL como MCS-48, -85, -51, y con el controlador de DMA 8237.

5.4.1. Descripción de la Arquitectura del MPSC-8274.

Este controlador tiene una arquitectura muy simple, que permite una fácil implementación de tres diferentes tipos de protocolos (asíncrono, IBM BYSINC, SDLC) con poco software y hardware. Además, cuenta con dos canales independientes de transmisión / recepción.

El MPSC puede manejar varias opciones de interface con un microprocesador : Poleo, Transferencia de Bloques, e Interrupciones.

Dentro de las características del MPSC se puede mencionar que para poder llegar a tener transparencia en los datos, que como se mencionó en párrafos anteriores, se debe llegar a tener inserción y eliminación de un bit cero. En el 8274 la inserción y eliminación de bit se realiza en forma automática.

Otra de las características de este dispositivo es que por lo que se refiere al campo de control, este puede ser manejado por el dispositivo internamente de tal forma que para saber el estado que guarda en ese momento la información es necesario que el usuario pida dicha información al dispositivo .

Por último, cabe hacer mencionar que la velocidad de transmisión y recepción de datos es independiente de la velocidad a la que trabaja normalmente el dispositivo.

Como se puede observar en el Apéndice B, existe una terminal de reloj del dispositivo para el manejo de la velocidad de procesamiento de información interna del mismo y otra terminal que se encarga del manejo de la velocidad de transmisión y recepción de la información, es decir, que se maneja un reloj interno del dispositivo independiente de la velocidad de transmisión de datos a manejar.

En el caso del MPSC se tendrá una velocidad de procesamiento de información máxima de 4 MHz, por lo que se deduce que todo los dispositivos (el microcontrolador 8031 puede manejar velocidades de procesamiento, como se mencionó en el capítulo anterior, de hasta 12 MHz) estará trabajando a esta velocidad.

En el MPSC los comandos, parámetros y estado de la información son guardados en 21 registros, de los cuales, 8 registros son de escritura para cada canal, se tienen 2 registros de lectura para el canal A y 3 registros de lectura para el canal B.

Los registros para ambos canales estan diseñados de la siguiente forma:

WR0 - WR7 Registros de escritura

RR0 - RR2 Registros de lectura

El agrupamiento funcional de cada uno de los registros, así como la asignación de bits en los mismos, están configurados de tal manera que permiten simplificar y organizar el proceso de programación del dispositivo.

A continuación se listan las funciones generales asignadas a cada uno de los registros. En el Apéndice B se muestran de manera más detallada las características y funciones asignadas a cada uno de los registros.

RRO Estado del "buffer" de Transmisión / Recepción, estado de las interrupciones.

RR1 Estado de las condiciones especiales en la recepción.

RR2 Vector de interrupción modificado.

WR0 Registro apuntador, inicialización de CRC, comandos.

WR1 Interrupciones en la Transmisión / Recepción y definición del modo de transferencia de datos.

WR2 Vector de interrupción.

WR3 Control y parámetros de recepción.

WR4 Modos y parámetros de Transmisión / Recepción.

WR5 Control y parámetros de transmisión.

WR6 Campo de dirección del SDLC.

WR7 Bandera de Inicio / Terminación SDLC.

En el Apéndice B se muestran las características generales del MPSC-8274.

Como se puede observar, para poder interfacear el MPSC con una CPU es necesario utilizar las terminales de entrada A0, A1, CS', RD' y WR' del MPSC, lo cual permite lograr una comunicación con los registros internos del MPSC. La figura 5.8, muestra como pueden ser direccionados estos registros internos del MPSC.

Por lo que respecta al lado serial, existen dos canales completamente independientes: Canal A y el Canal B. Cada canal consiste de un bloque transmisor, un bloque receptor y la puesta de los registros de escritura / lectura los cuales son utilizados para inicializar el dispositivo. En adición, un bloque de control lógico provee terminales de interface con un modem. La interface serial lógica del Canal B es un espejo de la interface

lógica del Canal A, excepto que en el canal B las terminales RTS y SYNDET están unidas en una misma terminal.

CS'	A1	A0	OPERACION LECTURA	OPERACION ESCRITURA
0	0	0	Ch. A Data Read	Ch.A Data Write
0	0	0	Ch. A Status Read	Ch.A Command
0	0	0	Ch. B Data Read	Ch.B Data Write
0	0	0	Ch. B Status Read	Ch.B Command

Fig. 5.8 Direccionamiento de Registros Internos

En la figura 5.9 se muestra el diagrama de bloques de la estructura del MPSC para la transmisión y recepción de datos.

Como se puede observar, los datos transmitidos por la CPU son transferidos a un registro de cambio serial de 20 bits (esto es porque 20 bits son los necesarios para guardar 2 bytes que corresponden a los dos bytes de los caracteres de sincronización usados en el modo bisync). Los datos a transmitir en el registro de transmisión son trasladados hacia la línea de TXD (transmisión de datos) con dos bits de retraso. Estos bits de retraso son usados para sincronizar el reloj interno con el reloj interno de transmisión.

Además, el dato que se encuentra en el registro de cambio son presentados para la lógica de inserción del bit cero, la cual inserta un cero después de que se presentan cinco unos en forma consecutiva. En paralelo a esta actividad, el generador de CRC está computando el CRC de transmisión para ser transmitido al final de la transmisión.

Por lo que se refiere a la estructura de la recepción de los datos, como se puede observar, estos datos son pasados con un bit de retraso, antes de que estos sean presentados para la comparación de bandera. En el modo SDLC, la llegada de los datos siempre pasa a través del registro de sync donde el patrón de datos es continuamente monitoreado para la detección de unos consecutivos, para en su defecto, realizar la eliminación del bit cero. El dato entra al "buffer" de tres bits y al registro de cambio de recepción (8 bits), entonces el dato es transferido a la parte baja del registro de recepción

de datos en forma de FIFO (primero en entrar es el primero en salir) que está compuesto de 3 bytes. El dato es transferido a la parte de arriba del registro FIFO a razón del reloj del sistema (no de acuerdo al reloj de recepción).

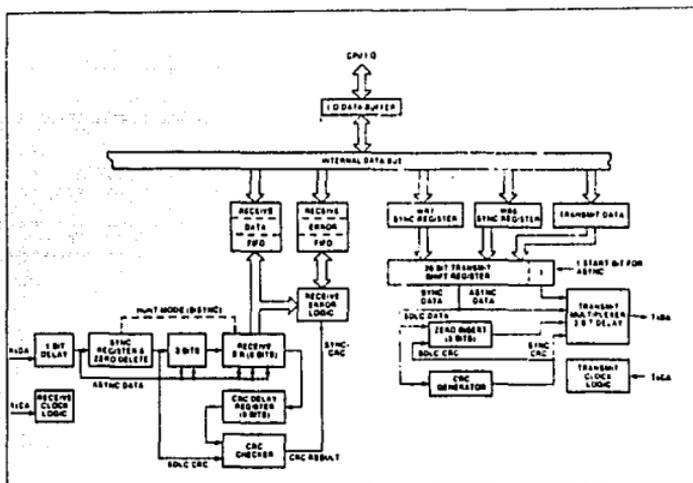


Fig. 5.9 Rutas en la Transmisión y Recepción

Se toma alrededor de tres ciclos de reloj para transferir el dato desde el registro serial de cambio hasta la parte de arriba del FIFO. Por otra parte el registro de recepción de error FIFO que esta compuesto de 3 bits, como se puede observar, transfiere cualquier condición de error que hubiese ocurrido durante la recepción de la trama.

Mientras todo esto esta sucediendo, el chegador de CRC esta checando el CRC correspondiente a los datos que están llegando. El CRC computado en el receptor es checado con los bytes de CRC relacionados con los datos de llegada y puede generarse un error debido a la condición de no-checa el CRC.

En la figura 5.10 se muestra la estructura de hardware completa de la Unidad de Comunicación Central (UCC) de tal forma que pueda ser conectado directamente (comunicación serial) a las diferentes Unidades de Procesamiento Local (UPLs) para llevar a cabo la transferencia o enlace de información proveniente de la Unidad de Procesamiento Central (UPC).

5.4.2. Descripción General de la Programación del MPSC-8274.

Por medio de los registros de escritura (WRO-WR7), como se mencionó anteriormente (Apéndice B), se debe inicializar el modo de operación del MPSC de tal manera que pueda ser programado para operar en el modo SDLC. Generalmente, debe ser inicializado con los siguientes parámetros: Modo SDLC (WR4; D5,D4), Longitud del carácter de transmisión (WR5; D5,D6), Polinomio SDLC (WR5; D2), Modos de interrupción (WR1,WR2), Habilitación del transmisor (WR5; D3), Habilitación del receptor (WR3; D0). Los parámetros del WR4 deben ser escritos antes de ser escritos los parámetros de los registros WR1, WR3, WR5, WR6 y WR7.

El MPSC provee dos canales de comunicación totalmente independientes que pueden ser configurados en varios modos de operación. Como se mencionó anteriormente, el MPSC puede operar en diferentes modos: Por poleo, Interrupciones (vectoreal o no vectoreal), Transferencia de Bloques bajo el control de un DMA. Además, cada uno de los canales pueden ser configurados dentro del modo dúplex integral "full duplex" y pueden operar en un modo o protocolo diferente en cada uno de los canales.

En nuestro Sistema debido a las necesidades y características del mismo, se optó por el uso de las Interrupciones no vectoriales. En este modo de operación, todas las operaciones de transmisión y de recepción son reportadas al procesador a través de las interrupciones, lo cual, nos permite tener un manejo de información en forma mas precisa y rápida. Todas las interrupciones son generadas por el MPSC cada vez que este requiere servicio.

En el Apéndice B se muestra el esquema de interrupciones del MPSC-8274.

A continuación se da una breve explicación de la forma en que se da la transmisión y la recepción de caracteres en el modo de operación por interrupciones.

5.4.2.1. Transmisión en el modo de interrupción.

Por lo que se refiere a la transmisión en el modo SDLC, como se mencionó anteriormente, nuestro sistema va a operar por medio de interrupciones. El MPSC debe ser configurado dentro de este modo de operación, es decir, que los canales son programados para manejar las interrupciones (inicialización del dispositivo a través de registros de lectura WRO-WR7).

Después de un "reset" de canal, el MPSC empieza mandando banderas SDLC (01111110).

Para iniciar la transmisión de una trama, el primer carácter de datos debe ser cargado al MPSC desde la CPU. Debe hacerse notar que en el modo SDLC, este primer carácter de datos debe ser la dirección de la estación esclava direccionada por el MPSC.

El "buffer" de transmisión esta constituido por un "buffer" de transmisión y un registro serial de cambio, como se muestra en la figura 5.8 . Cuando el carácter a transmitir es transferido del "buffer" de transmisión hacia el registro serial, una interrupción se genera debido a que el "buffer" de transmisión esta vacío. La CPU tiene un byte de tiempo para dar servicio a esta interrupción y cargar otro carácter dentro del "buffer" de transmisión, en caso contrario el MPSC generaría otra interrupción debida a la condición de "Transmit Underrun".

Este proceso continúa hasta que no haya más caracteres que mandar. En este punto, la CPU no responde a la interrupción con un carácter sino que simplemente usa el comando de Reseteo de Interrupción en la Transmisión "Reset Tx INT/DMA pending" (WR0: D5 D4 D3 = 101). El MPSC entraría en el estado de "underrun", lo cual simplemente significa que tanto el "buffer" de transmisión así como el registro serial estan vacíos. En este punto, el carácter de bandera (7EH) o el byte de CRC estan cargados en el registro de transmisión de cambio "transmit shift register". El bit de "transmit underrun" es puesto (RR0: D6 = 1) y se genera una interrupción .

Por otra parte, como se mencionó anteriormente, el campo de FCS (CRC) consiste de 2 bytes. El MPSC genera el CRC en todos los datos que son cargados por la CPU. Durante la inicialización se debe escoger el CCITT-CRC (WR5: D2) para el modo de operación SDLC. Este CRC es mandado al final del campo de información. Durante la transmisión el comando "Reset Tx Underrun / EOM" (WR0: D7 D6 = 11) asegura que al final del mensaje el CRC va a ser mandado automáticamente al final del mensaje. Si este comando no es usado durante la transmisión de caracteres de datos, el CRC no va a ser insertado y en lugar de esto se mandarían las banderas de SDLC (7EH).

Asumiendo que la transmisión de caracteres fué satisfactoria, después de que los bytes de CRC han sido transmitidos, la bandera de terminación (7EH) es cargada dentro del "buffer" de transmisión y es transferida al registro serial, una interrupción de "buffer" de transmisión vacío se genera. Si ninguna otra trama va a ser transmitida el comando de "Reset Transmit INT/DMA pending" resetea esta interrupción.

Por el contrario, cuando es detectado un error en la transmisión se debe usar el comando de Aborto (WR0: D5 D4 D3 = 001) y tendrá que ser retransmitida la información.

5.4.2.2. Recepción en el modo de interrupción.

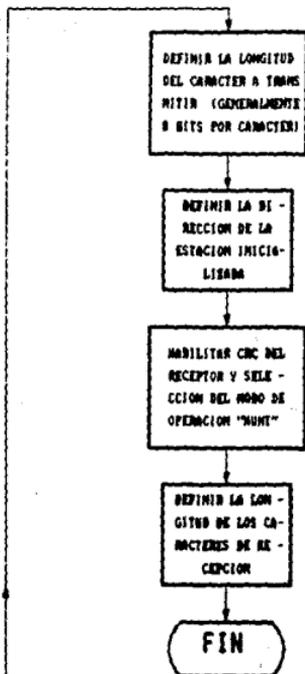
Al igual que en la transmisión, el receptor debe ser inicializado apropiadamente para operar por interrupciones. El receptor debe ser programado dentro del modo "Hunt" (WR3: D4) antes de que sea habilitado (WR3: D0). El MPSC entra en un estado de caza de fase "Hunt Phase" y queda en este estado hasta que la primera bandera SDLC es recibida. El MPSC nunca más vuelve a entrar en este estado a menos que por medio del microprocesador se vuelva a escribir el comando o parámetro de caza de fase. El MPSC podría detectar banderas por un simple cero. Por ejemplo, el patrón de bits 0111111011111110 puede ser detectado como dos banderas.

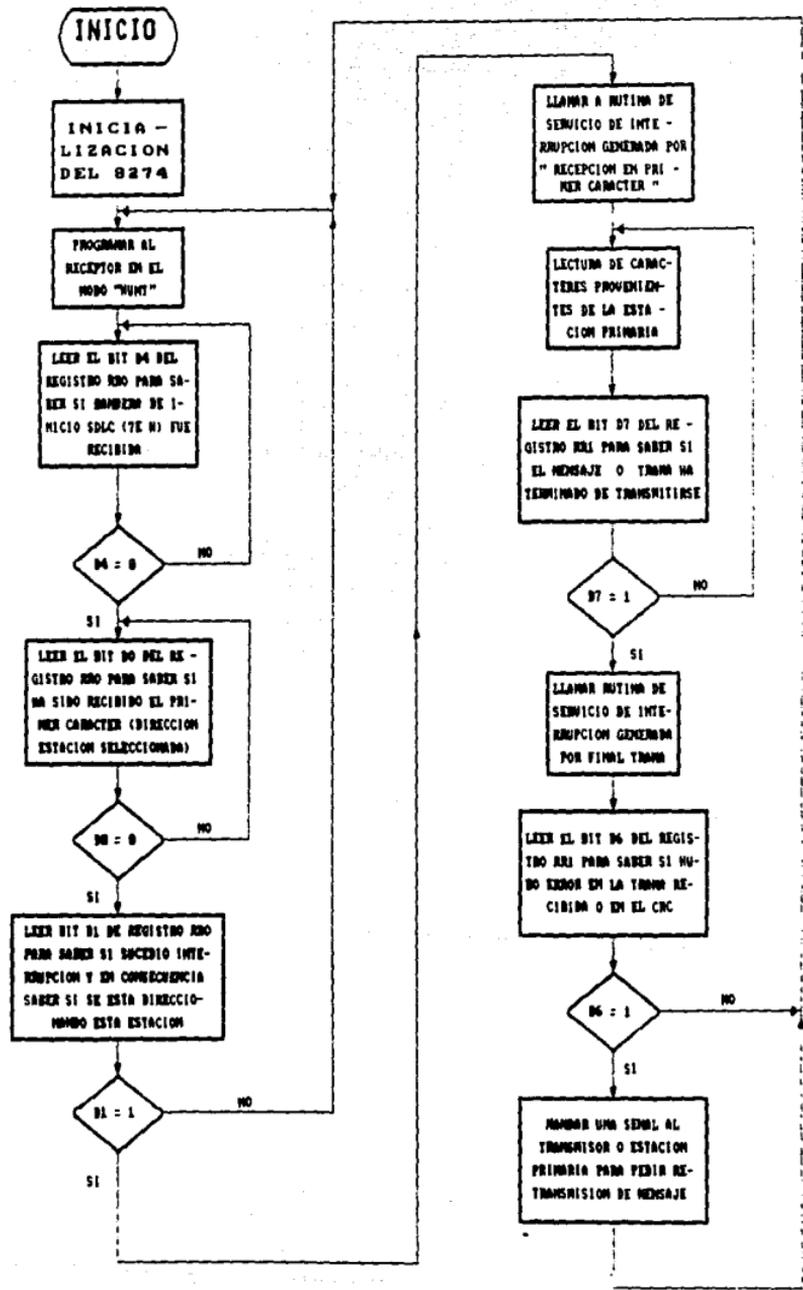
El MPSC puede ser programado para recibir todas las tramas o bien puede ser programado el modo de búsqueda de dirección "Address Search Mode". En este último modo, sólo las tramas con la dirección semejante al valor cargado en el registro WR6 son recibidas por el MPSC.

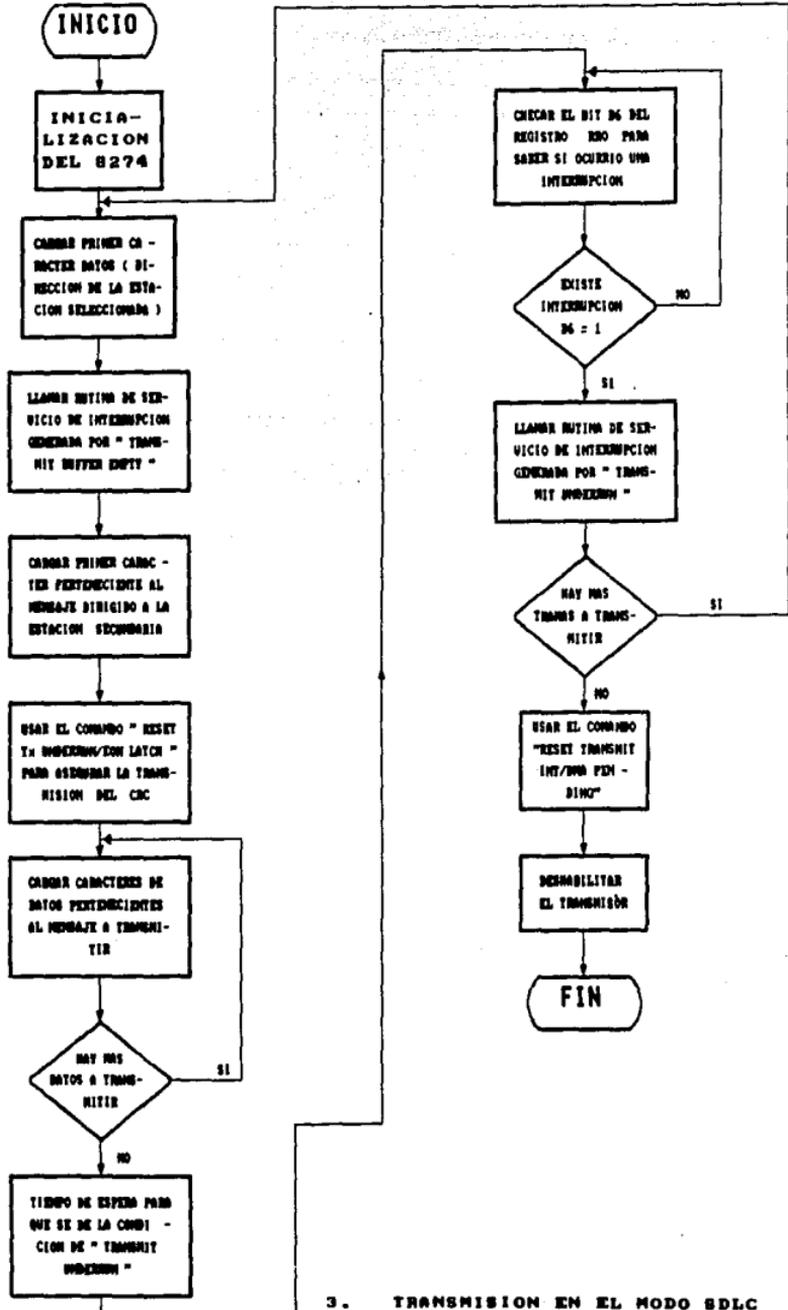
Por otra parte, al recibir los caracteres provenientes del transmisor, en la recepción se dan ciertas condiciones que a su vez generan diferentes tipos de interrupción. Por ejemplo, cuando el receptor es programado en el modo de interrupción en el primer carácter de datos (WR1: D4 D3 = 01), en el momento que el primer carácter es recibido (dirección de la estación) se genera una interrupción la cual es deshabilitada por el comando de deshabilitación de la interrupción en la recepción (WR1: D4 D3 = 00).

Otra de las fuentes de interrupción se genera cuando se dan las condiciones especiales en la recepción como: Error en la trama, Final de la trama, Recepción sobrecorrida "Receive Overrun". (Ver Apéndice B).

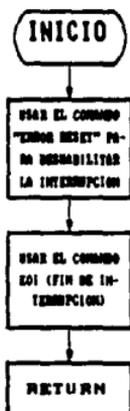
Por último, se muestra a continuación el algoritmo para llevar a cabo la transmisión y recepción de caracteres en el modo SDLC utilizando el MPSC-8274.







3. TRANSMISION EN EL MODO SDLC



ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

CAPITULO 6

6.**LA UNIDAD DE PROCESAMIENTO CENTRAL**

Este capítulo está dedicado al análisis de la Unidad de Procesamiento Central (UPC). Dentro del mismo plantearemos la estructura general de dicha unidad y el principio de operación, tanto de hardware como de software.

Recordemos que esta unidad está formada por una **computadora personal (PC)** compatible con IBM del tipo XT; una **tarjeta de adquisición de datos**, la cual permitirá establecer la comunicación entre la PC y la Unidad Central de Comunicación (UCC). Así mismo, un software de comunicación entre ambas unidades y un software que permitirá al usuario generar reportes estadísticos y gráficos de cada uno de los subprocesos que se involucren en el proceso industrial de que se trate.

6.1. ARQUITECTURA DE LA COMPUTADORA PERSONAL IBM XT

Como la parte esencial de esta unidad es una computadora personal, consideramos pertinente realizar un breve análisis de la arquitectura de dicha computadora, para así poder comprender en forma más clara la interface con el resto del sistema.

La **tarjeta del sistema "MotherBoard"** es el corazón de una PC, y es indispensable conocer su funcionamiento para poder realizar cualquier interface con ella. La figura 6.1 muestra un diagrama de bloques de la tarjeta, cuyas funciones principales se describen a continuación.

6.1.1. Microprocesador 8088.

El 8088 es un microprocesador de 16 bits, con un bus de datos a memoria de 8 bits. Las instrucciones propias del microprocesador pueden manejar datos de 16 bits, sin embargo, los datos y las instrucciones se escriben a memoria en conjuntos de 8 bits cada vez. Con el microprocesador 8088 es posible direccionar hasta un megabyte de memoria, ya sea para datos o programas.

6.1.2. Circuitos de Reloj.

La computadora personal con microprocesador 8088, cuenta con un **reloj de 4.77 MHz**, lo que da por resultado un sistema con un reloj de 210 nanosegundos. La mayoría de los ciclos de bus constan de 4 ciclos de reloj, por lo que un ciclo típico de memoria, requiere de 840 nanosegundos. Las señales de tiempo utilizadas por el microprocesador, así como por las diversas funciones de la tarjeta del sistema, son derivadas de un cristal

de cuarzo que oscila a una frecuencia de 14.31818 MHz. Esta señal es dividida entre 3 por el circuito 8284A, para obtener la señal de 4.77 MHz. Una señal de 1.19 MHz, resulta de dividir la señal original entre 4, y es utilizada para manejar las entradas al reloj para los contadores de tiempo del sistema.

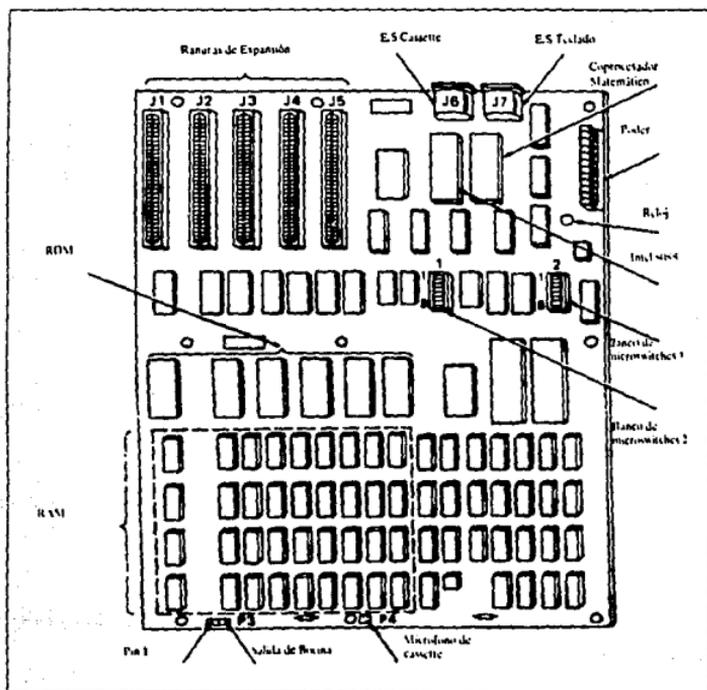


Fig.6.1 Componentes de la Tarjeta del Sistema

6.1.3. Bus del Sistema.

La mayoría de los componentes funcionales de la tarjeta del sistema, están conectados al microprocesador 8088 a través del bus del sistema. Este bus está formado por líneas de distintos tipos de señales: datos, direcciones, control, tiempo, requerimiento de interrupciones y control del DMA (Acceso Directo a Memoria). A este bus se encuentran conectados:

- Un conector auxiliar para el coprocesador numérico 8087
- Un control de interrupciones de 8 niveles (8259A)
- El controlador de bus 8288
- Circuitos demultiplexores y de recarga de bus

La señal del controlador del bus y de los circuitos demultiplexores del bus local, forman las señales básicas que representan el bus del sistema.

6.1.4. Memoria ROM (Read Only Memory).

Conectada al bus, se encuentra una ROM (Memoria de Solo Lectura) de 40 KBytes. La ROM está decodificada de tal manera, que reside en la parte superior del área de memoria de 1 MByte direccionado por el microprocesador. Las funciones soportadas por la ROM son:

- a) Inicialización del sistema.
- b) Diagnóstico de encendido y revisión del sistema.
- c) Determinación de la configuración del sistema.
- d) Manejo de dispositivos de E/S, comunmente llamado BIOS (Sistema Básico de E/S).
- e) Cargado del sistema operativo.
- f) Patrones de bits para los primeros 128 caracteres ASCII de un grupo de 256.

6.1.5. Memoria RAM (Random Access Memory).

La tarjeta principal del sistema, está diseñada de tal forma que se pueden instalar bancos de memoria con incrementos desde 64 hasta 256 KBytes. La memoria RAM reside en el espacio que el microprocesador 8088 direcciona a partir de la dirección 00000 y se extiende hasta la dirección 0FFFF; asumiendo que se encuentran instalados 64 KB de memoria. Si la memoria instalada, fuera menor de 64 KB no se podría instalar memoria adicional a través de tarjetas de expansión de memoria, ya que el bus del sistema hacia las ranuras de expansión "slots" no está habilitado para manejar direcciones menores de 64 KBytes. En caso de pretender instalar tarjetas de expansión de memoria, sin antes haber instalado los 64 KB de memoria básicos, la memoria instalada no sería continua.

La tarjeta principal del sistema usa circuitos integrados de memoria dinámica de 64 o 256 Kb X 1, con un tiempo de acceso de 100 a 150 nanosegundos. Estos componentes requieren de tres niveles de alimentación: +5, -5, +12 volts.

6.1.6. Contador de Tiempo.

Para realizar las funciones de cronometría y conteo, existe un circuito integrado 8253-5, conectado al bus. Las entradas a este reloj son manejadas por la señal de un oscilador a 1.19 MHz. La salida del temporizador en su canal 0 está asociada a la interrupción del sistema a nivel 0 y se encuentra programada para generar una interrupción cada 54.925 milisegundos (aproximadamente 18.2 veces por segundo). Este temporizador es usado por las rutinas de E/S del sistema y por el reloj del mismo, que mantiene la fecha y la hora del día. La salida del canal 1 es usada para generar un requerimiento al DMA en el canal 0 del mismo. Este requerimiento se realiza para refrescar la memoria dinámica del sistema, al crear un ciclo inútil de lectura a memoria por cada 72 señales de reloj del procesador o cada 15.12 s. La salida del canal 2 es usada para manejar el audio del sistema.

6.1.7. DMA (Direct Memory Access).

Los dispositivos externos como la unidad de discos, transmiten datos en forma más rápida de lo que el procesador puede manejar, por lo que la tarjeta del sistema cuenta con una facilidad para permitir la transmisión directa de datos entre un dispositivo y la memoria, sin involucrar al microprocesador. Esta función se realiza a través del circuito 8237-5, controlador de DMA. Este dispositivo soporta transmisión de datos entre la memoria y cuatro adaptadores. Cuando un adaptador o dispositivo desea transmitir información, solicita al 8237-5 la realización de esta función. El controlador de DMA pide al 8088 dejar libre el bus del sistema, para tomar control sobre él y transmitir los datos.

El canal 0, de los cuatro con que cuenta el DMA, es usado para realizar el refresco de memoria. Los otros tres canales están disponibles en el bus del sistema a través de las 5 ranuras de expansión localizadas en la tarjeta principal del sistema.

6.1.8. Interrupciones.

El microprocesador 8088 cuenta con dos fuentes de interrupciones: **maskables** y **no maskables**. Para manejar más niveles de interrupciones, se cuenta con el circuito 8259A, que es un controlador de interrupciones. Este componente expande las interrupciones maskables a 8 niveles. El circuito integrado no se encuentra conectado al bus del sistema, sino al bus local. De los ocho niveles de interrupción, dos son usados por la tarjeta del sistema y los seis restantes se encuentran disponibles en el bus del sistema, pudiéndose manejar a través de las ranuras de expansión.

6.1.9. Operaciones del Bus.

La mayoría de las aplicaciones que requieren de interface utilizan una de las ocho ranuras de la tarjeta principal del sistema (ver figura 6.2), para conectarse a la PC. Los datos son transmitidos en un ciclo que se ha denominado **ciclo de bus**. Existen dos clasificaciones de los ciclos de bus: los manejados por el procesador 8088 y los manejados por el DMA. Cuando el microprocesador genera un ciclo de bus, maneja el

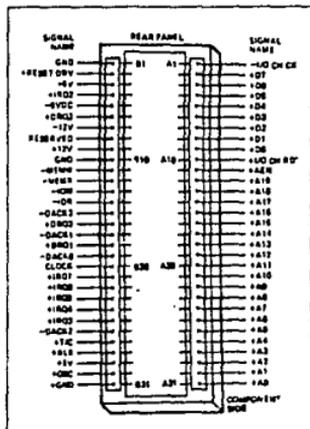


Fig.6.2 Señales para las Ranuras de Expansión

bus del sistema con la dirección de una localidad de memoria o un puerto de E/S; controla la dirección del flujo de datos y es, ya sea la fuente o el destino de los datos. Cuando el 8088 maneja el bus, se generan cinco tipos de ciclos del bus. El primer tipo es un ciclo de lectura a memoria. El segundo es un ciclo de escritura a memoria. El tercer tipo es un ciclo de lectura de un puerto de E/S. El cuarto es un ciclo de escritura a un puerto de E/S, en tanto que en el quinto tipo es un ciclo de reconocimiento de interrupción.

La segunda clasificación general de los ciclos de bus abarca aquellos manejados por el controlador DMA. Durante las operaciones de acceso a memoria (DMA), se retira el microprocesador 8088 del bus del sistema, dejando el control de los ciclos del bus transferidos directamente entre el adaptador de la interface y la memoria.

6.1.9.1. Ciclo de lectura a un puerto de entrada/salida.

El ciclo de lectura de un puerto de E/S se inicia cada vez que se ejecuta una instrucción de entrada del microprocesador 8088. Este ciclo es similar al ciclo de lectura a memoria. En el diseño de una PC, este ciclo consta de un mínimo de cinco ciclos de reloj, lo que equivale a 1.05 microsegundos. Durante un ciclo de E/S al puerto, el microprocesador 8088 coloca una dirección de 16 bits correspondiente a un puerto, en el bus de direcciones. Durante este ciclo, los 4 bits más significativos del bus de direcciones, nunca son activados.

Durante el ciclo T_1 , la señal ALE es activada, indicando que los bits 0-15 del bus de direcciones contienen una dirección de E/S válida. En el tiempo T_2 , la señal de control IOR, es activada, indicando que el ciclo corresponde a un ciclo de lectura de un puerto y que el puerto asociado a la dirección debe responder manejando el bus de datos con su contenido. Al inicio del ciclo de reloj T_4 , el procesador toma los datos presentes en el bus de datos y la señal IOR es desactivada. El ciclo de lectura termina al finalizar el ciclo T_4 . Se puede notar que un ciclo de lectura se compone de cuatro ciclos de reloj, sin embargo, en la arquitectura de una PC se agrega un ciclo de reloj extra, llamado TW (tiempo de espera), por cada ciclo.

6.1.9.2. Ciclo de escritura a un puerto de entrada/salida.

Un ciclo de escritura a un puerto de E/S se inicia cada vez que se realiza una instrucción de salida del microprocesador 8088. Este ciclo escribe datos del microprocesador a la dirección específica de un puerto de E/S. Normalmente, este ciclo comprende cuatro ciclos de reloj. Pero en la arquitectura de la PC se inserta un ciclo extra TW. Sólo los bits 0-15 del bus de direcciones son usados para direccionar a los puertos de E/S.

La señal ALE es activada durante el ciclo de reloj T_1 , en seguida la señal de control IOW es activada en el tiempo T_2 , indicando que el ciclo es de escritura a un puerto de E/S y que el puerto seleccionado debe tomar datos del bus de datos. Al iniciar el ciclo T_4 la señal IOW es desactivada.

6.1.9.3. Comunicación síncrona y asíncrona.

El único acceso al medio exterior de una PC, oficialmente reconocido por IBM hasta antes de la liberación de la familia PS/2, fue el puerto de comunicación asíncrona, conocido también como puerto serial, que trabaja bajo el estándar RS232C. Todos los puertos seriales de IBM tienen la misma función, toman 8 bits al mismo tiempo y los convierten en un tren de pulsos seriales.

La característica que distingue a una comunicación serial es que todos los datos e información de control son transmitidos en una sola línea, un bit a la vez. En la comunicación asíncrona, la información es transmitida utilizando un bit de inicio y un bit de paro, de tal manera que tanto el emisor como el receptor puedan sincronizar la transmisión. En contraste, la comunicación síncrona generalmente se caracteriza por transmitir la información de acuerdo a intervalos de tiempo fijos.

Un bit de datos es transmitido al poner la línea a un estado lógico. La duración de cada bit determina la velocidad de transmisión, medida en bits por segundo. Tanto el emisor como el receptor deben tener la misma velocidad. Cuando no está ocurriendo ninguna transmisión la línea se deja en un estado de "marcado", con un nivel de 1 lógico. Para empezar la transmisión, se transmite un bit de inicio, cuyo estado lógico es opuesto al del estado marcado.

Después de que se transmite el bit de inicio, se transmiten los bits de datos. La configuración de bits de datos soportada por el puerto de IBM es de 5,6,7 y 8. el bit de menor orden es el primero en ser transmitido. Después de transmitir los bits de datos se puede mandar un bit de paridad que es opcional. Los bits de paridad son usados para ayudar a detectar errores de transmisión. Finalmente, después del paquete de datos, se transmiten uno o dos bits de paro. El uso de bits de inicio y de fin permiten que tanto el emisor como el receptor puedan transmitir información a cualquier tiempo, es decir, asincrónamente.

Todas las computadoras personales de IBM y la familia PS/2 cuentan con un circuito integrado dedicado, llamado UART (Universal Asynchronous Receiver Transmitter). El circuito específico es el INS8250 (INS6450 para AT e INS16550 para PS/2).

El INS8250 es un microprocesador programable que permite que los datos sean transmitidos y recibidos serialmente. Soporta las velocidades más comunes, tipos de paridad, número de bits de datos y bits de inicio y paro.

El UART usa cuatro registros internos para la transmisión de datos: el registro receptor de datos, el registro receptor de corrimiento, el registro transmisor de retención y el registro transmisor de corrimiento. Cuando una entrada de datos llega a una línea de datos, es colocada en el registro receptor de corrimiento. Después de que se han recibido todos los bits necesarios (inicio, datos, paridad y paro); los bits de datos son transferidos al registro receptor de datos. Cuando los datos están listos, el UART prende una bandera; entonces los datos pueden ser transferidos del registro receptor de datos del UART, al CPU ejecutando una instrucción IN. Después de que los datos son leídos, se limpia la bandera de "datos listos", hasta que se recibe otro carácter.

La transmisión de datos es similar a la recepción. Un carácter es colocado en el registro transmisor de retención, usando una instrucción OUT. El UART lo copia al registro transmisor de corrimiento, después de colocarle los bits necesarios (inicio, paridad y paro).

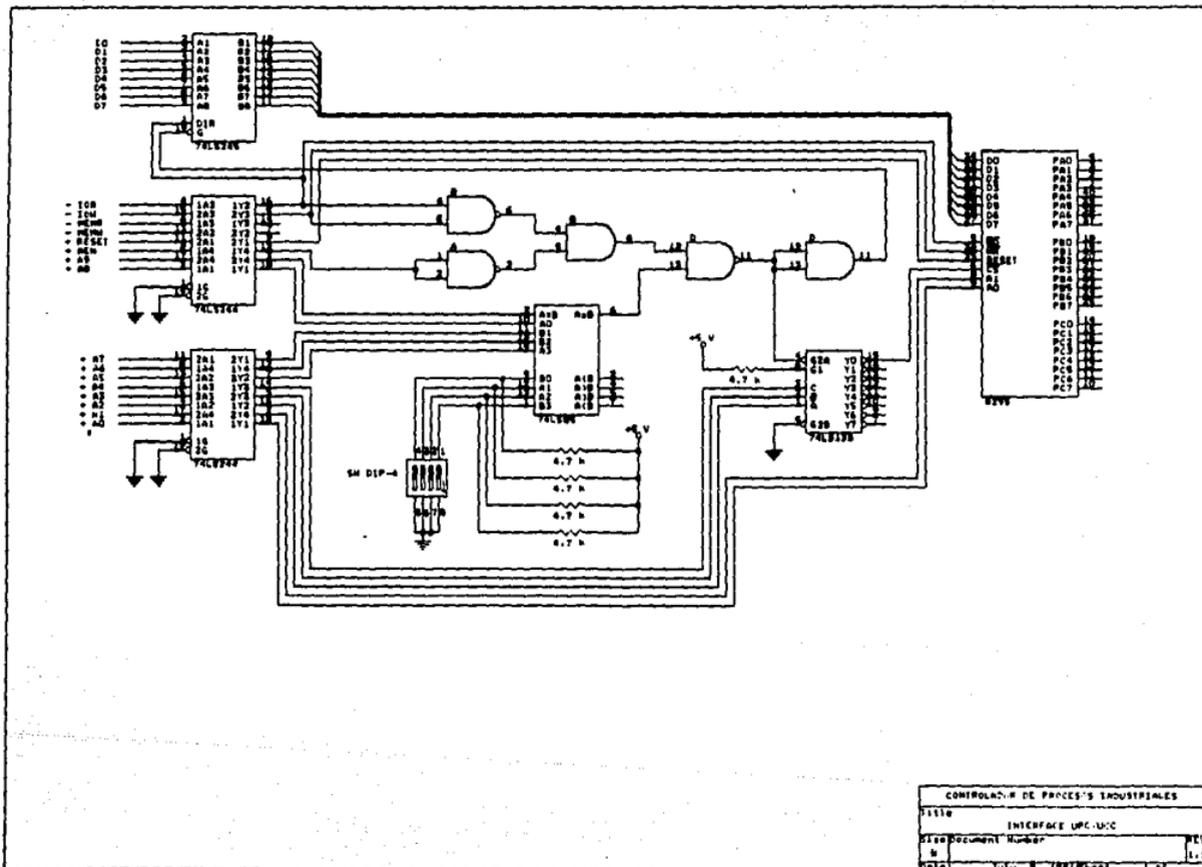
6.2. TARJETA DE ADQUISICION DE DATOS

Una vez revisada en forma bastante general la arquitectura de una PC, expondremos el principio de funcionamiento de la tarjeta de adquisición de datos (ver figura 6.3) que diseñamos para realizar la interface de comunicación entre la UPC y la UCC; dicha comunicación es de tipo **paralelo**.

El circuito 1 es el 74LS245, éste permite que los datos fluyan desde el "slot" de la computadora hacia la circuitería de la tarjeta, o que los datos vayan desde la tarjeta hacia el "slot".

La dirección del flujo de datos está determinada por el "pin" DIR en el chip. El flujo es controlado por la línea "Output Enable". Si la línea no está habilitada, el dispositivo cambia a su 'tercer' estado. En ese tiempo, se considera que no existen cargas en el bus; esto permite que cualquier dispositivo comparta el mismo bus sin tener problemas.

El 74LS245 manejará un bus de 64 pines durante lecturas de puertos de E/S en el rango de 300-31F hexadecimal, o 768-799 decimal.



CONTROLADOR DE PROCESOS INDUSTRIALES	
1110	INTERFACE OPC-UCC
THIS DOCUMENT NUMBER	
1110	1.1
DATE	JULY 9, 1981

Los circuitos 2 y 3, 74LS244, son manejadores de líneas por byte que habilitan las terminales que están conectadas a tierra. Dado que la señal de habilitación está en un nivel activo bajo, este chip siempre pasa las señales del bus directamente a un "buffer".

El circuito 6 es alimentado por señales de dos fuentes lógicas. La función del 74LS85 es comparar los niveles que llegan desde esas dos fuentes y entrega una señal de nivel alto en el pin 6 cuando son iguales. El bit de dirección A9 es usado como una compuerta para habilitar el 74LS85 y debe ser de nivel alto para permitir la función del dispositivo.

Las líneas de dirección A8-A5 son comparadas con las entradas que vienen del banco de resistencias y del switch. Para tener una mayor flexibilidad en el direccionamiento, se tiene el switch.

El circuito 5, un 74LS138, decodifica las direcciones desde A4 hasta A2. Cada salida seleccionada del 74LS138 habilitará cuatro direcciones consecutivas. Por ejemplo, el pin 14 será activo (bajo) para direcciones de 304-307 hexadecimal.

Estos bloques de cuatro direcciones son muy convenientes cuando se utilizan dispositivos tales como 8253, 8255, 8259 y 8250.

Los circuitos 4 y 7 proporcionan el resto de la lógica requerida para usar la tarjeta en un amplio rango de control de E/S.

La habilitación de direcciones, AEN (Address Enable), es usada durante un acceso directo a memoria, DMA (Direct Memory Access) y no debe ser verdadera durante las operaciones de E/S.

Para activar completamente esta tarjeta, las siguientes condiciones deben presentarse.

- a) Lectura o escritura de E/S.
- b) Ninguna dirección habilitada de DMA.
- c) La salida del comparador es verdadera.

La tarjeta que planteamos es de propósito general, ya que si se tiene cuidado al decodificar el rango adecuado de direcciones, se pueden tener controladores de juegos, puertos paralelos, un puerto de comunicaciones asíncronas, un reloj calendario, etc.

Para establecer la comunicación entre las dos unidades se ha empleado el circuito 8255, que es un dispositivo de E/S programable de propósito general (Ver Apéndice C).

6.3. SOFTWARE PARA EL CONTROL DE SISTEMAS EN LINEA

Dentro de la industria, se requiere de sistemas que respondan en el momento en que son alterados los distintos parámetros manejados en los procesos. La mayoría de éstos, son controlados manualmente y no siempre responden a tiempo.

Los avances en microcomputadoras y la tecnología de comunicaciones han producido un incremento en la funcionalidad y distribución física de una planta de control industrial, así como en los sistemas de información. Una de las primeras etapas de esta evolución fue la interconexión entre elementos de control, terminales y recursos de cómputo. Esa combinación de tecnología de la computadora y los sistemas dinámicos da como resultado gran eficiencia y alta productividad en los procesos.

Lo anterior es consecuencia de que en nuestros días el uso de microcomputadoras personales se ha incrementado explosivamente. Una razón por la cual ha ocurrido esto, es por que conforme aumenta el número de computadoras personales en el mercado, el precio de las mismas baja. Por otro lado, se cuenta con el software desarrollado que ofrece las facilidades para tener varios programas actuando al mismo tiempo.

El contar con programas que operen concurrentemente, responde a la necesidad que se tiene en el control de procesos industriales, de supervisar diversas variables físicas y ejercer acciones de control cuando sean requeridas.

Una computadora puede realizar las siguientes funciones: capturar información, procesarla y emitir acciones a ejecutar. Con esta herramienta se va más allá de mallas cerradas de control de parámetros, pasando a estrategias más sofisticadas de control interactivo.

Las formas en que la computadora es usada en aplicaciones de control son básicamente dos:

a) Control Digital Directo.

Reemplaza a los controladores analógicos y la acción de control se ejecuta por medio de un programa de la computadora digital, de esta manera varios controladores analógicos pueden ser substituidos por un programa de computadora apropiado.

b) Control Supervisor o Puntos de Control.

Consiste de un programa maestro que constantemente calcula y actualiza los puntos de control de los controladores analógicos en base a una estrategia de operación predeterminada. La parte física del controlador ajusta los puntos de control de los controladores analógicos para mantener la operación de una planta en el nivel más adecuado.

Los programas desarrollados para computadoras personales son por lo general muy sencillos de aprender y usar, además de que ofrecen características que se encuentran en minicomputadoras multiusuarios.

En los párrafos posteriores se expondrán las técnicas básicas para desarrollar sistemas de control en línea.

6.3.1. Técnicas Básicas.

Existen dos técnicas comerciales de procesamiento de datos: los procesamientos en cola y los procesamientos en línea. Los primeros no requieren de una respuesta inmediata, las peticiones son atendidas una por una y en base a prioridades, a diferencia del procesamiento en línea.

Algunas de las ventajas que ofrecen los procesamientos en línea son:

i) Las actualizaciones son ejecutadas en el momento mismo en que se realizan las operaciones, con lo que se dispone de datos reales al acceder el sistema.

ii) El control de este tipo de procesos es real, con lo que no se debe someter a procesamientos posteriores.

6.3.1.1. Sistemas en línea.

Las etapas principales de los sistemas en línea son:

i) El sentido o captura de datos, esto se realiza por medio de terminales a través del teclado o una tarjeta de adquisición de datos. Las terminales son diseñadas de acuerdo a necesidades específicas dependiendo de la aplicación.

ii) La validación de las entradas consiste de un programa que se encarga de identificar los datos capturados, con la finalidad de decidir el procesamiento de los mismos.

iii) El procesamiento de la información, esto es, la computadora se programa para tomar decisiones sobre las entradas. El procesamiento se efectúa rápidamente, verificando todas aquellas condiciones excepcionales que hayan sido programadas.

iv) El resultado del procesamiento es una acción a ejecutar, ésta se envía a través de los actuadores del sistema.

6.3.1.2. Definición de software en tiempo real.

Como se mencionó en capítulos anteriores, hay aplicaciones donde la velocidad de respuesta ante una exigencia externa es decisiva. El número y frecuencia de estas exigencias determinarán la forma en que sean manejadas. El empleo de dispositivos de

E/S es la solución de hardware más económica, pero es lenta en comparación al uso de interrupciones.

Se denomina software en tiempo real, al conjunto de programas capaces de obtener resultados a entradas de datos en forma rápida, los cuales pueden influir en el proceso a ser controlado. La gran ventaja de permitir el manejo de unidades usadas en procesos industriales, muestra que tan variable puede resultar el cuantificar procesos en tiempo real.

Los elementos de software en tiempo real son: un conjunto de datos que se colectan y que informan sobre un ambiente exterior; un análisis que transforma la información que se requiere de la aplicación; un controlador de salida que responde al ambiente externo, y un monitoreo que coordina a todos aquellos componentes que responden en tiempo real al ser controlados (típicamente ocurren en un rango de un milisegundo a un minuto). Hay que notar que el término de tiempo real difiere del término interactivo, debido a que un sistema en tiempo real debe responder estrictamente en el tiempo que se requiere; en cambio, la respuesta en tiempo de un sistema interactivo puede normalmente ser excedido sin resultados desastrosos.

Para llevar a cabo lo anterior se usa una herramienta del sistema de control de la computadora, o mejor llamada sistema operativo, el cual puede ejecutar varios programas al mismo tiempo, y tener residente en memoria un programa al cual se pueda hacer referencia en el tiempo en que se requiera.

6.3.1.3. Sistema de adquisición de datos.

Un sistema de adquisición de datos puede localizarse fuera o dentro de la planta, de tal forma que las terminales puedan estar esparcidas a través de toda la organización. La instalación de un captador y un sistema de colección de datos tiene como efecto directo un mejor aprovechamiento en la precisión con lo que los registros son capturados, así como la forma en que son reconocidos.

La colección de datos es sólo el primer paso para usar la computadora como herramienta de control de procesos de la organización. Este es un paso esencial y con frecuencia difícil, puesto que hay que seleccionar debidamente los datos son indispensables para un sistema dado. El segundo paso consiste en formular las reglas de decisión necesarias para implementar las diferentes situaciones a las que se enfrentará durante los procesos a controlar, y el tercer paso será cerrar el ciclo con la toma de alguna decisión.

Varias de las acciones controladas se manejan con base en decisiones, y éstas se pueden realizar a través de la computadora. La computadora viene siendo un sistema controlador de información y tiene dos tipos de comunicación con el usuario.

i) A través del monitor y el teclado. Este es un modo de operar a la computadora muy sencillo, puesto que ofrece un ambiente amigable. Por este medio se pueden realizar

encuestas; inspeccionar información en archivos y realizar modificaciones sobre procesos; entre otras cosas.

ii) El segundo tipo de comunicación se realiza por medio de reportes impresos, medio por el cual se pueden obtener informes en papel de los sucesos de cualquier proceso.

6.3.1.4. Multiprogramación.

La multiprogramación consiste básicamente en ejecutar de una forma virtual varios programas al mismo tiempo, contando únicamente con un microprocesador.

La importancia de la multiprogramación radica en que hay ocasiones en las que mientras la máquina está leyendo de un archivo y procesando la información, se desea imprimir los resultados del mismo, con lo que se requiere hacer uso de diversos dispositivos. La computadora no puede llevar a cabo todas las operaciones simultáneamente, con lo que tiene que simularlo, y a esta forma de trabajar del procesador se le conoce como multiprogramación. El tiempo compartido es un buen ejemplo de multiprogramación, ya que lo que en realidad se hace es dar pequeñas porciones de tiempo para atender a cada proceso, con lo que pareciera que se realizan éstos simultáneamente. El encargado de realizar lo anterior es el sistema operativo del equipo de cómputo que se utilice.

6.3.1.5. Programas residentes en memoria.

Un programa residente en memoria es aquél que se encuentra en cualquier instante de tiempo dentro de la memoria RAM, y con solo ser invocado se tiene acceso a él. Para realizar lo anterior es necesario tener memoria RAM disponible y que el sistema operativo lo permita. Esta técnica ha sido muy popular en muchos programas comerciales, a nivel de computadoras personales.

Existen herramientas que permiten crear programas residentes en memoria para computadoras que tienen un solo procesador, como lo son las microcomputadoras personales. Los programas residentes en memoria (TSR) se basan en llamar a una interrupción, cuya rutina de atención pone al programa en una porción de memoria que jamás se borra. Anteriormente era muy útil realizar este tipo de programas en lenguaje ensamblador y a niveles de interrupción muy bajos, actualmente se puede escribir este tipo de programas en lenguajes de alto nivel como son Pascal y C.

Una vez que un programa se encuentra residente en memoria, el sistema operativo toma el mando de la máquina y realiza otras tareas. Los programas residentes en memoria tienen una rutina de servicio de interrupción (ISR), que se encuentra ligada a las interrupciones del teclado. Las llamadas a un programa residente pueden realizarse de dos maneras: mediante el ISR, que detecta el momento en el cual el usuario oprime una tecla específica, o con una simple invocación al programa.

Esto lleva a definir otro término, la "reentrada". Esta consiste en que un programa pueda ser usado más de una vez al mismo tiempo por varias rutinas, por lo que los programas residentes en memoria pueden ser reentrantes, trayendo consigo múltiples ventajas en lo referente al tamaño del código de programas.

6.4. SOFTWARE EN LINEA PARA COMPUTADORAS PERSONALES

Como ya vimos, hoy en día se cuenta con los recursos tanto de software como de hardware, para que una computadora personal compatible con IBM pueda ejercer control de sistemas en línea.

Por parte de software existen sistemas operativos que ofrecen ventajas para desarrollar estos sistemas, así como lenguajes de programación de alto nivel que sirven como herramienta para resolver satisfactoriamente problemas de control en línea.

Los sistemas operativos más conocidos para este tipo de computadora son DOS y UNIX. Sin embargo, el que ha alcanzado mayor difusión es el primero de ellos.

En cuestión de hardware existen gran cantidad de periféricos y tarjetas de propósito específico, dependiendo de las aplicaciones que se deseen desarrollar.

6.4.1. Sistema Operativo DOS (Disk Operating System).

El sistema operativo de una computadora es la parte más importante de su software, esencialmente en la industria y en aplicaciones de control de procesos. Esto se debe a que el sistema operativo determina que tan rápido puede responder una computadora a ocurrencias de eventos en tiempo.

Aunque el DOS es el sistema operativo dominante, tiene muchas limitantes en el uso industrial, ya que su arquitectura no fue diseñada para atender este tipo de aplicaciones. Por ejemplo, las microcomputadoras personales solo cuentan con un microprocesador y este tipo de máquinas se utilizarían para gran variedad de aplicaciones pero con un solo usuario a la vez.

Estos equipos podrían ser manejados en forma natural por varios usuarios, si y solo si el sistema operativo cuenta con técnicas de multiprogramación, pero DOS no tiene dichas características.

Estas limitaciones son de gran importancia en los programas de aplicación que ejecutan un trabajo de monitoreo en línea, puesto que no deben quedarse en instrucciones cíclicas por ninguna razón. Por ejemplo, si se desea estar monitoreando cuatro entradas

de temperatura, el programador debe permitir que el programa capture estas lecturas durante una operación repetitiva cuando estén cambiando dichas señales.

La tercera limitante consiste en que DOS no puede manejar más de 640 KBytes de RAM en forma natural. Ante esta limitante, actualmente se cuenta con memoria extendida, con la cual una computadora personal tipo AT puede aumentar su memoria de acceso aleatorio, con drivers especiales.

Sin embargo, también se tienen ventajas con este sistema operativo.

El costo es un factor muy importante, ya que DOS es un sistema operativo accesible y con el que la mayoría de los proveedores de computadoras personales compatibles con IBM entregan sus equipos.

El número de lenguajes de programación y compiladores para los mismos es muy variado, además de que cuenta con herramientas para el desarrollo de sistemas en línea.

Las computadoras personales son sistemas abiertos a los cuales se les puede añadir hardware comprado o diseñado por uno mismo, DOS permite tener contacto con este tipo de hardware también.

Existe una gran variedad de paquetes en diversas áreas de negocios, por ejemplo, la administración, dibujo, diseño gráfico entre otros, y dichos paquetes están desarrollados para correr bajo el sistema operativo DOS.

6.4.2. Desarrollo del Software de la UPC.

El software empleado en la UPC podemos pensar que es de dos tipos: un software de comunicación entre la computadora y la UCC; gracias al cual es posible obtener la información presente en la tarjeta de adquisición de datos; así como el software que hace que dicha comunicación sea más amigable para el usuario, ya que a través del programa, se podrán variar los parámetros más significativos de cada uno de los subprocesos, (valor máximo, mínimo, tiempo de adquisición de datos, etc.), obtener valores de cada uno de los subprocesos o variables que se estén manejando. Con esa información se realiza una serie de procesos y ajustes para que se realice, ya sea en forma tabular o por medio de gráficas un análisis estadístico, y opcionalmente la impresión de reportes.

El software involucrado en la UPC puede ser desarrollado en lenguaje Pascal o C. Para poder cumplir con los requerimientos mínimos de este tipo de software es necesario recurrir al empleo de interrupciones ya sea al BIOS o al DOS. En el Apéndice D se muestran en detalle las características principales de dichas interrupciones, así como sus parámetros asociados.

Toda la información que se genera en cada uno de los subprocesos puede ser analizada por métodos gráficos o estadísticos, ya que el programa debe proporcionar al usuario la opción de ver la información en forma tabular o gráfica. En ambos casos, se

presentarán los valores obtenidos antes de efectuar la acción de control y los que se obtuvieron posteriormente.

El programa deberá tener una forma de operación bastante simple, ya que se basa en una serie de opciones que el usuario selecciona de un menú principal. Al realizar la selección del menú, el programa habrá de desplegar ya sea un nuevo menú o la información correspondiente al parámetro seleccionado. En todo momento el usuario tendrá a su disposición una ventana de ayuda en la que se le indique cual es la acción más apropiada a seguir en caso de que se presente alguna dificultad.

El programa tiene que ser capaz de detectar el hardware instalado para el control del video, (tarjeta de gráficas, tipo de monitor, etc.), y si éste no es capaz de realizar gráficas de mediana o alta resolución el programa no trabajará en dicho equipo.

Así mismo, si al momento de mandar a impresión un reporte o gráfica, el programa verificará en primera instancia el estado del impresor; si éste no está conectado o presenta algún error en su funcionamiento, dicho estado se indica al usuario por medio de algún mensaje de error del impresor; de esta manera se podrán tomar las acciones correspondientes de suspender la impresión o corregir el error.

Uno de los aspectos más sobresalientes del programa es que presente una opción de 'Autoverificación', en la que el sistema mande a la UCC una serie de parámetros conocidos y espera en respuesta una serie equivalente de valores plenamente identificados, de esta manera puede establecerse si la comunicación entre ambas unidades es correcta.

Así mismo, contar con una opción de 'Reinicialización', la cual pondrá en condiciones iniciales a todos los subprocesos involucrados. Esta opción tendrá una clave de acceso, (la cual idealmente solo debe conocer el responsable de la planta), debido a la importancia y repercusiones que conlleva una acción de esta magnitud.

En los párrafos posteriores se hará una breve descripción de los aspectos técnicos que involucran el desarrollo del programa.

6.4.2.1. Adquisición de datos.

Para poder obtener la información presente en la tarjeta de adquisición de datos, es necesario conocer la dirección asociada al puerto de entrada de la tarjeta. Conociendo esta dirección se desarrolla un procedimiento que puede leer la información contenida en dicha dirección. Para poder realizar un mejor control de este proceso se establece que la PC realice la adquisición de información de cada una de sus unidades en un período dado, contado a partir del momento en que el sistema es inicializado.

El almacenamiento de la información dentro de la PC se realiza en forma temporal en memoria RAM, empleando una serie de vectores. Posteriormente, se almacena la información en disco, ya sea duro o flexible.

6.4.2.2. Procesamiento de información.

Una vez que la información ha sido almacenada en la memoria de la PC, se puede realizar el tratamiento correspondiente. La información es almacenada originalmente en una serie de vectores; con estos valores se pueden realizar gráficas o tablas.

6.4.2.3. Manejo y control de video.

Para realizar el control de las diferentes opciones del video, se utilizan algunas de las funciones que para tal efecto trae implementadas el compilador Turbo Pascal, en otros casos es necesario el desarrollo e implantación de diferentes rutinas de manejo de video. Para la realización de dichas rutinas, se utiliza la interrupción 10 hexadecimal del BIOS, que es la encargada de todas las funciones del video (Ver Apéndice D).

6.4.2.4. Manejo y control de impresión.

Para la obtención de reportes impresos es necesario emplear la interrupción 17 hexadecimal del BIOS, que es la encargada de atender a la impresora (ver Apéndice D). Es a través de esta interrupción como puede saberse el estado del impresor antes de mandar a imprimir cualquier tipo de información.

La impresión de gráficas se lleva cabo por medio de la interrupción 5 hexadecimal que es la encargada de imprimir la pantalla (ver Apéndice D). Para poder realizar la impresión de gráficas es necesario que la impresora a emplear sea capaz de imprimir gráficas y que este modo de operación sea inicializado en la impresora. Dicha inicialización puede realizarse de dos formas: empleando el comando GRAPHICS de sistema operativo DOS, el cual inicializa la impresora en modo gráfico; y la otra es generar el "driver" correspondiente de gráficas para la impresora desde el programa.

CAPITULO 7

7. PROCESAMIENTO ANALOGICO/DIGITAL CON MICROCONTROLADORES DE LA FAMILIA MCS-51 DE INTEL

7.1. INTRODUCCION

Existen por lo menos tres arquitecturas populares para convertir un voltaje analógico a un valor digital: "Flash", integración y aproximaciones sucesivas.

Los A/D "Flash" son los más rápidos, y los más caros para una exactitud dada. Los convertidores "Flash" tienen velocidades de conversión medidas en las décimas de nanosegundo.

Los convertidores integradores son los más lentos, pero los más exactos. La conversión por integración es generalmente insensible al ruido en la entrada, pero los tiempos de conversión se miden en milisegundos.

Los convertidores por aproximaciones sucesivas proveen un balance entre velocidad y exactitud. Los tiempos en la conversión por aproximaciones sucesivas se mide en las décimas de microsegundo, y la implementación para una cierta exactitud es bastante económica.

La serie MCS-51 de microcontroladores de Intel contienen muchas facultades que pueden ser integradas directamente a muchos tipos de instrumentación. Para ilustrar el poderío del 8031 a continuación se elaborará sobre dos técnicas para realizar la conversión analógica a digital (A/D). En ambos ejemplos se asume que algún hardware adicional se conecta en las terminales de E/S del 8031.

7.2. TECNICA DE CONVERSION POR APROXIMACIONES SUCESIVAS

La conversión A/D por aproximaciones sucesivas involucra una "búsqueda binaria" de un voltaje desconocido relativo a una referencia "fija" conocida. La referencia se divide selectivamente por múltiplos de 2 hasta que se alcanza la exactitud deseada. La figura 7.1 es un diagrama de flujo de un convertidor por aproximaciones sucesivas, por lo general esta técnica requiere que un convertidor D/A divida la referencia de voltaje y que un comparador de voltaje compare el voltaje desconocido con la referencia "dividida". Los convertidores D/A y los comparadores de voltaje son fáciles de conseguir y relativamente

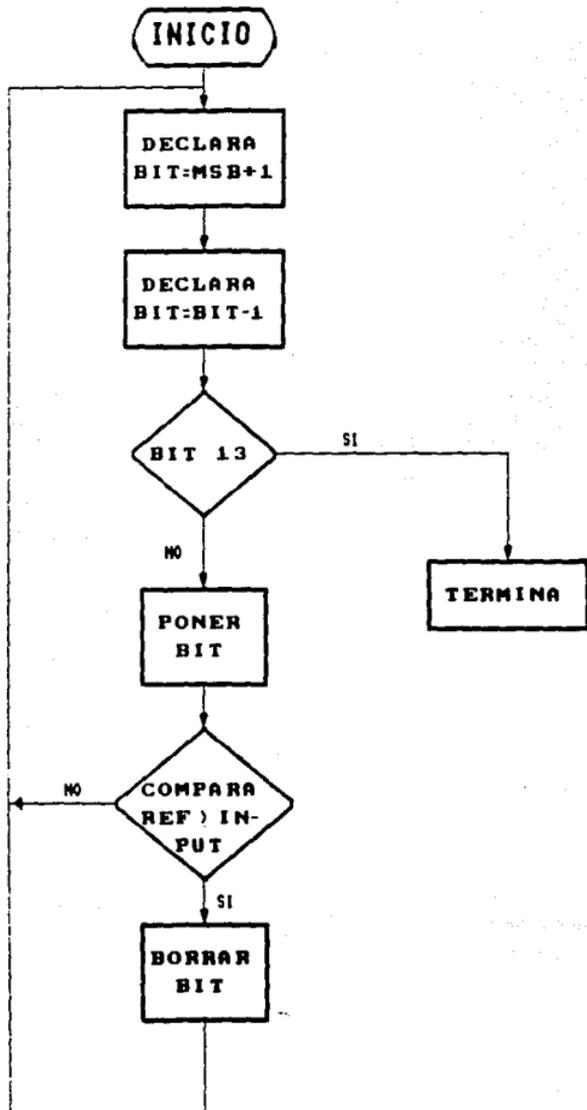


FIG. 7.1 ALGORITMO DE CONVERSION POR APROXIMACIONES SUCESIVAS.

económicos. Un diagrama de bloques de un convertidor A/D basado en 8031 se muestra en la figura 7.2.

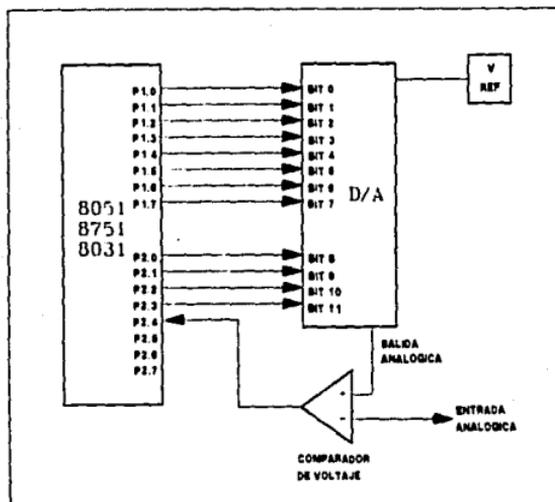


Fig. 7.2 Diagrama de Bloques de un ADC por Aproximaciones Sucesivas

Muchos convertidores A/D industriales requieren de una exactitud de 12 bits. Un convertidor de 12 bits provee un buen "rango dinámico" y es capaz de contar con una resolución de $1/4096$. Si el voltaje de entrada aplicado está entre 0 y 10 V, un convertidor de 12 bits tendría una resolución de 2.4 milivolts, dentro de dicho rango. La exactitud teórica de un convertidor de 12 bits es de $0.024\% \pm 1/2 \text{ LSB}$.

El poderío de un 8031 en este tipo de aplicaciones se revela más fácil examinando el software requerido para implementar el algoritmo de aproximaciones sucesivas. La rutina para el 8031 se muestra en la Tabla 7.1.

	JNB P2.4,L5	3	2
	CLR P1.7	2	1
L5:	SETB P1.6	2	1
	JNB P2.4,L6	3	2
	CLR P1.6	2	1
L6:	SETB P1.5	2	1
	JNB P2.4,L7	3	2
	CLR P1.5	2	1
L7:	SETB P1.4	2	1
	JNB P2.4,L8	3	2
	CLR P1.4	2	1
L8:	SETB P1.3	2	1
	JNB P2.4,L9	3	2
	CLR P1.3	2	1
L9:	SETB P1.2	2	1
	JNB P2.4,L10	3	2
	CLR P1.2	2	1
L10:	SETB P1.1	2	1
	JNB P2.4,L11	3	2
	CLR P1.1	2	1
L11:	SETB P1.0	2	1
	JNB P2.4,L12	3	2
	CLR P1.0	2	1

L12:CONVERSION TERMINADA

TOTAL	90	46 μ s
-------	----	------------

Nota: los tiempos típicos son

Peor caso = 52 μ s

Mejor caso = 40 μ s

Los tiempos de ejecución dados asumen un reloj oscilador a 12 MHz. Comparando estos con la siguiente rutina de un Z-80 a 4 MHz, que ejecuta el mismo algoritmo con el hardware de conversión D/A conectado a un puerto de E/S, se muestra en la Tabla 7.2 (asuma que todos los bits en el PUERTO3 están aterrizados, exceptuando la entrada del comparador).

Tabla 7.2. Rutina de aproximaciones sucesivas para el Z-80.

INSTRUCCION	BYTES	TIEMPO
-------------	-------	--------

BORRA TERMINALES DE LOS PUERTOS

LD A,0	2	1.75
OUT (PUERTO1),A	2	2.75
OUT (PUERTO2),A	2	2.75

COMIENZA LA CONVERSION

LD A,08H	2	1.75
----------	---	------

	OUT	(PUERTO2),A	2	2.75
	IN	A,(PUERTO3)	2	2.75
	OR	A	1	1.00
	IN	A.(PUERTO2)	2	2.75
	JP	Z,L1	3	2.50
	AND	0F7H	2	1.75
L1:	OR	04H	2	1.75
	OUT	(PUERTO2),A	2	2.75
	IN	A,(PUERTO3)	2	2.75
	OR	A	1	1.00
	IN	A,(PUERTO2)	2	2.75
	JP	Z,L2	3	2.50
	AND	0FBH	2	1.75
L2:	OR	02H	2	1.7

Repitase 10 veces más entre L1 y L2, e inicialice los bits de E/S apropiados

TOTAL	179	180 μ s
-------	-----	-------------

Nota: los tiempos típicos son

Peor caso = 190.25 μ s

Mejor caso = 169.25 μ s

Se puede argumentar que "mapeando en memoria" los puertos de E/S del Z-80 se pueden mejorar los tiempos de ejecución, ya que el usuario puede hacer uso de las instrucciones SET y RESET para el manejo de bits en memoria. En realidad sólo se llegan a ahorrar unos cuantos bytes de la memoria, pero muy poco tiempo. Esto es porque las instrucciones del Z-80 orientadas al manejo de bits en la memoria son muy lentas, requiriendo entre 3 y 5 μ s con un reloj de 4 MHz.

Esto no significa que el Z-80 no sea un buen procesador de 8 bits. La limitante es que las decisiones (v.gr. saltos) no se pueden llevar a cabo directamente con el estado de una terminal de E/S. En la mayoría de los procesadores, las instrucciones de salto se realizan según el estado de banderas, después de algún tipo de operación lógica o aritmética. Esto significa que la información se debe mover a un registro interno del CPU antes de que se pueda realizar una decisión. Este "movimiento" de información entre registros internos y E/S hace bastante ineficiente al microprocesador, comparado con el microcontrolador, al involucrarse el manejo de E/S. Notese que el algoritmo del 8031 nunca "mueve" datos de una localidad a otra, directamente fija, prueba y borra los bits. Esta característica le dá al 8031 su notable ventaja de ejecución.

Otra ventaja del 8031, para este tipo de aplicaciones, tiene que ver con el hecho de que las terminales de E/S de los puertos se pueden fijar, borrar, complementar y probar con la misma velocidad que un microprocesador manipula sus registros internos. Notese que el 8031 requiere de 1 μ s para hacer la búsqueda de una instrucción y fijar o borrar una terminal de un puerto. Un microprocesador debe primero buscar y decodificar el código de operando, después fijar la dirección de memoria o E/S en el bus, y después realizar la operación necesaria. Toda esta "comunicación" en el bus del microprocesador retarda significativamente al microprocesador.

7.3. CONVERTIDOR INTEGRADOR DE PENDIENTE DOBLE

Convertidores A/D integradores operan mediante un método indirecto que convierte un voltaje en un período de tiempo, para después medir el período de tiempo con un contador. Las técnicas de integración son algo lentas, en comparación con aproximaciones sucesivas, pero son capaces de entregar mediciones muy exactas (5 o más dígitos decimales), si las adecuadas técnicas analógicas son empleadas. También tienen la ventaja de permitir un período de integración múltiplo de 60 Hz (16.67 ms), con lo que se puede eliminar las inexactitudes causadas por la siempre presente "línea de alimentación". Casi la totalidad de los voltímetros digitales emplean algún tipo de técnica de integración. La figura 7.3 es un diagrama de bloques de un convertidor A/D típico.

Las figuras 7.4a, 7.4b y 7.4c muestran las tres fases involucradas en la técnica de pendiente doble. La figura 7.4a ilustra la fase de autoajuste a cero. En esta fase el lazo integrador se cierra y el offset del integrador analógico se acumula en el capacitor de auto cero. En la figura 7.4b, el interruptor de entrada se cierra y el integrador integra el voltaje de entrada para un período de tiempo fijo T_1 . En la figura 7.4c, el interruptor de referencia se cierra y el integrador integra el voltaje de referencia hasta que el comparador sensa una condición de cruce por cero. El tiempo que toma para que esta fase ocurra es directamente proporcional a la amplitud del voltaje de entrada. Se pueden agregar circuitos adicionales para determinar la polaridad del voltaje de entrada, y posteriormente conmutar a una referencia de polaridad contraria, pero la técnica básica permanece igual.

El 8031 es un controlador ideal para un sistema A/D integrador inteligente. Los temporizador/contadores de 16 bits pueden proveer más que 4.5 dígitos decimales de exactitud, el puerto serial se puede emplear para enviar la lectura analógica a una impresora u otro procesador, el CPU puede ser interrumpido por la línea de 60 Hz para

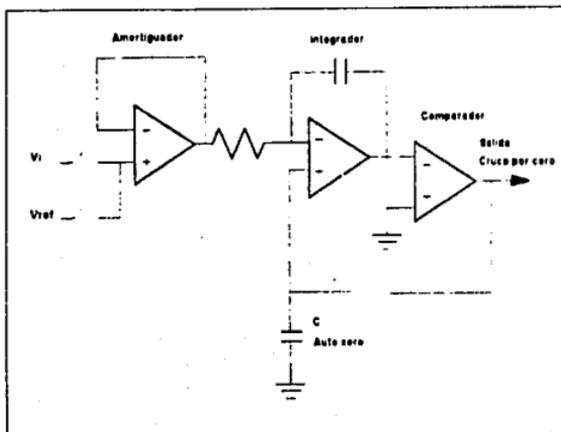


Fig. 7.3 ADC Integrador

que las conversiones comiencen en intervalos precisos, y se puede emplear software para calcular y guardar promedios, máximos o lecturas RMS.

Otra ventaja de este tipo de convertidor es que no se necesitan muchas terminales de E/S para controlar el hardware del convertidor A/D, con lo que opto-acopladores pueden ser empleados para aislar por completo al sistema digital del hardware analógico. Los opto-acopladores proveen una ventaja adicional ya que pueden proveer un corrimiento del nivel lógico si lo requiere la circuitería analógica. La figura 7.5 muestra como un 8031 puede ser conectado al sub-sistema analógico. En la práctica, los interruptores analógicos pueden ser cualquier cosa dentro del rango de los CMOS hasta los VFETs. El código necesario para generar la función básica de conversión A/D se muestra en la Tabla 7.3.

Las interrupciones de los temporizadores pueden ser empleadas para que el CPU pueda realizar otras cosas mientras que la conversión esté en proceso. Notese que se requiere de muy poco tiempo para realizar la eventual conversión A/D.

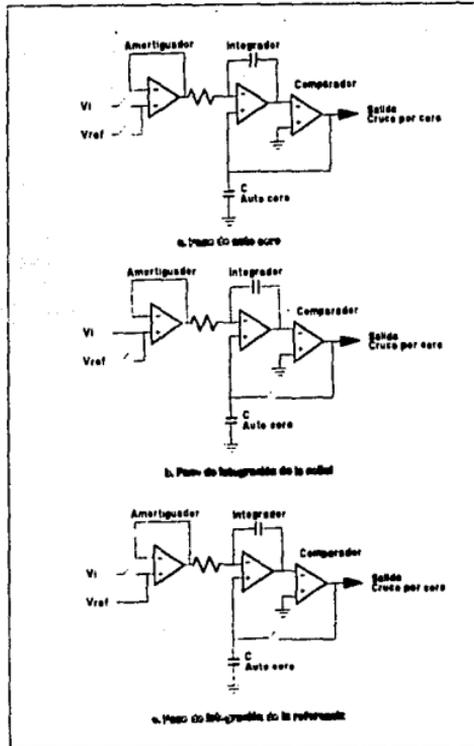


Fig.7.4 Fases del ADC Integrador

Tabla 7.3. Software para el convertidor A/D integrador

;

;Comienza el programa

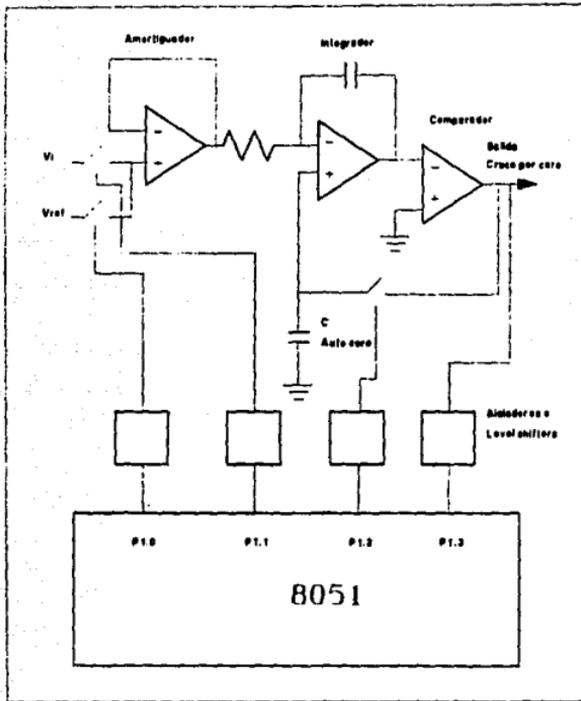


Fig. 7.5 Típico Subsistema Analógico, Controlado por 8051

CLR TR0 ;apaga temporizador

;

MOV TH0,#ALTO TAC ;carga auto cero

MOV TL0,#BAJO TAC ;tiempo

;

ANL P1,#0F0H ;desactiva el A/D

```
SETB P1.2           ;fase de auto cero
SETB TR0           ;enciende el temporizador
JNB TF0,$          ;espera un overflow
;
CLR TR0           ;apaga temporizador
CLR TF0           ;inicializa bandera TOV
;
MOV TH0,#ALTA INTT ;carga integración
MOV TL0,#BAJA INTT ;tiempo
;
CLR P1.2          ;termina auto cero
SETB P1.1         ;comienza integración
SETB TR0         ;comienza temporizador
JNB TF0,$        ;espera un overflow
;
CLR P1.1          ;termina la integración
;
;Ahora integra la referencia
;
SETB P1.0
```

;En este punto el temporizador 0 tiene un valor de 2
;el temporizador es igual a cero cuando sucede un

;overflow y se ha incrementado dos veces durante las

;últimas dos instrucciones

;

;Ahora espera un cruce por cero

;

JNB P1.3,\$

;

;Apaga el temporizador

;

CLR TRO

;

;Ahora el temporizador 0 \pm Vin + 3 cuentas

;

7.4. USANDO EL 8031 CON TRANSDUCTORES RESONANTES

7.4.1. Introducción.

La mayoría de los transductores sensantes no son directamente compatibles con los controladores digitales, ya que generan señales analógicas. Algunas compañías de transductores están desarrollando familias comunes de sensores que generan señales directamente compatibles con los sistemas digitales. Estos no son sensores con conversión A/D interconstruida, sino más bien osciladores cuya frecuencia depende de alguna manera de la propiedad física que se esta midiendo (ver apéndice F para un caso comercial).

Estos dispositivos mandan un tren de pulsos rectangulares cuya razón de repetición codifica el valor de la cantidad que se mide. Estos pulsos, en la mayoría de los casos, se pueden alimentar directamente al 8031, el cual mide ya sea la frecuencia o el periodo de la señal entrante, basando la medición, en la exactitud de su propio reloj oscilador.

Cuando se ha terminado con la medición de frecuencia o período, el 8031 realiza, en pequeños intervalos de tiempo, una verificación de seguridad en la medición y la convierte en software dentro de cualquier escala establecida para la cantidad de la medición. La conversión a software puede incluir correcciones para las no-linealidades en la función de transferencia del transductor.

La resolución también se controla por software, y puede variarse dinámicamente para dar cabida a cambios en alguna etapa crítica. Por ejemplo, en un controlador de procesos se puede incrementar la resolución (hacer un ajuste "fino" en el control) al estarse aproximando el proceso a su objetivo.

La tecnología se aplica a cualquier tipo de parámetro medible: presión, densidad de gas, posición, temperatura, fuerza, etc. El transductor opera en un rango del voltaje de 3V a 20V, por lo que el sensor y el microcontrolador pueden operar de la misma fuente de suministro de voltaje. En 5V, el transductor consume menos de 5 mA, por lo que el sensor, en la mayoría de los casos, se puede conectar directamente a una terminal de los puertos del microcontrolador (ver figura 7.6).

La frecuencia nominal de referencia de la señal de salida de estos dispositivos está en el rango de 20 Hz a 500 kHz, dependiendo del diseño. Un cambio en la medición fuera de estas condiciones de referencia ocasiona un corrimiento en la frecuencia por una cantidad relacionada con el cambio del valor de medición. El microcontrolador detecta el cambio en la frecuencia o período y lo convierte en software según el valor medido.

7.4.2. Conectando el Transductor Digital al 8031.

Normalmente la salida del transductor se puede conectar directamente a alguna terminal del 8031. Una excepción ocurrirá cuando la señal del transductor no se restringe a un rango de voltaje de -0.5 a +5.5 V.

El 8031 no es sensible a los tiempos de subida y bajada de sus señales de entrada. Detecta las transiciones muestreando sus terminales de puertos en intervalos fijos (una vez cada ciclo de máquina), y responde a un cambio según la secuencia de las muestras. Si la razón de "slew" de la señal del transductor es muy baja, el ruido superpuesto en la señal puede causar una secuencia de señales que muestre transiciones falsas. Por ello en ocasiones la señal del transductor se debe pasar por un "Schmitt trigger" para cuadrarla.

7.4.3. Estructura Temporizador/Contador en el 8031.

El 8031 tienen dos temporizador/contador de 16 bits: temporizador 0 (timer 0) y temporizador 1 (timer 1). Ambos se pueden configurar por software para operar como temporizadores o como contadores de eventos.

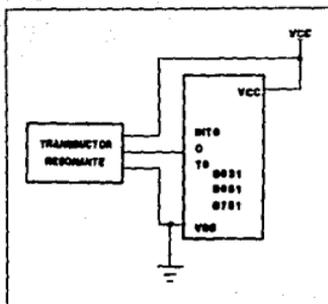


Fig.7.6 Conexión de un Transductor Resonante

En la función de "temporizador", el registro se incrementa automáticamente cada ciclo de máquina. Ya que el ciclo de máquina en el 8031 consiste de 12 períodos de reloj, el temporizador se incrementa a una razón constante de 1/12 la frecuencia del reloj.

En la función "contador", el registro se incrementa como respuesta a una transición de 1 a 0 en su correspondiente terminal de entrada externa (T0 o T1). Esta función trabaja al muestrear la terminal externa cada ciclo de máquina, y cuando las muestras son un 1 en un ciclo y un 0 en el siguiente, la cuenta se incrementa.

Notese que se necesita de dos ciclos de máquina (24 períodos de reloj) para reconocer una transición 1 a 0, la razón máxima de conteo es de 1/24 la frecuencia del reloj. Si la frecuencia de reloj es de 12 MHz, la razón máxima de conteo será de 500 kHz. No existen restricciones en el ciclo de trabajo de la señal a contar.

7.4.4. Criterio para Medir la Frecuencia o el Período.

El medir frecuencia requiere de contar pulsos del transductor para un tiempo de muestreo fijo. El medir el período requiere el medir un intervalo de tiempo para un número fijo de pulsos del transductor. Para un nivel de exactitud dado, en la determinación del valor medido, usualmente es más rápido medir el período que la frecuencia, aún cuando la medición sea proporcional a la frecuencia y no al período.

Ambos temporizador/contador se pueden emplear, uno para marcar el tiempo y el otro para contar los pulsos del transductor. Si la frecuencia contada no se excede de los 50 kHz, se puede conectar la señal del transductor a una terminal de interrupción y contar los pulsos del transductor con software. Esto libera un temporizador, con bajo costo en el tiempo del CPU.

7.4.5. Mediciones de la Frecuencia del Transductor.

El medir frecuencia significa medir pulsos de transductor para un tiempo de muestreo deseado. La cuenta que se obtiene directamente es TxF , donde T es el tiempo de muestreo y F es la frecuencia. El rango para la escala completa es $Tx(F_{max}-F_{min})$. Para una resolución de n bits

$$1 \text{ LSB} = \frac{Tx(F_{max}-F_{min})}{2^n}$$

Por ello el tiempo de muestreo requerido para una resolución de n bits es

$$T = \frac{2^n}{Tx(F_{max}-F_{min})}$$

Por ejemplo, una resolución de 8 bits en la medición de una frecuencia que varía entre 7 kHz y 9 kHz requerirá, según la fórmula, un tiempo de muestreo de 128 ms. La cuenta de frecuencia máxima aceptable será $128 \text{ ms} \times 9 \text{ kHz} = 1152$ conteos. El mínimo será 896 conteos. Restando 896 de cada cuenta de frecuencia permitirá que se reporte la frecuencia en una escala de 0 a FF en dígitos hexadecimales.

Para implementar la medición, un temporizador se emplea para establecer el tiempo de muestreo. En esta función se autoincrementa cada ciclo de máquina. El temporizador se inicializa a un valor que lo reinicializa al final del tiempo de muestreo. Entonces genera una interrupción, y la interrupción lee e inicializa el contador de pulsos de transductor, y vuelve a cargar el temporizador con el valor de inicialización adecuado.

El valor inicial es el complemento a 2 negativo del tiempo de muestreo en ciclos de máquina. El tiempo de muestreo se puede convertir a ciclos de máquina multiplicándolo por $(F_{xtal})/12$, donde F_{xtal} es la frecuencia del 8031. Por ejemplo, si la frecuencia del reloj es de 12 MHz, entonces el tiempo de muestreo es de 128 ms

$$(128 \text{ ms}) \times (12000 \text{ kHz})/12 = 128000 \text{ ciclos de máquina}$$

Entonces el valor inicial requerido para provocar la interrupción en los 128 ms es :

$$-128000 = FE0C00, \text{ en dígitos hexadecimales}$$

Notese que el valor inicial tiene un ancho de 3 bytes mientras que el ancho del temporizador es de 2 bytes. Esto significa que el temporizador se debe aumentar a tres

bytes con software en la rutina de interrupción del temporizador. El 8031 tiene una instrucción DJNZ (decrementa y salta si no es cero) la cual facilita codificar el tercer byte para que cuente hacia abajo en vez de hacia arriba. Si el tercer byte del temporizador cuenta hacia abajo, su valor inicial es el complemento a 2 de lo que sería para un contador hacia arriba. Por ejemplo, si el complemento a 2 del tiempo de muestreo es FE0C00, entonces el valor inicial para el tercer byte del temporizador será 02, en vez de FE. La rutina de interrupción del temporizador podría ser:

RUTINA_INTERRUPCION_TIMER:

DJNZ TERCER_BYTE,SALE

MOV TLO,#0

MOV TH0,#0CH

MOV TERCER_BYTE,#2

MOV FRECUENCIA,CONTADOR_LO

;Inicializar CONTADOR a -896:

MOV CONTADOR_LO,#80H

MOV CONTADOR_HI,#0FCH

SALE: RETI

En este punto el valor de la frecuencia de la señal del transductor, medido en una resolución de 8 bits, está contenido en la variable "frecuencia". Notese que el temporizador se puede inicializar sobre la marcha. Notese también que para una resolución de 8 bits sólo es necesario leer el byte inferior del contador de frecuencia, ya que el byte superior necesariamente es 0. Sin embargo, uno puede verificar que el byte superior sea cero, para mayor seguridad en los datos.

El problema con medir la frecuencia no sólo radica en su lentitud, sino que es un desperdicio del poderío de los temporizadores del 8031. Un temporizador con una resolución de microsegundos se emplea para marcar períodos de tiempo de 100 ms. De cualquier modo la técnica no sólo es útil si el temporizador atiende otros propósitos, con lo que el tiempo de muestro sale gratis. Pero en la mayoría de los casos es más rápido e igualmente exacto medir la frecuencia derivándola de una medición del período.

7.4.6. Mediciones del Período del Transductor.

Medir el período de una señal del transductor significa medir el tiempo total elapsado sobre un número, N, de pulsos del transductor. La cantidad que se mide directamente es NT, donde T es el período de la señal del transductor en ciclos de máquina. La relación entre T en ciclos de máquina y la frecuencia del transductor F en unidades arbitrarias de frecuencia es

$$T = \frac{F_{xtal}}{F} \times (1/12)$$

donde F_{xtal} es la frecuencia del reloj del 8031, en las mismas unidades que F.

El rango para la escala completa es entonces $N \times (T_{max} - T_{min})$. Para una resolución de n bits

$$1 \text{ LSB} = \frac{N \times (T_{max} - T_{min})}{2^n}$$

Por ello el número de períodos sobre los cuales el tiempo elapsado debe ser medido es

$$N = \frac{2^n}{T_{max} - T_{min}}$$

Sin embargo, N también tiene que ser un entero. La fórmula anterior se puede evaluar lógicamente (sin olvidar que T_{max} y T_{min} están en ciclos de máquina) y seleccionar N para el siguiente entero superior. Esta selección da una medición de período que tiene algo más que una resolución de n bits, pero en caso necesario se puede regresar a la resolución de n bits mediante

$$\text{valor reportado} = \frac{NT - NT_{min}}{NT_{max} - NT_{min}}$$

donde NT es el tiempo elapsado sobre N períodos.

El cálculo se puede realizar matemáticamente si se cuenta con una adecuada rutina de división. Para una resolución de 8 bits es razonable buscar el valor reportado en una tabla.

Por ejemplo, supongase que se desea una resolución de 8 bits en la medición del período de una señal cuya frecuencia varía entre 7.1 kHz y 9 kHz. Si la frecuencia del reloj es de 12 MHz, entonces la T_{max} es $(12000 \text{ kHz}/7.1 \text{ kHz}) \times (1/12) = 141$ ciclos de máquina. T_{min} es 111 ciclos de máquina. El valor requerido para N es, según la fórmula, $256/(141-111) = 8.53$ períodos. Usando N = 9 períodos dará un máximo valor de NT de $141 \times 9 = 1269$ ciclos de máquina. El mínimo valor de NT será $111 \times 9 = 999$ ciclos de máquina. Una tabla se puede utilizar para ubicar estos valores en una escala de 0 a 255, dando con precisión la resolución deseada de 8 bits.

Para implementar la medición, se emplea un temporizador para medir el tiempo elapsado, NT. El transductor se conecta a una de las terminales de interrupción externa (INT0 o INT1), y esta interrupción se configura para el modo de transición de nivel. En este modo cada transición de 1 a 0 en la salida del transductor generará una interrupción. La rutina de interrupción cuenta pulsos de transductor, y cuando llega a la N predeterminada, lee y borra el temporizador. Para el ejemplo anterior, la rutina de interrupción podría ser:

RESPUESTA_INTERRUPCION:

DJNZ N,FUERA

MOV N,#9

CLR EA

CLR TR1

MOV NT_LO,TL1

MOV NT_HI,TH1

MOV TL1,#9

MOV TH1,#0

SETB TR1

SETB EA

CALL TABLA

FUERA: RETI

En esta rutina un contador de pulsos N se decrementa de su valor inicial, 9, a cero. Cuando el contador llega a cero se recarga con 9. Entonces todas las interrupciones se bloquean por un instante mientras el temporizador es leído y borrado. El temporizador se detiene durante las operaciones de lectura y borrado, para que el "borrado" signifique inicializándolo a 9, para compensar los 9 ciclos de máquina que se pierden al estar detenido el temporizador.

Si se habilitan otras interrupciones, la que se conecta con el transductor se debe establecer con prioridad 1 (más alta), y las demás con prioridad 0 (más baja). Esto es para controlar el tiempo de atención a la interrupción. El tiempo de atención a la interrupción no afectará la medición si es el mismo para cada medición. El tiempo de atención se limita al rango de 3 a 8 ciclos de máquina.

La subrutina TABLA se usa para regresar la medición a la resolución de 8 bits. También puede incluir una corrección de errores interna, para errores en las no-linealidades de la función de transferencia del transductor.

Las subrutinas para búsqueda en tablas se facilitan en el 8031 empleando las instrucciones `MOVC A, @A + PC` y `MOVC A, @A + DPTR` para acceder a la tabla. DPTR es un registro del 8031 que sirve como apuntador de datos de 16 bits. Sus dos bytes se pueden direccionar independientemente como DPL (byte bajo) y como DPH (byte alto).

En el ejemplo la tabla está formada por $NT_{max} - NT_{min} = 270$ valores. Comenzando en la dirección de 16 bits referida por INICIO. La subrutina debe encontrar la dirección de la tabla que corresponda al valor medido de NT. Esta dirección es

$$DPTR = INICIO + NT - NTMIN,$$

donde $NTMIN = 999$, para este ejemplo.

TABLA:

PUSH ACC

PUSH PSW

MOV A, #LOW(INICIO-NTMIN)

ADD A, NT_LO

```
MOV    DPL,A
MOV    A,#HIGH(INICIO-NTMIN)
ADDC   A,NT_HI
MOV    DPH,A
CLR    A
MOVC   A,@A+DPTR
MOV    PERIODO,A
POP    PSW
POP    ACC
RET
```

En este punto el valor del período de la señal del transductor, medido con una resolución de 8 bits, está contenido en la variable "periodo".

Es razonable pensar que una tabla con más de 270 valores resulte algo grande. Sin olvidar sus ventajas de velocidad y corrección de errores. Pero en dicho caso un buen algoritmo de división puede ser escrito, haciendo uso a priori del conocimiento sobre el tamaño de los números involucrados en el cálculo.

7.4.7. Mediciones para el Ancho del Pulso.

Los temporizadores del 8031 tienen un modo de operación particularmente apto para este tipo de mediciones y puede ser útil en estas aplicaciones si la señal del transductor tiene un ciclo de trabajo fijo.

En este modo el temporizador se enciende por circuitería propia del integrado en respuesta a un nivel alto en la terminal de interrupción externa, y se apaga con un nivel bajo. La interrupción externa en sí se habilita, para que la misma transición de 1 a 0 del transductor que apaga al temporizador genere una interrupción. Entonces la rutina de interrupción leerá e inicializará el temporizador.

La ventaja de este método es que la señal del transductor tiene acceso directo a la compuerta del temporizador, con el resultado de que variaciones en el tiempo de atención a la interrupción dejan de tener efecto alguno sobre la medición. El temporizador puede ser leído y borrado en cualquier momento antes del siguiente nivel alto en la salida del transductor.

7.4.8. Exactitud y Resolución.

La exactitud con la que el 8031 medirá la frecuencia o el período de la señal del transductor depende de dos cosas: de la exactitud del reloj oscilador y de las variaciones en el tiempo de atención a las interrupciones.

Como normalmente la señal de reloj se genera por un oscilador de cristal, la exactitud del oscilador normalmente excede el error de cuantización inherente en la resolución finita de n bits.

Como se mencionó previamente, el tiempo de atención a la interrupción no introduce un error, por sí mismo, en la medición, pero sí pueden las variaciones en el tiempo de atención a la interrupción. El tiempo de atención a la interrupción en el 8031 puede variar de 3 a 8 ciclos de máquina, dependiendo de que instrucción se esté ejecutando en el momento de suceder la interrupción. Esto representaría un error de ± 5 cuentas en el valor de NT medido en una medición de período. Un error de ± 5 cuentas de NT no necesariamente se traduce en ± 5 LSBs en el resultado final, pero aún así representa un error que excede a la resolución.

En una medición directa de frecuencia las variaciones en el tiempo de atención a la interrupción representarán un error de $\pm 5 \mu s$ en el tiempo de muestreo.

Si este tipo de errores son inaceptables existen formas de manejarlos. Para mediciones del período, cuando el ciclo de trabajo del transductor es constante, puede ser empleada la técnica de medición del ancho de pulso, anteriormente descrita. Su ventaja es la de apagar el temporizador al generarse la interrupción, en vez de cuando se da atención a la interrupción.

En otros casos uno puede simplemente incrementar el tiempo de muestreo arriba del mínimo requerido para obtener la resolución deseada. Por ejemplo, si la medición requiere de una resolución de 8 bits, uno puede diseñar el software para una resolución de 11 bits y truncar el resultado a 8 bits.

CAPITULO 8

8. DISEÑO DE SISTEMAS CON MICROCONTROLADORES PARA AMBIENTES ELECTRICAMENTE RUIDOSOS

Por lo general se piensa que los circuitos digitales son inmunes a los problemas de ruido, pero en realidad no lo son. Los ruidos en los sistemas digitales producen fallas en el software: saltos del programa a localidades aparentemente aleatorias en la memoria. El ruido inducido en las líneas de las señales pueden provocar dichas fallas, aunque el voltaje de suministro es más sensible a dicho ruido que las líneas de señalización.

Condiciones severas de ruido, aquellas que involucran descargas electrostáticas o aquellas encontradas en ambientes automotrices, pueden provocar un daño permanente en el hardware. Las descargas electrostáticas (DES) pueden formar un cráter en la pastilla de silicio. En el ambiente automotriz, y bajo operación ordinaria, la línea de suministro de "12V" puede tener transitorios de $\pm 400V$.

8.1. SINTOMAS DE LOS PROBLEMAS DE RUIDO

Los problemas de ruido por lo general no suceden en la etapa de desarrollo de un sistema con microcontroladores. Esto es porque un laboratorio común rara vez simula el ambiente destino. Los problemas de ruido tienden a surgir al instalar y operar el sistema en su ambiente destinado. Pueden tener una causa conocida, tal como una DES del dedo de un operador en algún teclado o la falla ocurre siempre que se enciende y se apaga una fotocopiadora. O una causa desconocida, y nada que el operador realice puede provocar nuevamente el problema. Pero unos minutos, horas o días más tarde, el problema vuelve a ocurrir.

Un síntoma de los problemas por ruido eléctrico es el hecho de ser aleatorios, tanto en su ocurrencia como en lo que es lo que el sistema hace al ocurrir la falla. Todas las fallas operacionales que ocurren en intervalos aparentemente aleatorios, no necesariamente son causados por ruido en el sistema. Un voltaje de polarización marginal, un desacoplamiento inadecuado, condiciones de software raramente encontradas o coincidencias en la temporización pueden producir fallas que parecen ser aleatorias. Por otro lado, algunas fuentes de ruido pueden producir fallas periódicas. Sin embargo, entre más difícil resulte encontrar la relación causa-efecto de una falla, lo más seguro sea que se trate de una falla producida por ruido.

8.2. TIPOS Y FUENTES DE RUIDO ELECTRICO

El nombre que se le dá a los ruidos eléctricos diferentes a aquellos inherentes a los componentes del circuito (tales como el ruido térmico) es interferencia electromagnética: IEM. Fuentes de IEM son motores, interruptores de potencia, luces fluorescentes, descargas electrostáticas, etc. Los tipos de IEM se describen a continuación.

8.2.1. Transitorios en la Línea de Suministro.

Cualquier cosa que conmute altas cargas de corriente hacia o desde líneas de suministro de CA o CD, ocasiona grandes transitorios en estas líneas de potencia. Por ejemplo, encender una máquina eléctrica de escribir, puede introducir un pico de 1000V en la línea de suministro de CA.

El mecanismo básico detrás de los transitorios en la línea de suministro reside bajo la igualdad $v = L(di/dt)$, donde L es la inductancia de la línea de suministro entre la fuente de poder y la carga. El paso de corriente a través del circuito genera un campo magnético en la inductancia, por lo que al disminuir la carga, el campo debido a dicha componente de línea se colapsa, generando transitorios de voltaje, los cuales tratan de mantener la corriente en su nivel original. Esto se conoce como "golpe inductivo". Y los transitorios se generan tanto al agregar carga como al disminuirla, pero los efectos son más severos al ocurrir lo último.

Existen otros mecanismos para transitorios en la línea, aquellos que involucran ruido externo inducido en las líneas. Los voltajes de ruido se suman al del suministro afectando circuitos susceptibles.

8.2.2. Pulsos Electromagnéticos (PEM) e Interferencia por Radio-Frecuencia (IRF).

Cualquier cosa que produzca un arco o una chispa radiará pulsos electromagnéticos (PEM) o interferencia por radio-frecuencia (IRF).

Las descargas por chispa probablemente han ocasionado más fallas de software en equipo digital que cualquier otra fuente de ruido. El mecanismo de falla es el PEM producido por la chispa. El PEM induce transitorios en el circuito, que son la causa fundamental de la falla.

Los arcos y las chispas ocurren en sistemas de ignición automotriz, motores eléctricos, interruptores, descargas estáticas, etc. Los motores eléctricos que cuentan con barras o delgas en el conmutador producen un arco al pasar el cepillo de una delga a otra. Los motores de CD y los "universales" (CA/CD) que se emplean en herramientas de potencia manuales, son el tipo que cuenta con delgas. En los interruptores, el mismo golpe inductivo que pone transitorios en la línea formará una chispa al abrirlo o cerrarlo.

8.2.3. Descarga Electroestática (DES).

La descarga electrostática (DES) es la chispa que ocurre cuando una persona acumula carga estática al caminar a través de una alfombra, y la descarga en un teclado o cualquier cosa que pueda tocar. Al caminar a través de una alfombra en un clima seco, una persona puede acumular un voltaje estático de 35kV. El pulso de corriente de una descarga electrostática tiene un tiempo de subida extremadamente rápido, típicamente 4A/ns.

Obviamente, la sensibilidad la DES se debe de considerar en el diseño de equipo que estará sujeto a el, tal como el equipo de oficina.

8.2.4. Corriente de Tierra.

Corrientes en las líneas de tierra son otra fuente de ruido. Estas pueden ser corrientes de 60 Hz, IRF o mezcla de señales que comparten la misma línea de tierra. El problema reside en que los nodos que comparten un lazo de tierras no siempre se encuentran al mismo potencial.

8.2.5. Ruido "Radiado" y Ruido "Conducido".

El ruido radiado es aquel que llega al circuito víctima en la forma de radiación electromagnética, tal como PEM e IRF. Ocasiona problemas al inducir voltajes extraños en el circuito. El ruido conducido es aquel que llega al circuito víctima como voltaje extraño, típicamente por vía de las líneas de suministro de CA o CD.

Uno se puede defender en contra del ruido radiado con un buen diseño en la distribución de partes y utilizar técnicas de blindaje. Y contra el ruido conducido con filtros y supresores, pero también son importantes las técnicas de distribución y aterrizaje.

8.3. SIMULANDO EL AMBIENTE

Una propuesta muy cara es corregir los problemas de ruido después de haber sido completado el diseño de un sistema. Pero también los daños generados por fallas son caros. Es más barato en el largo plazo invertir un poco de tiempo y dinero para aprender sobre el ruido y equipo de simulación de ruido, para que pruebas controladas se puedan realizar en el laboratorio el desarrollarse el diseño.

Simular el ambiente destino con ruido requiere de dos etapas. La primera consiste en reconocer el ambiente ruidoso, esto es, es necesario conocer que tipos de ruido están presentes y cuáles van a ocasionar problemas. La segunda consiste en generar ruido eléctrico de una manera controlada. Esta segunda etapa es más difícil de lo que aparenta

ser; en principio uno imagina la simulación en términos de un generador de funciones y algunas partes extras, para luego notar que es necesario un amplificador de potencia de banda-ancha con un rango dinámico de 200V.

8.4. TIPOS DE FALLAS Y MECANISMOS DE FALLA

Un gran problema que puede causar la IEM en un sistema digital es una mala operación intermitente. Estas fallas en el software ocurren cuando el sistema está operando y una fuente de IEM se hace presente, y por lo general se caracterizan por pérdida de información o un salto en la ejecución del programa a alguna localidad de memoria desconocida. Por lo general no se daña el software, y se regresa a la operación normal al eliminar la IEM, con un "reset" manual o automático, o reestableciendo la información perdida.

Un mecanismo de falla por DES tiene que ver con el voltaje de polarización en el sustrato de la pastilla. Si la polarización en el sustrato oscila por valores lejanos al nominal, suceden errores en la lectura del programa. Además de un daño permanente en el óxido por el incremento de la corriente de drenado, con la que se reduce el voltaje en el sustrato a un valor marginal o inaceptable. Con lo que el sistema pierde confiabilidad u operabilidad hasta que se reponga el CPU. Si el CPU ha sufrido una descarga de este tipo, lo más seguro es que ocurrirá de nuevo.

Algunas pastillas tienen un sustrato aterrizado y pueden soportar algún daño en el óxido sin quedar inoperables. En este caso el daño está presente y existe una mayor corriente de drenado, sin embargo, como el sustrato mantiene su voltaje nominal, el daño permanece oculto.

Por ello se debe reconocer que el conectar las patas, sin proteger, de un puerto a un teclado o cualquier cosa que sea objeto de descargas electrostáticas, forma una configuración muy peligrosa.

Es importante notar que las pastillas MOS no son los únicos componentes susceptibles a daño permanente por una DES. Las pastillas bipolares y lineares también se pueden dañar de esta manera. Las junturas PN son candidatas a un mecanismo de falla conocido como ruptura térmica secundaria, en donde un pico de corriente hace que puntos microscópicos en la juntura alcancen temperaturas de fundición. Los TTL de baja potencia son susceptibles a esto, así como los amplificadores operacionales. Estos últimos, además, tienen capacitores MOS en la misma pastilla, conectados a través de una combinación de patas externas, y estos son susceptibles a ruptura del dieléctrico.

Los transitorios por ruido pueden hacer que una pastilla falle por cualquier pata, inclusive una pata de salida, ya que cada pata se conecta al sustrato por medio de una

juntura pn. Sin embargo, la pata más vulnerable es la de VCC, por tener acceso directo a todos los componentes de la pastilla: cada registro, compuerta, flip-flop y buffer.

Un menú de posibles mecanismos de falla es bastante largo. Un transitorio en el sustrato en el momento inoportuno generalmente causa un error en la lectura del programa. Un nivel falso en una entrada de control puede ocasionar una búsqueda de instrucción errónea. Un disturbio en la línea de alimentación puede modificar un bit en el contador de programa o el registro de instrucciones. Una pequeña interrupción o inversión en la polaridad de la línea de alimentación, puede apagar el procesador, pero no lo suficiente como para que el capacitor de reset se descargue. Por lo que al terminar el transitorio, el circuito integrado (c.i.) comienza de nuevo sin un reset.

Es necesario recalcar que los mecanismos de falla tienen que ver con la llegada de transitorios inducidos por ruido en las patas de los c.i., en vez de la generación de pulsos ruidosos dentro del c.i., esto es, no es el c.i. quien recoge el ruido, es el circuito.

8.5. MÉTODOS PREVENTIVOS

La prevención es por lo general una solución más económica que la supresión, por lo que primero se considerarán algunos métodos preventivos que ayudarán a la minimización de la generación de ruidos en el sistema.

8.5.1. Lazos de Corriente.

La geometría física de cualquier lazo de corriente es la clave del porque genera IEM, porque es susceptible al IEM y como blindarlo. En específico, lo que importa es el área del lazo.

Para reducir tanto la generación de IEM como susceptibilidad al IEM, la regla a seguir es la de minimizar el área del lazo. Esto es lo mismo que decir que la inductancia de un circuito debe ser minimizada. La inductancia es por definición la constante de proporcionalidad entre la corriente y el campo magnético que produce: $\phi = LI$. Manteniendo los cables de alimentación y de retorno lo más próximos, para promover la cancelación de campo magnético, corresponde a minimizar el área del lazo o minimizar la L . Y, como la inductancia mutua de un lazo de corriente A a un lazo de corriente B es la misma que la inductancia mutua del lazo B al lazo A, los campos de ambos lazos tienden a cancelarse. Es importante notar que un circuito que no radia interferencia tampoco la recibe.

8.5.2. Blindaje.

Existen tres tipos principales de blindaje: blindaje contra acoplamiento capacitivo, blindaje contra acoplamiento inductivo y blindaje por RF.

8.5.2.1. Blindaje contra acoplamiento capacitivo.

El acoplamiento capacitivo involucra el paso de señales de interferencia a través de capacitancias mutuas o "parásitas" que no se muestran en el diagrama del circuito, pero de que se sabe su existencia. Como ejemplo, este tipo de acoplamiento causa, en un sistema digital, interferencia por intermodulación en cables de varios conductores.

Para bloquear el acoplamiento capacitivo es necesario cubrir el circuito o conductor a proteger con un blindaje metálico. Esto se conoce como blindaje electrostático o de Faraday. Si la cobertura es del 100%, no se tiene que aterrizar el escudo, pero por lo general siempre se aterriza, para garantizar que las capacitancias entre el circuito y el escudo vayan a la tierra de referencia de señal, en vez de actuar como elementos de retroalimentación y de intermodulación. Para el caso de circuitos electrónicos dispuestos en tarjetas, se puede insertar un plano aterrizado entre ellas para eliminar la mayor parte del acoplamiento capacitivo.

8.5.2.2. Blindaje contra acoplamiento inductivo.

Con un acoplamiento inductivo, el mecanismo físico involucrado es la densidad de flujo magnético, B , de alguna fuente externa de interferencia, que se enlaza con un lazo de corriente en el circuito víctima, y genera un voltaje en el lazo de acuerdo con la ley de Lenz: $v = -NA(dB/dt)$, donde en este caso $N = 1$ y A es el área del lazo de corriente en el circuito víctima.

Existen dos maneras de defender un circuito contra el acoplamiento inductivo. Una es minimizar los campos ofensivos en sus orígenes. Esto se logra minimizando el área del lazo de corriente en la fuente, para así promover la cancelación de campos. La otra manera consiste en minimizar el lazo de corriente del circuito víctima, ya que mediante Lenz, el voltaje inducido es proporcional a dicha área.

Como ejemplos de éste "blindaje" tenemos a: el cable coaxial, el par de hilos, el plano de tierras y la distribución de tierras en malla en una tarjeta de circuito impreso (TCI).

a) Cable coaxial

La figura 8.1 muestra un cable coaxial llevando una corriente I desde una fuente de señal hasta una carga receptora. El blindaje lleva la misma corriente que el conductor central. Afuera del blindaje, el campo magnético producido por $+I$ que fluye en el conductor central, se cancela con el campo producido por $-I$ que fluye en el blindaje. El cable, por tanto, muestra una inductancia $\phi = 0$. Esto sólo se cumple si el blindaje lleva la misma corriente que el conductor central.

En aplicaciones reales, tanto la señal fuente como la carga receptora pueden tener una terminal conectada a una tierra común. Es necesario conectar ambos lados del cable a tierra. La figura 8.2a muestra el caso cuando el blindaje del cable sólo se aterriza de un lado. En dicho caso el área del lazo no está bien definido. El lazo de corriente corre por

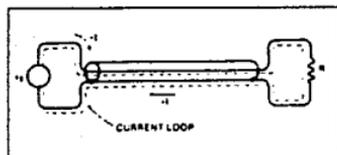
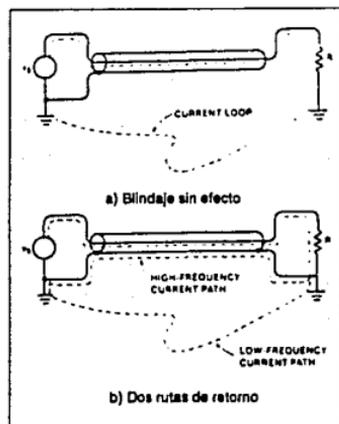
Fig. 8.1 Externo al Blindaje, $\phi = 0$ 

Fig.8.2 Uso de un Cable Coaxial

el conductor central y regresa por la conexión de tierra común. El blindaje no lleva corriente. No se efectúa ninguna cancelación de campo. El blindaje no tiene efectos en cuanto a la generación de IEM o susceptibilidad a la IEM.

La figura 8.2b muestra la situación cuando el cable se aterriza de ambos lados. Dependiendo del contenido en frecuencia de la señal, el blindaje llevará toda la corriente de retorno, o sólo una porción debido al efecto de desvío en la conexión de tierra común. En general, el lazo de corriente seguirá la ruta con menor impedancia. En bajas frecuencias, 0 Hz a unos cuantos kHz, donde la reactancia inductiva es insignificante, la corriente seguirá la ruta con menor resistencia. Arriba de unos cuantos kHz, donde predomina la reactancia inductiva, la corriente seguirá la ruta de menor inductancia. La ruta de menor inductancia es la ruta del lazo con área mínima. Por ello, para frecuencias mayores el blindaje acarrea virtualmente la misma corriente que el conductor central, y es por tanto efectivo contra la generación y recepción de IEM.

Aquí se ha introducido el problema del "lazo de tierra", como se muestra en la figura 8.3a. En un ambiente ruidoso se puede romper el lazo de tierra y mantener aún la efectividad de blindaje del cable coaxial, al insertar un opto-acoplador, como se muestra en la figura 8.3b. Básicamente, lo que logra el opto-acoplador es redefinir la fuente de señal como no aterrizada, con lo que dicho lado del cable no se aterriza, y aún así permite que el blindaje lleve la misma corriente que el conductor central.

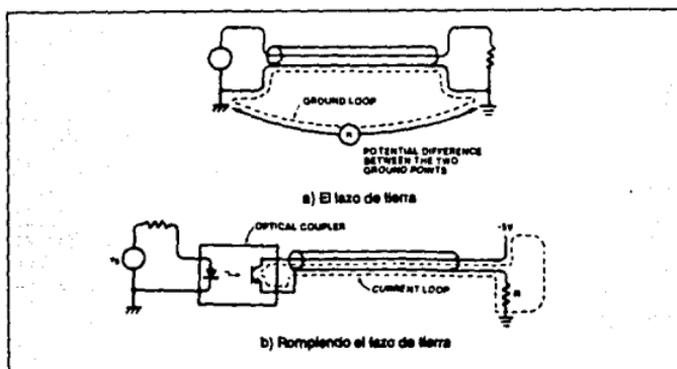


Fig.8.3 Uso de un Optoacoplador

b) Par de hilos

Una manera más económica de minimizar el área de lazo consiste en disponer los cables de ida y de retorno juntos. No es tan efectivo como el cable coaxial. Un cable coaxial ideal agrega un área cero al lazo, mientras que los cables juntos agregan un área finita.

Sin embargo, tiene dos ventajas este método que lo hace tan bueno como el cable coaxial. En primer lugar, el cable coaxial real no es ideal. Si la corriente en el blindaje no se distribuye uniformemente alrededor del conductor central en cada tramo del cable, entonces la cancelación de campo es incompleta. Si dicha cancelación es incompleta, entonces el área efectiva que se agrega al lazo no es cero. En segundo lugar, en el método más económico, los cables de ida y retorno se entrelazan. Esto no solo mantiene su proximidad, sino que el ruido recogido en un lazo tiende a cancelar el ruido recogido en el siguiente lazo de la línea. Por lo que el par de hilos ofrece un blindaje contra acoplamiento inductivo como el cable coaxial.

El par de hilos no ofrece un blindaje contra ruido electrostático (v.gr. blindaje contra acoplamiento capacitivo). El cable coaxial funciona mejor en frecuencias más altas. Esto por que el par de hilos agrega más carga capacitiva a la señal fuente que el cable coaxial. El par de hilos se considera funcionalmente apto hasta frecuencias de 1 MHz, mientras que el cable coaxial llega hasta el GHz.

c) El plano de tierras

La mejor manera de minimizar el área de lazos al estar involucrados varios de ellos es utilizando un plano de tierras. Un plano de tierras es una superficie conductora que sirve como una ruta de retorno para todos los lazos de corriente en un circuito. Normalmente, es una o más capas de una TCI de varias capas. Todos los puntos de aterrizaje van no a un trazo aterrizado en la TCI, sino directamente a el plano de tierras. Esto le dá la libertad a cualquier lazo de tierra para completarse en cualquier configuración que otorgue el menor área posible (para frecuencias en donde la impedancia de la ruta de tierra es primordialmente inductiva). El objetivo es minimizar el almacenamiento de energía en el campo magnético producido por cada lazo de corriente. El mínimo flujo magnético significa área mínima efectiva de lazo y mínima susceptibilidad al acoplamiento inductivo.

d) Distribución de tierras en malla en una TCI

Lo más próximo a un plano de tierras es una malla de tierras, como se muestra en la figura 8.4. Disponiendo trazos horizontales en un lado de la tarjeta y trazos verticales del otro lado, permite el paso de trazos de señal y de alimentación. Donde se cruzan los trazos de tierra verticales y horizontales, se conectan a través de un puente.

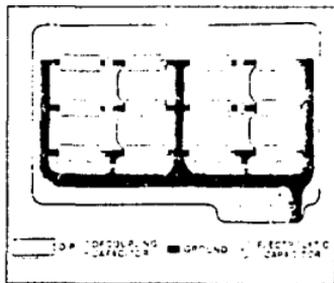


Fig.8.4 TCI con Malla de Tierras

Esta configuración opera casi tan bien como el plano de tierras para minimizar el área de lazo.

8.5.2.3. Blindaje contra IRF.

Un campo eléctrico variable en el tiempo genera un campo magnético variable en el tiempo, y viceversa. Lejos de la fuente de un campo electro-magnético (EM) variable, la razón de las amplitudes de los campos eléctrico y magnético siempre es de 377Ω . Sin embargo, cerca de la fuente de los campos, esta razón es diferente y depende de la naturaleza de la fuente. Cuando la razón está cerca a los 377Ω se le llama campo lejano, y donde la razón es significativamente diferente a 377Ω se le conoce como campo cercano. La razón en sí misma se denomina impedancia de onda, E/H.

El campo cercano se distribuye hasta 1/6 de la longitud de onda desde la fuente. Para 1 MHz esto es como 46 mts y para 10 MHz es como 5 mts. Esto significa que si la fuente de IEM se encuentra en el mismo cuarto con el circuito víctima, lo más seguro sea que se trate de un problema de campo cercano. Esto es importante ya que en el campo cercano, un problema de IRF es causado completamente por acoplamiento de campo eléctrico (campo-E) o acoplamiento de campo magnético (campo-H), y esto influye en la selección del tipo de blindaje contra IRF o bien si es que puede ayudar un blindaje de este tipo.

En el campo cercano de una antena unipolar, la razón E/H es mayor a 377Ω , lo cual significa que es principalmente un generador de campo-E. Una terminal de "wire-wrap" puede ser una antena de este tipo. La interferencia debida a una antena unipolar será por acoplamiento de campo eléctrico, el cual es básicamente acoplamiento capacitivo. Los métodos para proteger un circuito del acoplamiento capacitivo, tales como el blindaje de Faraday, son efectivos contra la IRF de una antena unipolar. Una distribución de tierras en malla no sería tan efectivo.

En el campo cercano de una antena de lazo, la razón E/H es menor de 377Ω , lo cual significa que es principalmente un generador de campo-H. Cualquier lazo de corriente es una antena de lazo. La interferencia debida a una antena de lazo será por acoplamiento de campo magnético, que es básicamente acoplamiento inductivo. Los métodos para proteger un circuito del acoplamiento inductivo, tales como la distribución de tierras en malla, son efectivos contra la IRF de una antena de lazo. Un blindaje de Faraday no sería tan efectivo.

Un caso más difícil de IRF, para campo cercano o lejano, puede que requiera un auténtico blindaje metálico de RF. La idea detrás del blindaje de RF es que los campos variables en el tiempo de la IEM inducen corrientes en el material de blindaje. Las corrientes inducidas disipan energía de dos maneras: pérdidas por I^2R (pérdidas por absorción) en el material de blindaje y pérdidas por radiación (pérdidas por reflexión) al estar radiando sus propios campos EM. La energía para ambos mecanismos proviene de la IEM, por lo que la IEM se debilita al penetrar el blindaje.

El mecanismo principal para blindaje contra campos-H es la pérdida por absorción, y la pérdida por reflexión es el mecanismo principal para blindaje contra campos-E. La pérdida por reflexión, siendo un fenómeno de superficie, es independiente de el grosor

del blindaje. Sin embargo, ambos mecanismos dependen de la frecuencia (ω) de la IEM, y de la permeabilidad (μ) y conductividad (σ) del material de blindaje. Estos mecanismos de pérdida varían de acuerdo a:

$$\text{pérdida de reflexión a un campo-E (en dB)} = \log \frac{\sigma}{\omega \mu}$$

$$\text{pérdida de absorción a un campo-H (en dB)} = t\sqrt{\omega \sigma \mu}$$

donde t es el espesor del material de blindaje.

La primera expresión indica que el blindaje para campo-E es más efectivo si el material de blindaje es altamente conductor, y menos efectivo si el material de blindaje es ferromagnético, y que los campos de menor frecuencia son más fáciles de bloquear que los de alta frecuencia. Esto se aprecia en la figura 8.5.

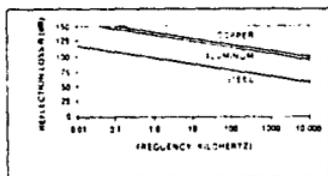


Fig.8.5 Blindaje contra Campo E

El cobre y el aluminio tienen la misma permeabilidad, pero el cobre es un poco más conductor y provee una pérdida de reflexión mayor. El acero es menos efectivo por su alta permeabilidad y menor conductividad.

Para el caso de la expresión para pérdida de absorción a un campo-H, el blindaje de campo-H es más efectivo en frecuencias mayores y con material de blindaje que tiene alta conductividad y alta permeabilidad. Sin embargo, en la práctica el seleccionar acero por su alta permeabilidad involucra algo de compromiso en su conductividad. Pero el incremento en la permeabilidad es tal que la baja en conductividad no influye mucho, como se aprecia en la figura 8.6. Esta figura también muestra el efecto del espesor del material de blindaje.

Un blindaje compuesto para el campo-E y el campo-H se muestra en la figura 8.7. Sin embargo, esta información tiene significado para el campo lejano. En el campo cercano la IEM podría ser un 90% de campo-H, en cuyo caso la pérdida por reflexión no es aplicable. Es recomendable irse por la pérdida de absorción, a pesar de la pérdida de reflexión, al escoger acero. Un mejor conductor que el acero puede resultar más económico, pero bastante ineficaz.

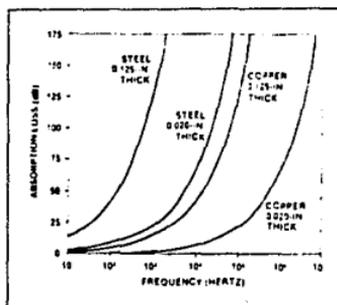


Fig. 8.6 Blindaje contra Campo H

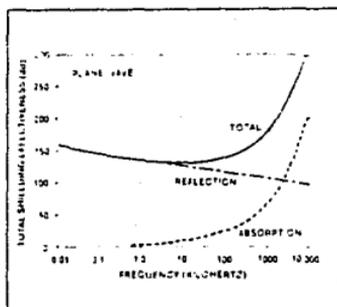


Fig. 8.7 Blindaje contra Campos E y H

En términos reales resulta menos importante la selección del material que la presencia de ranuras, uniones y agujeros en la estructura física del blindaje. Los mecanismos de blindaje se relacionan con la inducción de corrientes en el material de blindaje, pero a las corrientes se les debe permitir un flujo libre. Si tienen que desviarse por ranuras y agujeros, como se muestra en la figura 8.8, el blindaje pierde mucho de su efectividad.

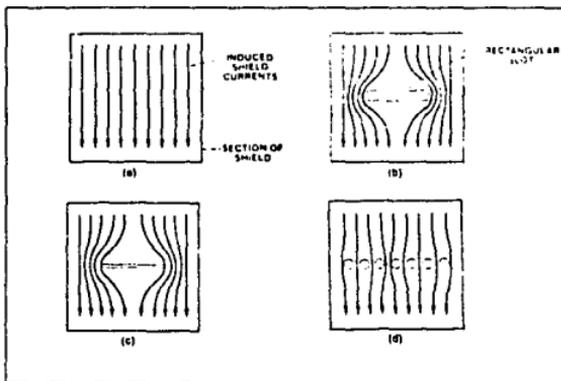


Fig. 8.8 Pérdida de Eficiencia en el Blindaje

Como se aprecia en la figura 8.8, la severidad de la desviación depende más de la geometría del agujero que de su área. Comparando la figura 8.8c con la 8d encontramos que una larga y delgada discontinuidad, tal como una ranura, provoca una mayor filtración de RF que una hilera de hoyos con un área total mayor.

8.5.3. Tierras.

Existen dos tipos de tierras: la tierra de seguridad y la tierra de señal.

8.5.3.1. Tierra de seguridad.

Generalmente se encuentra a potencial de tierra y los circuitos se conectan a ella para protección contra descargas peligrosas.

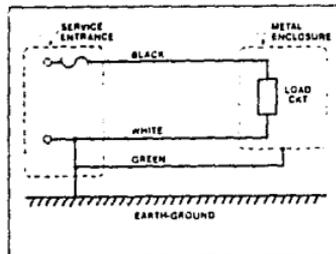


Fig.8.9 Distribución Monofásica de Potencia

El sistema estándar de distribución de CA de tres hilos-una fase, se representa en la figura 8.9. El cable blanco está conectado a la tierra de seguridad en la entrada de servicio. Si un circuito de carga se dispone en una caja o chasis metálicos, y si el cable negro forma un corto con dicha parte metálica, entonces habrá peligro de descarga para los operadores, si dicha parte no se encuentra aterrizada. Si la caja se encuentra aterrizada, un corto resulta en un fusible quemado. La ventaja del sistema de potencia de tres hilos es que distribuye una tierra de seguridad junto con la potencia.

8.5.3.2. Tierra de señal.

La tierra de señal es un nodo arbitrario de referencia de un circuito. Se define como un camino de baja impedancia por el cual la corriente retorna a la fuente. Los cables que se conectan a este punto también se denominan como tierra de señal. Dependiendo de como se conecten estos cables a dicho punto, se distinguen tres métodos de conexión: serie, paralelo y multipunto. Estos métodos se aprecian en la figura 8.10.

La conexión serie es bastante común por ser sencilla y económica. Sin embargo, es la más ruidosa, por el acoplamiento de impedancia de la tierra común. Los circuitos que

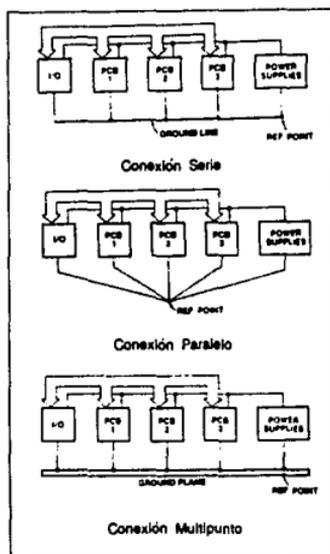


Fig.8.10 Tres Maneras de Alambrear las Tierras

8.5.3.3. Aterrizaje práctico.

Una combinación de los métodos serie y paralelo se puede emplear por razones económicas y consideraciones eléctricas. El concepto reside en emplear conexiones serie para circuitos con propiedades de ruido similares y conectarlos en paralelo a un sólo punto de referencia, como se muestra en la figura 8.11.

En la figura 8.11, la "señal de tierra ruidosa" se conecta a equipo como motores y relevadores. La tierra de hardware es la conexión de seguridad a chasis, repisas y gabinetes. Es un error el emplear la tierra de hardware como una ruta de retorno para corrientes de señal, ya que es algo ruidosa y tiende a tener alta resistencia debido a juntas y ranuras.

En la figura 8.12 se aprecia una aplicación del método serie/paralelo en el nivel de la TCI. Las corrientes en "displays" de LEDs multiplexados tienden a introducir mucho

producen altas corrientes afectan las etapas de bajo nivel. Y ya que en un sistema digital las corrientes presentan picos y la impedancia común es principalmente reactancia inductiva, las variaciones pueden ser lo suficientemente graves como para provocar errores en los bits en situaciones de alta corriente y particularmente ruidosas.

La conexión paralelo elimina los problemas de la impedancia en la tierra común, pero emplea mucho cable. Otras desventajas son que las impedancias de las líneas de tierra individuales pueden ser muy altas, y que las líneas de tierra pueden ser fuentes de IEM por sí solas.

En el sistema multipunto, la impedancia de tierra se minimiza usando un plano de tierra (generalmente el chasis del equipo) con los diversos circuitos conectados a él mediante líneas pequeñas. Este método se emplea por lo general en circuitos de RF arriba de los 10 MHz y en circuitaría digital para minimizar la impedancia de la tierra, debido a su baja inductancia.

ruido debido a las interrupciones constantes y los cambios involucrados en el proceso de examinación. La tierra del driver de segmentos esta relativamente quieta, ya que no conduce corrientes de los LEDs. La tierra del driver de dígitos está más ruidosa y necesita de una ruta separada a la terminal de tierra de la TCI, aún cuando la disposición de tierras de la TCI sea en malla. Las rutas de alimentación y retorno para los LEDs deben ser dispuestas en lados opuestos de la tarjeta como conductores paralelos.

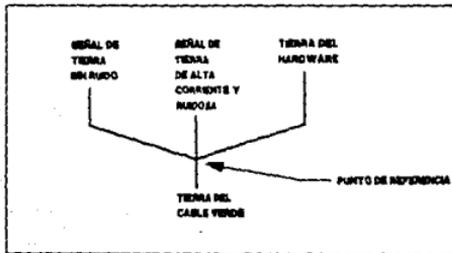


Fig.8.11 Conexión en Paralelo de Tierras en Serie

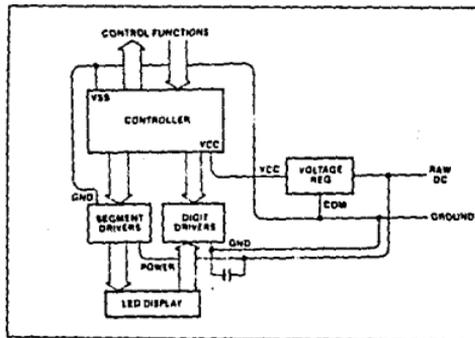


Fig.8.12 Tierra Separada para un Display Multiplexado

8.5.3.4. Cable trenzado.

Los problemas de impedancia en la línea de tierra se pueden eliminar con cable trenzado. La reducción en la impedancia se debe al efecto "skin": a mayores frecuencias la corriente tiende a fluir sobre la superficie del conductor en vez de uniformemente por todo su grosor. Mientras que este efecto tiende a incrementar la impedancia de un conductor dado, también indica la manera de minimizar la impedancia, y esto es manipulando la forma de la sección transversal para que provea mayor área.

8.5.4. Distribución y Desacople de la Fuente de Poder.

Al igual que las líneas de señalización, en las líneas de distribución se busca minimizar el área de los lazos de corriente. Sin embargo, es más difícil ya que estas líneas llegan a todas las TCIs del sistema. Y una anomalía en el suministro es una anomalía que llega a cada tarjeta del sistema.

La fuente de poder provee corriente de baja frecuencia a la carga, pero la inductancia en las distribuciones de entre tarjetas y entre integrados dificulta el mantener un VCC nominal y a la vez proveer transitorios de corriente requeridos por el sistema digital. Además, el lazo de corriente de la fuente de poder es muy grande y por tanto susceptible a ruido. La figura 8.13a muestra un circuito de carga intentando tomar picos de corriente de un voltaje de alimentación a través de la impedancia de la línea. A la forma de onda del VCC mostrado se le debe agregar el ruido del tipo inductivo asociado a un lazo de gran área.

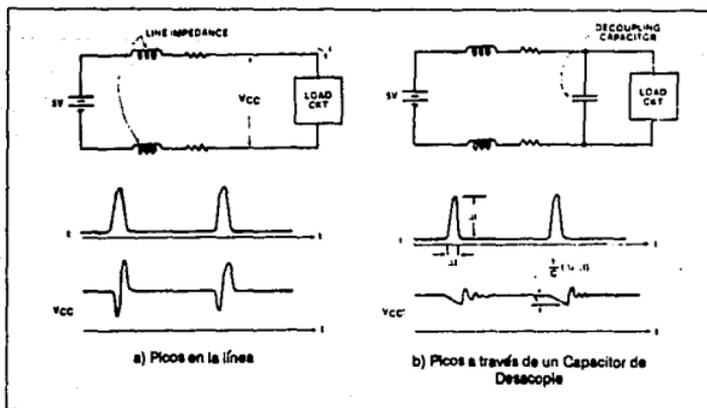


Fig.8.13 Función de un Capacitor de Desacople

Agregando un capacitor de desacople soluciona dos problemas: El capacitor actúa como una fuente cercana de carga para proveer los transitorios de corriente a través de una impedancia de línea más pequeña, y define un área de lazo más pequeña para los componentes de alta frecuencia de la IEM. Esto se ilustra en la figura 8.13b, la cual muestra el capacitor proveyendo el transitorio de corriente, y donde el VCC cae desde los 5V hasta la cantidad mostrada. El capacitor se recupera a través de la impedancia de línea entre los transitorios de corriente.

Es recomendable no agregar una resistencia o inductor al desacoplador, para formar un filtro paso-bajas RC o LC genuinos; ya que disminuye la velocidad de "refresco" del capacitor. El buen filtrado y el buen desacople no son necesariamente lo mismo.

Existen dos tipos de capacitores de desacople: los desacopladores de tarjeta y los desacopladores de integrado. Un desacoplador de tarjeta normalmente es un capacitor electrolítico de 10 a 100 μF , colocado en la proximidad de la entrada de la alimentación en la TCI; su colocación no es crítica. Su propósito es el de "refrescar" la carga de los desacopladores de los integrados. Los desacopladores de los integrados son los que realmente proveen los transitorios de corriente. Un desacoplador de integrado normalmente tiene un valor de 0.1 a 1 μF del tipo cerámico y conectado al integrado por líneas que minimicen el área del lazo formado por el capacitor y el integrado. Su colocación sí es crítica. Y la colocación correcta e incorrecta se muestra en la figura 8.14.

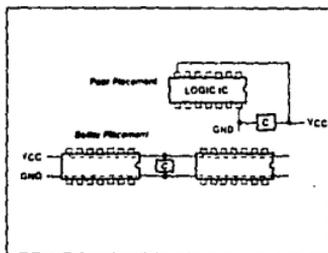


Fig.8.14 Colocación de los Capacitores de Desacople

La distribución de las líneas de potencia en la TCI debe ser tal que sea posible obtener un área mínima en los lazos formados por cada integrado y su desacoplador, y por los desacopladores de integrado y los desacopladores de tarjeta. Una manera de hacerlo es con un plano de potencia. Un plano de potencia es lo mismo que un plano de tierras pero con potencial de VCC. Algo más económico es una malla de potencia similar a la malla de tierras discutida anteriormente (figura 8.4). Sin embargo, si los lazos de desacople de integrados son pequeños, no se necesita de planos o mallas de potencia; las líneas de potencia se deben disponer lo más próximo posible a las líneas de tierra.

8.5.4.1. Como seleccionar el valor del capacitor de desacople.

Por lo general se dice, y no muy acertadamente, que el desacoplador del integrado no debe tener un valor muy grande. Existen dos razones para esta declaración. Una es

que algunos capacitores, por la naturaleza de sus dieléctricos, tienden a ser inductivos en altas frecuencias. Esto es cierto para los capacitores electrolíticos, pero los dieléctricos de mica, vidrio, cerámica y poliestireno funcionan bastante bien en los cientos de MHz. La otra razón tiene que ver con la inductancia de las terminales.

El capacitor con su inductancia de terminal forma un circuito LC serie. Debajo de la frecuencia de resonancia en serie, la impedancia neta del arreglo es capacitiva. Arriba de esa frecuencia, la impedancia neta es inductiva. Por lo que un capacitor de desacople es solamente capacitivo debajo de la frecuencia de resonancia en serie. La frecuencia se da por :

$$f_0 = \frac{1}{2\pi\sqrt{LC}}$$

donde C es la capacitancia de desacople y L es la inductancia de terminal entre el capacitor y el integrado.

En una TCI esta inductancia se determina por la distribución y es independiente del capacitor. Por ello, el incrementar la capacitancia disminuye la frecuencia de resonancia serie. Por la fórmula anterior notamos que si se incrementa C por un factor de 100, esto hace que la frecuencia de resonancia se disminuya por un factor de 10.

8.5.4.2. El caso para regulación de voltaje en la misma tarjeta.

Las anomalías en la alimentación no siempre provienen de las redes de alimentación, pero pueden provenir del circuito de la fuente de poder. En tal caso, una red de distribución bien diseñada llevará fielmente la anomalía a todo el sistema. Y un desacople normal no será suficiente para solventar la falla, por lo que en algunos casos es necesario agregar regulación de voltaje en la misma tarjeta.

El hacer esto permite facilitar los requerimientos de disipación de calor en la fuente de poder, y el desacople en la distribución y tarjeta. Sin embargo, también trae la posibilidad de que diferentes tarjetas estén operando con niveles de VCC diferentes debido a la tolerancia en los integrados de regulación; esto conduce a niveles lógicos diferentes entre las tarjetas.

8.5.5. Recuperándose Adecuadamente de una Falla del Software.

Aún cuando se tomen todas las medidas del diseño en ambientes ruidosos, siempre es posible que un transitorio de ruido ocurra excediendo el nivel de inmunidad del circuito. En tal caso, uno debe procurar una recuperación adecuada.

Los esquemas de recuperación adecuada involucran hardware y/o software adicional, que debe regresar al sistema a un modo normal de operación después de ocurrir una falla del software. Es necesario realizar dos decisiones: Como reconocer cuando ha ocurrido una falla y que hacer al respecto.

Si el diseñador sabe que tipos y que combinaciones de salidas se pueden generar legalmente por el sistema, puede utilizar compuertas para reconocer y avisar cuando ha ocurrido una anomalía. Al percatar el aviso se puede realizar un brinco a una rutina de recuperación, la cual puede revisar o inicializar los datos, tal vez sacar un mensaje de error, o generar una simple inicialización.

El esquema más confiable consiste en utilizar el llamado circuito "watchdog". En este el CPU se programa para generar una señal periódica mientras que el sistema ejecuta instrucciones de un modo esperado. La señal periódica se emplea, entonces, para detener un circuito que disparará un salto a una rutina de recuperación. La señal periódica debe de acoplarse en CA al circuito disparador para que una falla "pegada" no continúe deteniendo el disparo. Así, si el procesador se paraliza en un estado, la señal periódica se pierde y el "watchdog" dispara una inicialización.

En la práctica, puede resultar conveniente manejar el circuito de "watchdog" con una señal que de cualquier manera se genera por el sistema. Sin embargo, uno debe de tener precaución y cerciorarse que una falla interrumpa dicha señal. Por ejemplo, y en específico, uno puede emplear una de las señales de manejo de un display multiplexado. Pero la examinación de un display se maneja generalmente en respuesta a una interrupción por temporizador, la cual puede continuar operando aún cuando el programa principal esta en modo de falla. Aún así, con un poco de software extra, la señal se puede emplear para controlar el "watchdog".

Esquemas más simples pueden funcionar para circuitos más simples. Por ejemplo, si un CPU no hace mas que examinar y decodificar un teclado, no se pierde mucho si se inicializa periódicamente con un multivibrador astable.

Una medida gratis consiste en llenar toda la memoria sin usar con NOPs y JMPs a una rutina de recuperación. La efectividad de este método se incrementa al escribir el programa en segmentos separados por NOPs y JMPs. Con este método se obtiene gran protección por un ínfimo costo.

CONCLUSIONES

CONCLUSIONES

Para poder conocer las necesidades reales de automatización de la Industria Mexicana, fué necesario realizar visitas a diferentes empresas. De dicha investigación pudimos observar que la mayoría de los procesos involucraban variables a controlar muy semejantes, de igual forma, se pudo inferir que la diferencia principal entre las mismas radica en los rangos de operación usados, los cuales varían dependiendo de los requerimientos de las líneas de producción de cada Industria.

En base a lo anterior, encontramos ciertas etapas fácilmente identificables en cualquier proceso industrial (almacenamiento, procesamiento, transporte, etc.). Así emprendimos el desarrollo de un *Sistema de Control* de ciertos parámetros o variables que intervienen en los distintos procesos industriales, de tal forma que dicho sistema presentase dos conceptos innovadores en relación a lo que se ha planteado en trabajos anteriores: El primero de ellos está relacionado con la forma en la cual se lleva a cabo la transferencia de información entre el usuario del sistema (a través de una computadora personal compatible con IBM) y las diferentes etapas que constituyen un proceso industrial determinado (parámetros o variables a controlar), ya que como se mencionó en el presente trabajo, la comunicación se realiza en forma serial (debido a la distancia existente entre los diferentes módulos a conectar) a través de un cierto protocolo de comunicación (SDLC), el cual nos permite establecer una comunicación relativamente rápida y en forma simultánea entre el usuario y las diferentes etapas del proceso. Además que este mismo protocolo nos permite estar monitoreando un mínimo de 256 unidades (parámetros o variables a controlar), de tal forma que se tiene el control de la mayoría de ellos, sino es que en su totalidad.

Por otra parte, el segundo concepto innovador planteado en el Sistema está relacionado con la definición del tipo de Controlador a usar en el esquema de control planteado. Como se recordará, el controlador en un sistema de control típico interpreta los cambios que puedan surgir en las características de la variable a controlar (la cual es suministrada por un cierto dispositivo de medición), además de que a través de este mismo dispositivo se realiza la acción controladora que es necesaria para mantener el balance deseado dentro del proceso. De tal forma, que en nuestro proyecto se plantea la idea de utilizar un Microcontrolador (8031 de Intel) para dichas acciones.

Se decidió utilizar un Microcontrolador debido a que presenta grandes ventajas, en comparación de ciertos Microprocesadores (usualmente utilizados como dispositivos de Control). En primer lugar, con los Microcontroladores se tiene una ROM integrada lo cual nos lleva a tener cierta reducción en cuanto al hardware asociado al Sistema.

Por otra parte, se pueden tener una gran rapidez de procesamiento (cerca de 1 millón de instrucciones por segundo con un reloj de 12 MHz). Finalmente, presenta una

gran versatilidad en cuanto al desempeño de sus funciones ("set" de instrucciones muy amplio), lo cual nos permite desarrollar e implantar acciones sumamente poderosas.

A fin de desarrollar un proyecto de esta naturaleza, se hizo indispensable el apoyo de ciertos conceptos relativos a la industrias mismas, a el control, equipo de cómputo, herramientas de desarrollo de software, elementos de hardware asociados a la arquitectura de una computadora personal, así como a técnicas de diseño de software y hardware específicas.

Cabe hacer mencionar, que una limitante del proyecto propuesto en el presente trabajo, es la etapa de acondicionamiento de señales, debido a que no se hace referencia a un estudio más profundo de esto, ya que lo que se espera es que en el uso real, la Industria contemple la definición y el manejo de las señales provenientes de los elementos de medición (esquema de control típico), de tal forma que se adecuen a las características y necesidades propias de las mismas; para que con la información que entregue la planta se realice la selección mas adecuada del dispositivo de medición. En contraposición a lo anteriormente expuesto, se puede decir que el objetivo primordial del presente trabajo estuvo enfocado al estudio relacionado con la interacción que surge entre el usuario del Sistema y las diferentes etapas, susceptibles a ser controladas, que componen un proceso industrial determinado.

Consideramos que el presente trabajo, cumple con los objetivos planteados al inicio del mismo, ya que mediante él, se puede llegar a tener un Sistema de Control sencillo y fácil de manejar, además de que puede ser adaptado a las características y necesidades particulares de cada una de las Industrias que lo requieran, es decir, que lo que se pretende es que no sea un trabajo aplicado a una cierta Industria en particular, sino que por el contrario, sea lo más universal posible y pueda ser usado por cualquier tipo de Industria.

Por otra parte, se trató de tener costos de producción muy bajos, por lo cual, se hizo una investigación exhaustiva de los dispositivos que pudieran manejar funciones muy poderosas al mínimo costo y así reducir la cantidad de hardware asociado al Sistema. Sin embargo, todo esto trajo consigo la elaboración de un software muy sofisticado.

En el momento en que iniciamos el diseño del sistema, utilizamos los componentes que en ese entonces representaban la tecnología de punta; sin embargo, debido al vertiginoso avance de la electrónica, los dispositivos propuestos, en algunos años pasarán a ser obsoletos.

BIBLIOGRAFIA

BIBLIOGRAFIA

Boylestad, N., *Electrónica Teoría de Circuitos*, Prentice Hall, 1983.

Chiang, J., *A/D Converters Easily Interface with 70 Series Microprocessors*, National Semiconductor Application Note 280.

Ciarcia, S., *Why Microcontrollers? (I y II)*, BYTE Magazine, Agosto y Septiembre de 1988.

Considine, D. M., *Process Instruments and Control Handbook*, McGraw-Hill.

Dixon, R., *Data Communications Facilities, Networks and Systems Design*, Wiley Interscience, New York, 1984.

Dogan A., *Data Transmission*, McGraw Hill, New York, 1986

Hagerty, D., *Data Acquisition Using INS8048*, National Semiconductor Application Note 281.

IBM[®] *Technical Reference Guide*, IBM Corporation, 1987.

Katausky, J., Horden, I., Smith, L., *Analog/Digital Processing with Microcontrollers*, Intel Article Reprint AR-526.

Kaufman, M., Seidman, A., *Manual para Ingenieros y Técnicos en Electrónica*, McGraw-Hill, 1982.

Manuales de INTEL

8-Bit Embedded Controllers, 1990.

Embedded Applications, 1990.

Peripherals, 1990.

Microcommunications, Vols. 1 y 2, 1990.

Manuales de National Semiconductor

Linear Databook 2, 1988

Linear Applications Handbook, 1986

Manuales de Texas Instruments

The TTL Data Book, Vol. 2, 1985.

Linear and Interface Circuits Applications, Vols. 1, 2 y 3, 1987.

Manual de Mantenimiento de PCs XT/AT y Periféricos, Facultad de Ingeniería UNAM. División de Educación Continua, 1991.

Microsoft Mouse. Programmer's Reference, Microsoft Press, 1989.

Nagel, H., Temperature Compensation, Calibration and Applications of Motorola's X-ducer Pressure Sensor, Motorola Application Note AN-922, 1987

Norton, P., Wilton, R., The New Peter Norton Programmer's Guide to The IBM[®] PC & PS/2[®], Microsoft Press, 1988.

Ogata, K., Ingeniería de Control Moderna, Prentice Hall, 1986.

Ott, H., Noise Reduction Techniques in Electronic Systems, New York, Wiley, 1988.

Power Tools, User Reference Manual, Blaise Computing, Inc, 1985.

Ryan, D., MCS-96 Analog Acquisition Primer, Intel Application Note AP-406, Diciembre 1987.

Schuler, C., McNamee, W., Industrial Electronics and Robotics, McGraw-Hill, 1988.

Wharton, J., An Introduction to the Intel MCS-51 Single-Chip Microcomputer Family, Intel Application Note AP-69, Mayo 1980.

Wharton, J., Using the Intel MCS-51 Boolean Processing Capabilities, Intel Application Note AP-70, Abril 1980.

Williamson, T., *Designing Microcontroller Systems for Electrically Noisy Environments*, Intel Application Note AP-125, Febrero 1982.

Williamson, T., *Designing With The 80C51BH*, Intel Application Note AP-252, Septiembre 1987.

Williamson, T., *Using the 8051 Microcontroller with Resonant Transducers*, IEEE Transactions on Industrial Electronics, Vol. IE-32, No. 4, Noviembre 1985.

APENDICES

APENDICE A

LA FAMILIA MCS-51[®] DE INTEL

INTRODUCCION

En 1976 Intel introdujo la familia MCS-48, formada por las microcomputadoras 8048, 8748 y 8035. Estos dispositivos marcaron el inicio de un sistema de microcomputadora completo en una sola pastilla. Formado por un CPU de 8 bits, 1K interno de ROM o EPROM para memoria del programa, 64 palabras de memoria de datos, puertos de E/S y un temporizador/contador. Dependiendo sólomente del contenido de la memoria del programa, un sólo integrado podía controlar una gran variedad de productos, desde aparatos domésticos o motores de automóviles hasta equipo de procesamiento de textos o datos. Productos subsecuentes encaminaron la arquitectura del MCS-48 a otras nuevas direcciones: el 8049 y el 8039 aumentaron la memoria interna y corrían un 83% más rápido; el 8021 redujo los costos al ejecutar un abstracto de las instrucciones del 8048 con un reloj mas lento; y el 8022 incluyó un convertidor A/D de 8 bits y dos canales, muy original, en la misma pastilla NMOS de la computadora, permitiendo una interface directa con transductores analógicos.

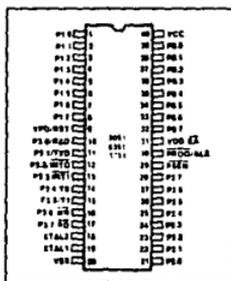


Fig.A.1 Diagrama de Terminales del 8051

Hoy día se cuenta con nuevas microcomputadoras en una sola pastilla, el 8051, 8751 y 8031, todos de Intel. Los cuales extienden las ventajas de la electrónica integrada a una gran variedad de nuevos productos. Gracias a la nueva tecnología HMOS de Intel, la familia MCS-51 proporciona hasta 4 veces la memoria de programa y el doble de la memoria de datos que un 8048. Nuevas facultades de periféricos y E/S incrementan el rango de aplicación y reducen el costo total de un sistema. Dependiendo de su uso, la capacidad de procesamiento se incrementa de un 2.5 a 10 veces.

El 8051 es el miembro original de la familia MCS-51, y es la esencia de todos los dispositivos MCS-51. Su distribución de terminales se muestra en la figura A.1.

Las características básicas del 8051 son:

- Alimentación de 5V única, empleando tecnología HMOS
- CPU de 8 bits optimizado para aplicaciones de control
- Capacidades para extenso procesamiento booleano
- Direccionamiento de hasta 64K en la memoria de programa
- Direccionamiento de hasta 64K en la memoria de datos
- 4K bytes de memoria de programa interna (ROM)
- 128 bytes internos de RAM
- Cuatro bancos de registros
- 128 banderas de software definibles por el usuario
- Ciclo de instrucción de 1 μ s empleando oscilador de 12 MHz
- 32 líneas de E/S bidireccionales y direccionables independientemente
- Dos temporizador/contador de 16 bits
- Puerto serial de alta velocidad, con varios modos de programación
- Estructura de interrupción de 6-fuentes/5-vectores con dos niveles de prioridad
- Stack completo para ligado y almacenamiento de datos al regreso de subrutinas
- Direccionamiento de un sólo byte o bit
- Aritmética binaria o decimal
- Multiplicación y división por hardware en 4 μ s
- Reloj oscilador en la pastilla

La arquitectura básica de los MCS-51 se muestra en la figura A.2.

Cada dispositivo de la familia MCS-51 tiene la misma arquitectura básica pero algunos cuentan con facultades adicionales. Una comparación de los dispositivos de la familia MCS-51 se muestra en la tabla A.1.

Tabla A.1 Familia de Microcontroladores MCS-51

Device	ROMless Version	EPROM Version	ROM Bytes	RAM Bytes	8-Bit I/O Ports	16-Bit Timer/Counters	Programmable Counter Array (PCA)	UART	Serial Expansion Port (SEF)	Global Serial Channel (GSC)	DMA Channels	A/D Channels	Interrupt Sources/Vectors	Power Down and Idle Modes
8051	8051	—	4K	128	4	2		✓					6/5	
8051AH	8031AH	8751H 8751BH	4K	128	4	2		✓					6/5	
8052AH	8032AH	8752BH	8K	256	4	3		✓					8/6	
80C51BH	80C31BH	87C51	4K	128	4	2		✓					6/5	✓
80C52	80C32	—	8K	256	4	3		✓					8/6	✓
83C51FA	80C51FA	87C51FA	8K	256	4	3	✓	✓					14/7	✓
83C51FB	80C51FA	87C51FB	16K	256	4	3	✓	✓					14/7	✓
83C152JA	80C152JA	—	8K	256	5	2		✓		✓	2		19/11	✓
—	80C152JB	—	—	256	7	2		✓		✓	2		19/11	✓
83C152JC	80C152JC	—	8K	256	5	2		✓		✓	2		19/11	✓
—	80C152JD	—	—	256	7	2		✓		✓	2		19/11	✓
83C452	80C452	87C452P	8K	256	5	2		✓					9/6	✓

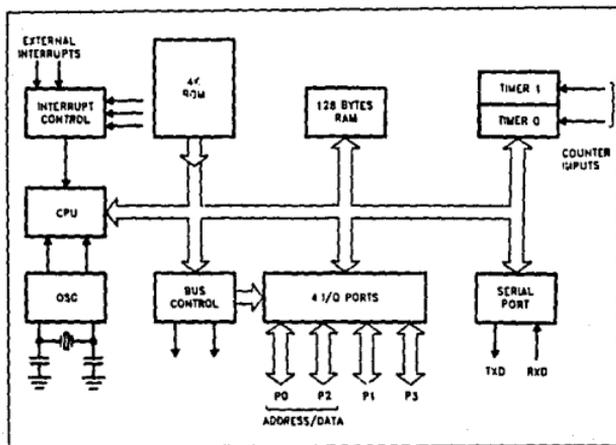


Fig.A.2 Diagrama de Bloques del 8051

ARQUITECTURA Y ORGANIZACION

La figura A.3 muestra el diagrama de bloques de la organización interna de la familia MCS-51. Cada microcomputadora combina un CPU, dos tipos de memoria (RAM para datos y ROM o EPROM para programa), puertos de E/S, registros de modo, estatus y datos, y lógica adicional empleada para una variedad de aplicaciones periféricas. Estos elementos se comunican a través de un bus de datos de bits que corre por todo el integrado. Cuando se desea expansión de memoria o de E/S, este bus se comunica al exterior a través de un puerto de E/S.

Unidad Central de Proceso.

Sus elementos primarios los forman un ALU de 8 bits y diversos registros A, B, PSW y SP, y los dos registros de 16 bits PC y DPTR. Todos estos registros se mapean en la memoria interna del integrado y se les denomina registros de función especial (SFR). En la figura A.4 se muestra el mapa de los SFR. Las funciones de los SFR se enlistan a continuación.

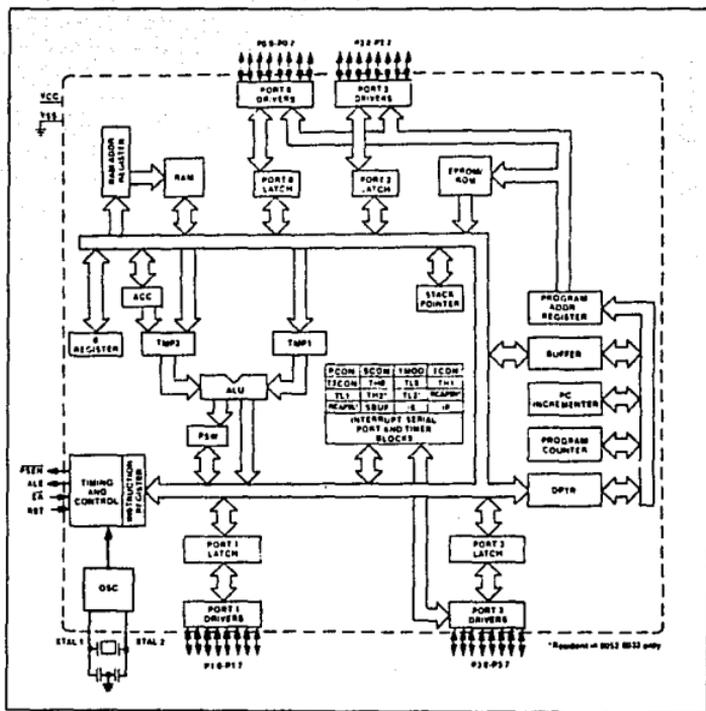


Fig.A.3 Arquitectura Interna de la Familia MCS-51

8 Bytes							
F6							FF
F0	B						F7
E8							EF
E0	ACC						E7
D6							D7
D0	PSW						D7
C6	(I2CCN1)	(RCAP2L)	(RCAP2H)	(TL2)	(TH2)		CF
C0							C7
B6	IP						B7
B0	P3						B7
A6	IE						AF
A0	P2						A7
96	SCCN	SB ₀₋₇					9F
90	P1						97
86	TCN	TMCD	TL0	TL1	TH0	TH1	8F
80	P0	SP	DPL	DPH		PCON	87

Fig.A.4 Mapa de Registros de Función Especial

a) Acumulador.

ACC es el registro acumulador. Guarda el operando fuente y recibe el resultado de las instrucciones aritméticas. El acumulador puede ser la fuente o el destino de operaciones lógicas y para un número de instrucciones especiales para el movimiento de datos, incluyendo búsqueda en tablas y expansión externa de RAM. Muchas funciones se aplican exclusivamente sobre el acumulador: rotaciones, cálculo de paridad, probar por cero, entre otras.

b) Registro B.

El registro B se emplea en operaciones de multiplicación y división. Para otras instrucciones se puede tratar como un registro de propósito general.

c) Program Status Word (PSW).

Muchas instrucciones implícita o explícitamente afectan varias banderas de estado, las cuales se agrupan entre sí para formar el PSW, como se muestra en la figura A.5.

d) Apuntador de Stack.

El registro SP tiene una longitud de 8 bits. Se incrementa antes de que se guarden datos con las instrucciones PUSH y CALL. Mientras que el stack puede residir en cualquier lado de la RAM interna, el SP se inicializa con 07H después de un reset. Esto hace que el stack empiece en la localidad 08H.

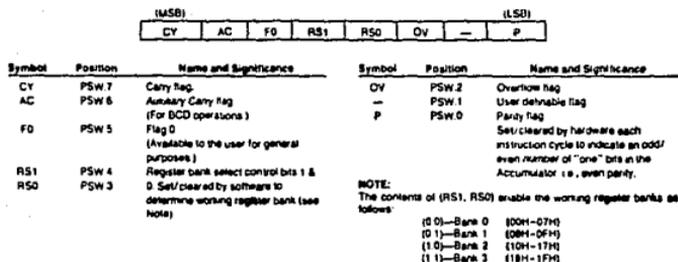


Fig.A.5 Registro PSW

e) Apuntador de datos.

El registro DPTR está formado de un byte alto (DPH) y un byte bajo (DPL). Su función intencional es la de guardar una dirección de 16 bits. Se puede manejar como un registro de 16 bits o como dos registros de 8 bits independientes.

f) Puertos 0 a 3.

P0, P1, P2 y P3 son los latches de los SFR para los puertos 0, 1, 2 y 3, respectivamente.

g) El "buffer" serial de datos.

El buffer serial de datos (SBUF) es en realidad dos registros separados, un registro de "buffer" de transmisión y un registro de "buffer" de recepción. Cuando se mueven datos al SBUF, se van al "buffer" de transmisión donde se mantienen para transmisión serial. Cuando los datos se mueven del SBUF, provienen del "buffer" de recepción.

h) Los registros temporizadores.

Los pares de registros (TH0, TL0), (TH1, TL1) y (TH2, TL2) son los registros contadores de 16 bits para los temporizador/contador 0, 1 y 2, respectivamente.

i) Registros de control.

Los SFRs IP, IE, TMOD, TCON, T2CON, SCON y PCON contienen bits de control y de estado para el sistema de interrupciones, los temporizador/contadores y el puerto serial.

Unidad lógica aritmética.

El ALU puede realizar funciones lógicas y aritméticas sobre variables de 8 bits. También puede realizar decisiones para saltos condicionales, y provee ruteo de datos y registros temporales para la transferencia de datos en el sistema. Otras instrucciones se forman de estas funciones primitivas.

Una característica importante y propia de la arquitectura MCS-51 es que el ALU puede manejar los tipos de datos de un bit o un byte. Bits individuales se pueden fijar, borrar, o complementar, mover, probar y emplear en cálculos lógicos. Esto le permite al 8051 establecerse muy bien en las aplicaciones de control. Tales algoritmos involucran inherentemente variables Booleanas de entrada y salida, los cuales son difíciles de implementar con microprocesadores convencionales.

Gracias a este ALU tan poderoso, el set de instrucciones del 8051 es adecuado para algoritmos de control en tiempo real e intensivos en datos. Un total de 51 operaciones separadas mueven y manipulan datos de tres tipos: booleano (1 bit), byte (8 bits) y dirección (16 bits). Con 11 modos de direccionamiento, 7 para datos y 4 para control de la secuencia de programa. La mayoría de las operaciones permiten varios modos de direccionamiento, elevando el número total de instrucciones a 111.

Set de instrucciones.

La tabla A.2 enlista estas 111 instrucciones clasificándolas en cinco grupos:

- Operaciones aritméticas
- Operaciones lógicas para variables tipo byte
- Instrucciones de transferencia de datos
- Manejo de variables booleanas
- Control de máquina y ruteo

El set de instrucciones está diseñado para hacer programas eficientes, tanto en términos de tamaño del código como velocidad de ejecución. Ninguna instrucción requiere de más de tres bytes de la memoria de programa, la mayoría requiriendo de 1 a 2 bytes. Virtualmente todas las instrucciones se ejecutan en uno o dos ciclos de instrucción (uno o dos microsegundos con un cristal de 12 MHz).

Organización de la Memoria en los Dispositivos MCS-51

Todos los dispositivos MCS-51 tienen espacios separados para las direcciones de la memoria de programa y la de datos, como se muestra en la figura A.6 (configuración no von Neumann). La separación lógica entre la memoria de programa y la de datos

permite que la memoria de datos se accese por direcciones de 8 bits, las cuales pueden ser guardadas y manipuladas más rápido con un CPU de 8 bits. Sin embargo, direcciones de datos de 16 bits pueden ser generadas a través del registro DPTR.

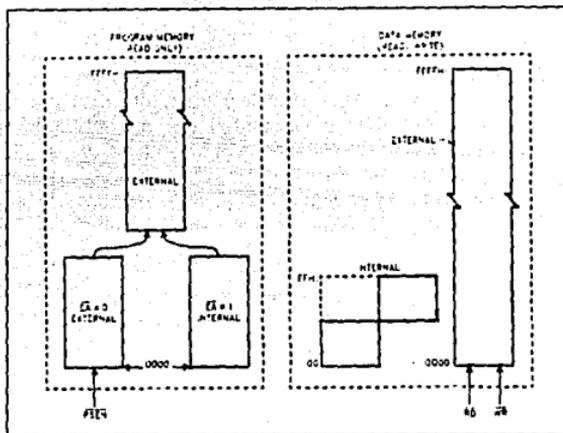


Fig.A.6 Estructura de la Memoria en los MCS-51

La memoria de programa sólo puede ser leída. Puede haber hasta 64K bytes de memoria de programa. En las versiones de ROM o EPROM los 4K, 8K o 16K iniciales se proveen en el mismo integrado. Referirse a la tabla A. 1 para saber la cantidad de ROM (o EPROM) de cada dispositivo. En las versiones sin ROM toda la memoria de programa es externa. El destellador (strobe) para la lectura de la memoria externa se dá con la señal PSEN' (Program Store Enable).

La memoria de datos ocupa un espacio separado de la memoria de programa. Se pueden accesar hasta 64K bytes de RAM externa. El CPU genera las señales de lectura y escritura, RD' y WR', según sea necesario en los accesos a memoria de datos externa.

Se pueden combinar ambas memorias externas aplicando las señales de RD' y PSEN' a las entradas de una compuerta AND y empleando la salida de la compuerta como el strobe a la memoria programa/datos externa.

Puertos de Entrada/Salida.

La estructura de puertos E/S de la familia MCS-51 es extremadamente versátil. El 8051 y el 8751 tienen 32 terminales de E/S, configuradas como cuatro puertos paralelos (P0, P1, P2 y P3). Cada terminal meterá o sacará datos bajo control del software, y cada uno puede ser referido por medio de un extenso repertorio de operaciones con byte o bit.

En varios modos de operación o expansión, algunas de estas terminales de E/S también se emplean para funciones especiales de entrada o salida. Las instrucciones que accesan memoria externa utilizan el puerto 0 como un bus multiplexado de dirección/datos. Las instrucciones de transferencia externa de datos que proveen una dirección de 16 bits y cualquier instrucción que accese la memoria externa de programa, sacan los 8 bits más significativos por el puerto 2. (El 8031 siempre usa las terminales de los puertos P0 y P2 para direccionamiento externo, pero P1 y P3 quedan libres para E/S estándar.)

Las ocho terminales del puerto 3 tienen una función especial. Dos interrupciones externas, dos entradas de contador, dos líneas de datos seriales y dos destelladores para el control de tiempo hacen uso de las terminales del puerto 3. Las terminales del puerto 3 que no se empleen están a disposición para E/S convencional.

Aún con un sólo puerto, las funciones de E/S se pueden combinar de muchas maneras: entrada y salida se puede realizar usando diferentes terminales al mismo tiempo, o las mismas terminales en tiempos distintos; en paralelo en algunos casos y en serie para otros casos; como terminales de prueba, o (como es el caso del puerto 3) como funciones especiales adicionales.

Funciones Especiales de Periféricos.

Existen algunas necesidades comunes en los sistemas computarizados orientados al control:

- Llevar la cuenta del tiempo real elapsado
- Mantener una cuenta de las transiciones en las señales
- Medir el ancho preciso de los pulsos de entrada
- Comunicarse con otros sistemas o gente
- Monitoreo cercano de eventos externos asíncronos

Hasta ahora, los sistemas basados en microprocesador necesitaban dispositivos periféricos como temporizador/contadores, USARTs o controladores de interrupción para dar cabida a estas necesidades.

Temporizador/contadores.

Existen dos temporizador/contador de 16 bits en el 8051, cada uno formado por un byte alto y un byte bajo. Estos registros se denominan TH0, TL0, TH1 y TL1. Cada par puede ser programado independientemente por software, de varias formas con el registro de modo TMOD y controlados según el registro TCON.

Los modos de temporizador se pueden emplear para medir intervalos de tiempo, determinar anchos de pulso, o iniciar algún evento, con una resolución de un microsegundo, hasta un intervalo máximo de 65,536 ciclos de instrucción (más de 65 ms). Retardos más largos se pueden acumular por software. Configurado como contador, el mismo hardware acumulará eventos externos con frecuencias de CD hasta 500 kHz, y con una precisión de hasta 16 bits.

Interface de puerto serial.

Cada microcomputadora contiene un puerto serial full-dúplex de alta velocidad, el cual es programable por software en cuatro modos básicos de operación: registro de corrimiento para expandir E/S, UART de 8 bits, UART de 9 bits o eslabón para comunicación entre procesadores. Los modos de UART se interfazan directamente con dispositivos de E/S estándar, con tasas de datos de 122 baud hasta 31 kbaud. Remplazando el cristal de 12 MHz con uno de 10.7 MHz permite 110 baud. También se pueden incluir paridad par o impar con simples rutinas de manejo de bits en el software. La comunicación entre procesadores en procesos distribuidos se da en 187 kbaud con hardware para reconocimiento automático de mensajes de dirección/datos. Simples registros de corrimiento TTL o CMOS proveen una expansión de E/S bastante rápida, hasta 1 Megabaud. Los modos de operación del puerto serial se controlan con el contenido del registro SCON.

Manejo y control de interrupciones.

El 8051 tiene cinco fuentes de interrupción: una del puerto serial cuando se completa una transmisión o recepción, dos de los temporizadores al ocurrir un overflow y dos de las terminales de entrada (INT0 e INT1). Cada fuente se puede habilitar o deshabilitar independientemente para permitir un poleo en algunas fuentes o en algunos instantes, y cada una se puede clasificar con alta o baja prioridad. Estas opciones se seleccionan mediante los registros de habilitación de interrupciones y de control de prioridad, IE e IP. Otros dispositivos MCS-51 tienen fuentes y vectores adicionales de interrupción, como se muestra en la Tabla A.1.

LA FAMILIA MCS-81 DE INTEL

Fuente de Interrupcion	Rutinas de Servicio Direccion inicial
(Reset)	0000H
External 0	0003H
Timer/Counter 0	000BH
External 1	0013H
Timer/Counter 1	001BH
Serial Port	0023H

Cada fuente tiene una dirección particular de la memoria de programa (tabla A.3), comenzando en la 0003H y continuando por intervalos de ocho bytes. Cuando se va a atender una interrupción el CPU automáticamente salta a la dirección correspondiente. Una subrutina del usuario que comience en tal dirección (o que realice un salto a partir de ésta) realizará la atención a dicha interrupción. Terminada la atención la ejecución retorna a su estado anterior.



MCS[®]-51 INSTRUCTION SET

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings⁽¹⁾

Instruction	Flag			Instruction	Flag		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	O	X		ANL C,/bit	X		
DIV	O	X		ORL C,bit	X		
DA	X			ORL C,bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

⁽¹⁾Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings

Note on instruction set and addressing modes:

- Rn — Register R7–R0 of the currently selected Register Bank.
- direct — 8-bit internal data location's address. This could be an Internal Data RAM location (0–127) or a SFR (i.e., I/O port, control register, status register, etc. (128–255)).
- @Ri — 8-bit internal data RAM location (0–255) addressed indirectly through register Ri or R0.
- #data — 8-bit constant included in instruction.
- #data 16 — 16-bit constant included in instruction.
- addr 16 — 16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.
- addr 11 — 11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.
- rel — Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
- bit — Direct Addressed bit in Internal Data RAM or Special Function Register.

Mnemonic	Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS			
ADD A,Rn	Add register to Accumulator	1	12
ADD A,direct	Add direct byte to Accumulator	2	12
ADD A,@Ri	Add indirect RAM to Accumulator	1	12
ADD A,#data	Add immediate data to Accumulator	2	12
ADDC A,Rn	Add register to Accumulator with Carry	1	12
ADDC A,direct	Add direct byte to Accumulator with Carry	2	12
ADDC A,@Ri	Add indirect RAM to Accumulator with Carry	1	12
ADDC A,#data	Add immediate data to Acc with Carry	2	12
SUBB A,Rn	Subtract Register from Acc with borrow	1	12
SUBB A,direct	Subtract direct byte from Acc with borrow	2	12
SUBB A,@Ri	Subtract indirect RAM from ACC with borrow	1	12
SUBB A,#data	Subtract immediate data from Acc with borrow	2	12
INC A	Increment Accumulator	1	12
INC Rn	Increment register	1	12
INC direct	Increment direct byte	2	12
INC @Ri	Increment direct RAM	1	12
DEC A	Decrement Accumulator	1	12
DEC Rn	Decrement Register	1	12
DEC direct	Decrement direct byte	2	12
DEC @Ri	Decrement indirect RAM	1	12

LA FAMILIA MCS-81 DE INTEL

Mnemonic	Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS (Continued)			
INC DPTR	Increment Data Pointer	1	24
MUL AB	Multiply A & B	1	48
DIV AB	Divide A by B	1	48
DA A	Decimal Adjust Accumulator	1	12
LOGICAL OPERATIONS			
ANL A,Rn	AND Register to Accumulator	1	12
ANL A,direct	AND direct byte to Accumulator	2	12
ANL A,@Ri	AND indirect RAM to Accumulator	1	12
ANL A,#data	AND immediate data to Accumulator	2	12
ANL direct,A	AND Accumulator to direct byte	2	12
ANL direct,#data	AND immediate data to direct byte	3	24
ORL A,Rn	OR Register to Accumulator	1	12
ORL A,direct	OR direct byte to Accumulator	2	12
ORL A,@Ri	OR Indirect RAM to Accumulator	1	12
ORL A,#data	OR immediate data to Accumulator	2	12
ORL direct,A	OR Accumulator to direct byte	2	12
ORL direct,#data	OR immediate data to direct byte	3	24
XRL A,Rn	Exclusive-OR Register to Accumulator	1	12
XRL A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR A	Clear Accumulator	1	12
CPL A	Complement Accumulator	1	12

Mnemonic	Description	Byte	Oscillator Period
LOGICAL OPERATIONS (Continued)			
RL A	Rotate Accumulator Left	1	12
RLC A	Rotate Accumulator Left through the Carry	1	12
RR A	Rotate Accumulator Right	1	12
RRC A	Rotate Accumulator Right through the Carry	1	12
SWAP A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER			
MOV A,Rn	Move Register to Accumulator	1	12
MOV A,direct	Move direct byte to Accumulator	2	12
MOV A,@Ri	Move indirect RAM to Accumulator	1	12
MOV A,#data	Move immediate data to Accumulator	2	12
MOV Rn,A	Move Accumulator to Register	1	12
MOV Rn,direct	Move direct byte to Register	2	24
MOV Rn,#data	Move immediate data to Register	2	12
MOV direct,A	Move Accumulator to direct byte	2	12
MOV direct,Rn	Move Register to direct byte	2	24
MOV direct,direct	Move direct byte to direct byte	3	24
MOV direct,@Ri	Move indirect RAM to direct byte	2	24
MOV direct,#data	Move immediate data to direct byte	3	24
MOV @Ri,A	Move Accumulator to indirect RAM	1	12

LA FAMILIA MCS-81 DE INTEL

Mnemonic	Description	Byte	Decoder Period
DATA TRANSFER (Continued)			
MOV @Ri, direct	Move direct byte to indirect RAM	2	24
MOV @Ri, #data	Move immediate data to indirect RAM	2	12
MOV DPTR, #data16	Load Data Pointer with a 16-bit constant	3	24
MOVC A, @A + DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVC A, @A + PC	Move Code byte relative to PC to Acc	1	24
MOVX A, @Ri	Move External RAM (8-bit addr) to Acc	1	24
MOVX A, @DPTR	Move External RAM (16-bit addr) to Acc	1	24
MOVX @RiA	Move Acc to External RAM (8-bit addr)	1	24
MOVX @DPTR, A	Move Acc to External RAM (16-bit addr)	1	24
PUSH direct	Push direct byte onto stack	2	24
POP direct	Pop direct byte from stack	2	24
ICH A, Ri	Exchange register with Accumulator	1	12
ICH A, direct	Exchange direct byte with Accumulator	2	12
ICH A, @Ri	Exchange indirect RAM with Accumulator	1	12
IOCH A, @Ri	Exchange low- order Dg1 indirect RAM with Acc	1	12

Mnemonic	Description	Byte	Decoder Period
BOOLEAN VARIABLE MANIPULATION			
CLR C	Clear Carry	1	12
CLR bit	Clear direct bit	2	12
SETB C	Set Carry	1	12
SETB bit	Set direct bit	2	12
CPL C	Complement Carry	1	12
CPL bit	Complement direct bit	2	12
ANL C, bit	AND direct bit to CARRY	2	24
ANL C, /bit	AND complement of direct bit to Carry	2	24
ORL C, bit	OR direct bit to Carry	2	24
ORL C, /bit	OR complement of direct bit to Carry	2	24
MOV C, bit	Move direct bit to Carry	2	12
MOV bit, C	Move Carry to direct bit	2	24
JC rel	Jump if Carry is set	2	24
JNC rel	Jump if Carry not set	2	24
JB bit, rel	Jump if direct Bit is set	3	24
JNB bit, rel	Jump if direct Bit is Not set	3	24
JBC bit, rel	Jump if direct Bit is set & clear bit	3	24
PROGRAM BRANCHING			
ACALL addr11	Absolute Subroutine Call	2	24
LCALL addr16	Long Subroutine Call	3	24
RET	Return from Subroutine	1	24
RETI	Return from Interrupt	1	24
AJMP addr11	Absolute Jump	2	24
LJMP addr16	Long Jump	3	24
SJMP rel	Short Jump (relative addr)	2	24

LA FAMILIA MCS-81 DE INTEL

Mnemonic	Description	Byte	Oscillator Period
PROGRAM BRANCHING (Continued)			
JMP @A+DPTR	Jump indirect relative to the DPTR	1	24
JZ rel	Jump if Accumulator is Zero	2	24
JNZ rel	Jump if Accumulator is Not Zero	2	24
CJNE A,direct,rel	Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3	24

Mnemonic	Description	Byte	Oscillator Period
PROGRAM BRANCHING (Continued)			
CJNE Rn,#data,rel	Compare immediate to register and Jump if Not Equal	3	24
CJNE @Ri,#data,rel	Compare immediate to indirect and Jump if Not Equal	3	24
DJNZ Rn,rel	Decrement register and Jump if Not Zero	2	24
DJNZ direct,rel	Decrement direct byte and Jump if Not Zero	3	24
NOP	No Operation	1	12

APENDICE B

CARACTERISTICAS GENERALES DEL MPSC-8274

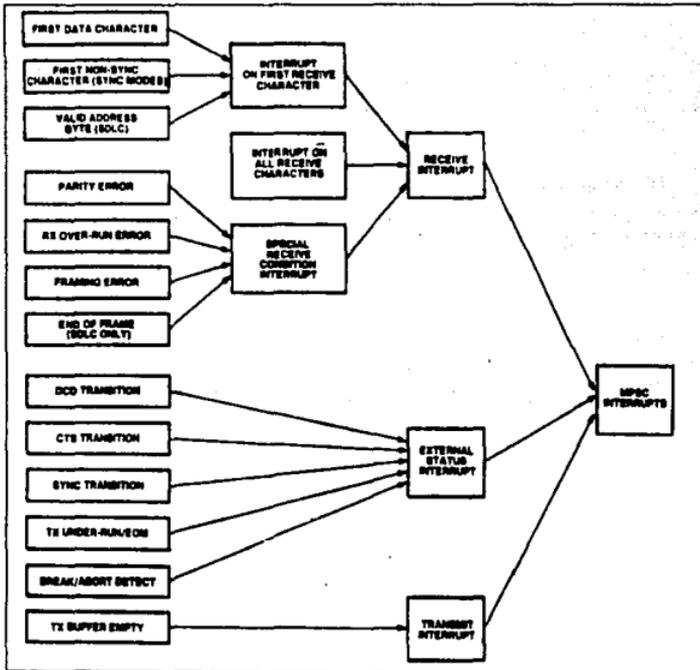


Fig. B.1 Esquema General de Interrupciones

El MPSC ofrece una estructura de interrupción muy poderosa, lo cual ayuda a que una respuesta a la condición de interrupción sea muy rápida. Hay múltiples fuentes de interrupción, las cuales pueden ser agrupadas dentro de tres categorías distintas, como se muestra en la figura B.1:

INTERRUPCIONES EN LA RECEPCION

Por lo que se refiere al primer grupo, interrupciones en la recepción, estas pueden ser clasificadas a su vez en dos grupos: Interrupción en el carácter de recepción e interrupciones debidas a condiciones especiales.

Una interrupción en la recepción es generada cuando un carácter es recibido por el MPSC, es decir, que este tipo de interrupción es generada por el MPSC después de que el carácter recibido ha sido ensamblado por el mismo MPSC. Cuando se está trabajando con el modo SDLC, y además el comando de encuentro de la dirección ha sido programado, esta interrupción se generaría sólo después de que el encuentro de la dirección válida se realiza, es decir, que esta interrupción marca el comienzo de una trama válida.

Así mismo, el final de la trama esta caracterizado por una interrupción (RR1; D7). Este bit es puesto y significa que una bandera de terminación válida ha ocurrido (01111110 - 7E H). Esta interrupción es reseteada por medio del comando reseteo de error "error reset" (WR0; D5D4D3 = 110) o hasta que se de la recepción del primer carácter de la siguiente trama.

Por otro lado, cualquier condición de error durante la recepción de una trama podría generar otro tipo de interrupción, conocida como interrupción debida a condiciones especiales. Este tipo de interrupciones de deben principalmente a cuatro diferentes condiciones:

- a) Error de paridad: Este tipo de error no se considero en nuestro estudio debido a que es encontrado solamente en el caso de usar el protocolo asíncrono.
- b) Error en la recepción sobrecorrida "receive overrun": Este tipo de interrupción se genera cuando la CPU no alcanza a leer un carácter después de tres bytes de tiempo de que ocurrió la interrupción y por lo tanto se generaría la interrupción debido a condiciones especiales y sería puesto el bit 5 en el registro RR1.
- c) Error en la trama: En el caso de los protocolos síncronos este error indica que el CRC calculado en la recepción no es igual al CRC calculado en la transmisión por lo que el bit D6 del registro RR1 es puesto en uno.
- d) Final de la trama: Se genera una interrupción cuando el MPSC recibe la bandera de terminación y es entonces que se genera un uno en el bit D7 del registro RR1.

INTERRUPCIONES EN LA TRANSMISION

Cuando se tiene el "buffer" de transmisión vacío se genera este tipo de interrupción. Cabe hacer notar que este tipo de interrupción es generada si y sólo si el "buffer" de transmisión llega a estar vacío, asumiendo que ha sido cargado con anterioridad algún otro carácter. Esto es porque en el comienzo de la transmisión de una trama, el primer carácter es cargado por la CPU sin tener la interrupción de "buffer" vacío. Después de que este carácter es cargado en el registro serial de cambio, el "buffer" llega a estar vacío y entonces se genera este tipo de interrupción.

INTERRUPCIONES EXTERNAS/ESTADO

Cuando el "buffer" de transmisión y el registro de cambio serial llegan a estar vacíos una interrupción por transmisión recorrida "transmit under-run" se generaría. Otra de las fuentes que causarían este tipo de interrupción sería cuando se presenta el comando de detección de aborto.

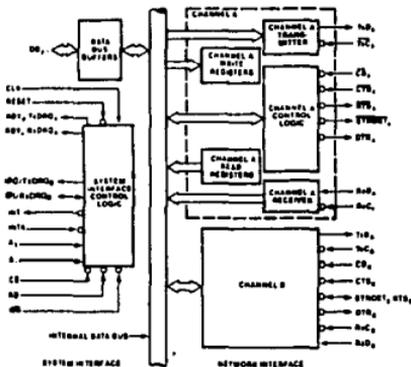


8274 MULTI-PROTOCOL SERIAL CONTROLLER (MPSC)

- Asynchronous, Byte Synchronous and Bit Synchronous Operation
- Two Independent Full Duplex Transmitters and Receivers
- Fully Compatible with 8048, 8051, 8085, 8088, 8086, 80188 and 80186 CPU's; 8257 and 8237 DMA Controllers; and 8089 I/O Proc.
- 4 Independent DMA Channels
- Baud Rate: DC to 880K Baud
- Asynchronous:
 - 5-8 Bit Character; Odd, Even, or No Parity; 1, 1.5 or 2 Stop Bits
 - Error Detection: Framing, Overrun, and Parity
- Byte Synchronous:
 - Character Synchronization, Int. or Ext.
 - One or Two Sync Characters
 - Automatic CRC Generation and Checking (CRC-16)
 - IBM Bitync Compatible
- Bit Synchronous:
 - SDLC/HDLC Flag Generation and Recognition
 - 8 Bit Address Recognition
 - Automatic Zero Bit Insertion and Deletion
 - Automatic CRC Generation and Checking (CCITT-16)
 - CCITT X.25 Compatible
- Available in EXPRESS and Military

The Intel 8274 Multi-Protocol Serial Controller (MPSC) is designed to interface High Speed Communications Lines using Asynchronous, IBM Bitync, and SDLC/HDLC protocol to Intel microcomputer systems. It can be interfaced with Intel's MCS-48, -85, -51; iAPX-86, -88, -186 and -188 families, the 8237 DMA Controller, or the 8089 I/O Processor in polled, interrupt driven, or DMA driven modes of operation.

The MPSC is a 40 pin device fabricated using Intel's High Performance HMOS Technology.



APENDICE C

8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
 - 24 Programmable I/O Pins
 - Completely TTL Compatible
 - Fully Compatible with Intel Microprocessor Families
 - Improved Timing Characteristics
 - Direct Bit Set/Reset Capability Easing Control Application Interface
 - Reduces System Package Count
 - Improved DC Driving Capability
 - Available in EXPRESS
 - Standard Temperature Range
 - Extended Temperature Range
 - 40 Pin DIP Package or 44 Lead PLCC
- (See Intel Packaging Order Number: 231308)

The Intel 8255A is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

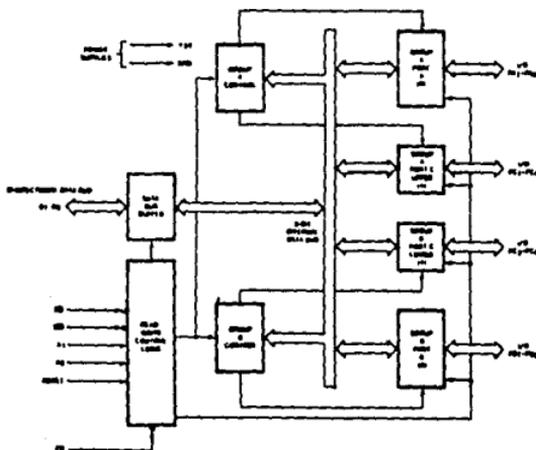


Figure 1. 8255A Block Diagram

231308-1

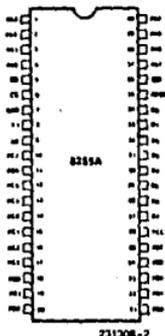


Figure 2. Pin Configuration



54LS00/DM54LS00/DM74LS00 Quad 2-Input NAND Gates

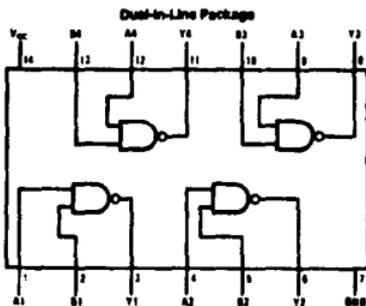
General Description

This device contains four independent gates each of which performs the logic NAND function.

Features

- Alternate Military/Aerospace device (54LS00) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



TL7/5420-1

Order Number 54LS00DM08, 54LS00P808, 54LS00LM08, DM54LS00J, DM54LS00W, DM74LS00M or DM74LS00M
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$Y = \overline{AB}$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H = High Logic Level
L = Low Logic Level



54LS04/DM54LS04/DM74LS04 Hex Inverting Gates

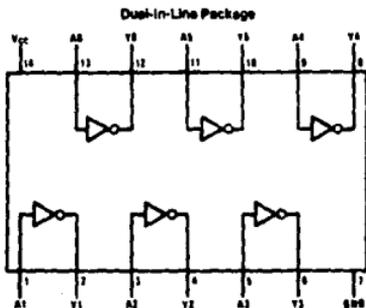
General Description

This device contains six independent gates each of which performs the logic INVERT function.

Features

- Alternate Military/Aerospace device (54LS04) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



T. 6345-1

Order Number 54LS04DMQB, 54LS04FMB, 54LS04LMB, DM54LS04J, DM54LS04W, DM74LS04M or DM74LS04N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$$Y = \bar{A}$$

Input	Output
A	Y
L	H
H	L

H = High Logic Level
L = Low Logic Level



54LS08/DM54LS08/DM74LS08 Quad 2-Input AND Gates

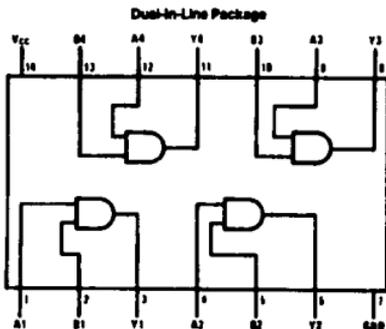
General Description

This device contains four independent gates each of which performs the logic AND function.

Features

- Alternate Military/Aerospace device (54LS08) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



TL/F/3347-1

Order Number 54LS08DM08, 54LS08FM08, 54LS08LM08, DM54LS08J, DM54LS08W, DM74LS08M or DM74LS08N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$Y = AB$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = High Logic Level

L = Low Logic Level



54LS85/DM54LS85/DM74LS85 4-Bit Magnitude Comparators

General Description

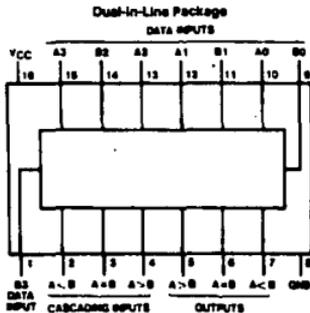
These 4-bit magnitude comparators perform comparison of straight binary or BCD codes. Three fully-decoded decisions about two, 4-bit words (A, B) are made and are externally available at three outputs. These devices are fully expandable to any number of bits without external gates. Words of greater length may be compared by connecting comparators in cascade. The $A > B$, $A < B$, and $A = B$ outputs of a stage handling less-significant bits are connected to the corresponding inputs of the next stage handling more-significant bits. The stage handling the least-significant bits must

have a high-level voltage applied to the $A = B$ input. The cascading path is implemented with only a two-gate-level delay to reduce overall comparison times for long words.

Features

- Typical power dissipation 52 mW
- Typical delay (4-bit words) 24 ns
- Alternate Military/Aerospace device (54LS85) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



Order Number 54LS85DMOB,
54LS85FMOB, 54LS85LMOB,
DM54LS85J, DM54LS85W,
DM74LS85M or DM74LS85N
See NS Package Number E20A,
J16A, M16A, N16E or W16A

TL7/8278-1

Function Table

Comparing Inputs				Cascading Inputs			Outputs		
A3, B3	A2, B2	A1, B1	A0, B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 < B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 < B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 < B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 < B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	H	L	L	H
A3 < B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 < B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

H = High Level, L = Low Level, X = Don't Care



54LS138/DM54LS138/DM74LS138, 54LS139/DM54LS139/DM74LS139 Decoders/Demultiplexers

General Description

These Schottky-clamped circuits are designed to be used in high-performance memory-decoding or data-routing applications, requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When used with high-speed memories, the delay times of these decoders are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoder is negligible.

The LS138 decodes one-of-eight lines, based upon the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented with no external inverters, and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The LS139 comprises two separate two-line-to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

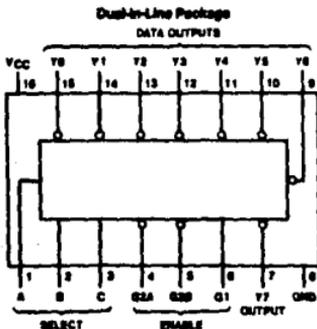
All of these decoders/demultiplexers feature fully buffered inputs, presenting only one normalized load to its driving circuit. All inputs are clamped with high-performance

Schottky diodes to suppress line-bouncing and simplify system design.

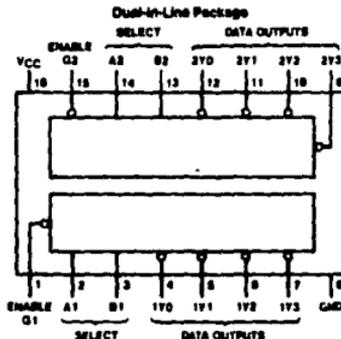
Features

- Designed specifically for high speed:
 - Memory decoders
 - Data transmission systems
- LS138 3-to-8-line decoder incorporates 3 enable inputs to simplify cascading and/or data reception
- LS139 contains two fully independent 2-to-4-line decoders/demultiplexers
- Schottky clamped for high performance
- Typical propagation delay (3 levels of logic)
 - LS138 21 ns
 - LS139 21 ns
- Typical power dissipation
 - LS138 32 mW
 - LS139 34 mW
- Alternate Military/Aerospace devices (54LS138, 54LS139) are available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagrams



Order Number 54LS138D808, 54LS138F808,
54LS138L808, DM54LS138J, DM54LS138W,
DM74LS1388 or DM74LS1389
See NS Package Number E28A, J16A,
M16A, N10E or W16A



Order Number 54LS139D808, 54LS139F808,
54LS139L808, DM54LS139J, DM54LS139W,
DM74LS1398 or DM74LS1399
See NS Package Number E28A, J16A,
M16A, N10E or W16A



54LS244/DM74LS244 Octal TRI-STATE® Buffers/Line Drivers/Line Receivers

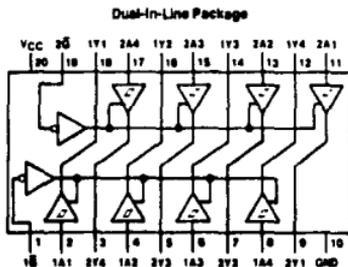
General Description

These buffers/line drivers are designed to improve both the performance and PC board density of TRI-STATE buffers/drivers employed as memory-address drivers, clock drivers, and bus-oriented transmitters/receivers. Featuring 400 mV of hysteresis at each low current PNP data line input, they provide improved noise rejection and high fanout outputs and can be used to drive terminated lines down to 133Ω.

Features

- TRI-STATE outputs drive bus lines directly
- PNP inputs reduce DC loading on bus lines
- Hysteresis at data inputs improves noise margins
- Typical I_{OL} (sink current)
 - 54LS 12 mA
 - 74LS 24 mA
- Typical I_{OH} (source current)
 - 54LS -12 mA
 - 74LS -15 mA
- Typical propagation delay times
 - Inverting 10.5 ns
 - Noninverting 12 ns
- Typical enable/disable time 18 ns
- Typical power dissipation (enabled)
 - Inverting 130 mW
 - Noninverting 135 mW

Connection Diagram



Order Number 54LS244DMQB, 54LS244FQB, 54LS244LMOB,
DM74LS244WB or DM74LS244N
See NS Package Number E20A, J20A, M20B, N20A or W20A

TL 7 8442-1

Function Table

Inputs		Output
\bar{G}	A	Y
L	L	L
L	H	H
H	X	Z

L = Low Logic Level
H = High Logic Level
X = Either Low or High Logic Level
Z = High Impedance



**National
Semiconductor**

54LS245/DM54LS245/DM74LS245 TRI-STATE® Octal Bus Transceiver

General Description

These octal bus transceivers are designed for asynchronous two-way communication between data buses. The control function implementation minimizes external timing requirements.

The device allows data transmission from the A bus to the B bus or from the B bus to the A bus depending upon the logic level at the direction control (DIR) input. The enable input (G) can be used to disable the device so that the buses are effectively isolated.

- PNP inputs reduce DC loading on bus lines
- Hysteresis at bus inputs improve noise margins
- Typical propagation delay times, port-to-port 8 ns
- Typical enable/disable times 17 ns

- I_{OL} (sink current)
 - 54LS 12 mA
 - 74LS 24 mA

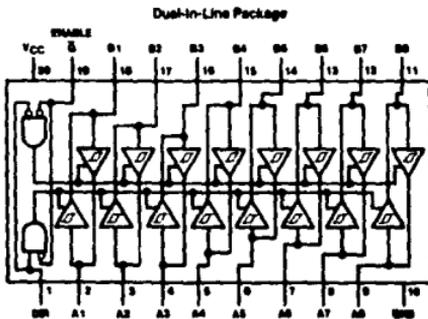
- I_{OH} (source current)
 - 54LS -12 mA
 - 74LS -15 mA

- Alternate Military/Aerospace device (54LS245) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Features

- B-Directional bus transceiver in a high-density 20-pin package
- TRI-STATE outputs drive bus lines directly

Connection Diagram



Order Number 54LS245DMQB, 54LS245FQB, 54LS245LQB,
DM54LS245J, DM54LS245W, DM74LS245W or DM74LS245N
See NS Package Number E20A, J20A, M20B, N20A or W20A

TUF7412-1

Function Table

Enable G	Direction Control DIR	Operation
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

H = High Level, L = Low Level, X = No Connection



DM54LS373/DM74LS373, 54LS374/DM54LS374/DM74LS374 TRI-STATE® Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

General Description

These 8-bit registers feature totem-pole TRI-STATE outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance state and increased high-logic level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers. I/O ports, bidirectional bus drivers, and working registers.

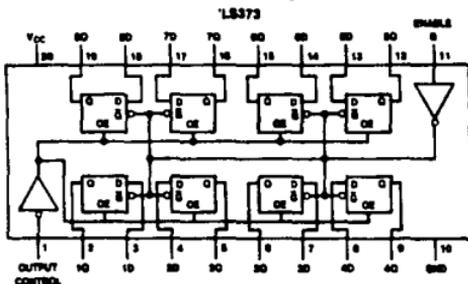
(Continued)

Features

- Choice of 8 latches or 8 D-type flip-flops in a single package
- TRI-STATE bus-driving outputs
- Full parallel-access for loading
- Buffered control inputs
- P-N-P inputs reduce D-C loading on data lines
- Alternate military/aerospace device (54LS374) is available. Contact a National Semiconductor sales office/distributor for specifications.

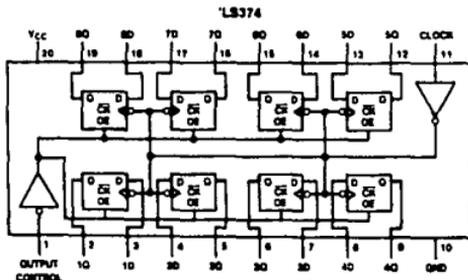
Connection Diagrams

Dual-In-Line Packages



Order Number
DM54LS373J,
DM54LS373W,
DM74LS373N or
DM74LS373WN
See NS Package Number
J20A, M20B, N20A or
W20A

TLF/6431-1



Order Number
54LS374DM08,
54LS374FM08,
54LS374LM08,
DM54LS374J,
DM54LS374W,
DM74LS374N or
DM74LS374N
See NS Package Number
E20A, J20A, M20B, N20A
or W20A

TLF/6431-2

General Description (Continued)

The eight latches of the DM54/74LS373 are transparent D-type latches meaning that while the enable (G) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was set up.

The eight flip-flops of the DM54/74LS374 are edge-triggered D-type flip-flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were set up at the D inputs.

A buffered output control input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or a high-impedance state in the high-impedance state the outputs neither load nor drive the bus lines significantly.

The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are off.

Function Tables

DM54/74LS373

Output Control	Enable G	D	Output
L	H	H	H
L	H	L	L
L	L	X	Q ₀
H	X	X	Z

DM54/74LS374

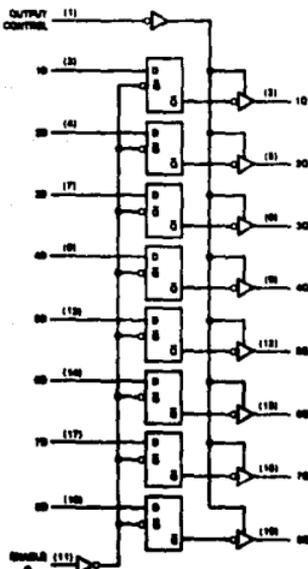
Output Control	Clock	D	Output
L	↑	H	H
L	↑	L	L
L	L	X	Q ₀
H	X	X	Z

H = High Level (Steady State), L = Low Level (Steady State), X = Don't Care
 ↑ = Transition from low-to-high level, Z = High Impedance State
 Q₀ = The level of the output before steady-state input conditions were established

Logic Diagrams

DM54/74LS373

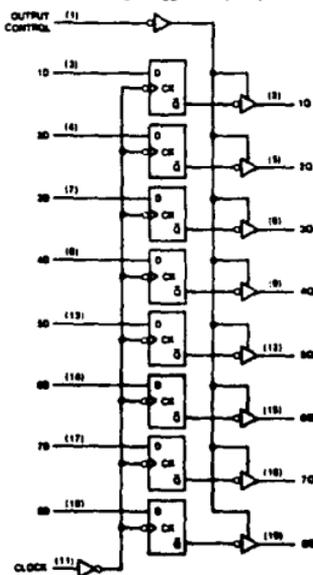
Transparent Latches



TU/P/6631-3

DM54/74LS374

Positive-Edge-Triggered Flip-Flops



TU/P/6631-4

APENDICE D

Service	Interrupt	Register		Notes
		Input	Output	
Print screen.	05H	N/A	N/A	Send screen contents to printer. Status and result byte at 0050:0000H.

Video Services

Set video mode.	10H	AH = 00H AL = video mode	None	<p><i>Video modes in AL:</i></p> <p>00H: 40 × 25 16-color text (gray-scaled on composite monitors).</p> <p>01H: 40 × 25 16-color text.</p> <p>02H: 80 × 25 16-color text (gray-scaled on composite monitors).</p> <p>03H: 80 × 25 16-color text.</p> <p>04H: 320 × 200 4-color graphics.</p> <p>05H: 320 × 200 4-color graphics (gray-scaled on composite monitors).</p> <p>06H: 640 × 200 2-color graphics.</p> <p>07H: 80 × 25 monochrome text (MDA, EGA, VGA).</p> <p>0DH: 320 × 200 16-color graphics (EGA, VGA).</p> <p>0EH: 640 × 200 16-color graphics (EGA, VGA).</p> <p>0FH: 640 × 350 monochrome graphics (EGA, VGA).</p> <p>10H: 640 × 350 16-color graphics (EGA, VGA).</p> <p>11H: 640 × 480 2-color graphics (MCGA, VGA).</p> <p>12H: 640 × 480 16-color graphics (VGA).</p> <p>13H: 320 × 200 256-color graphics (MCGA, VGA).</p>
-----------------	-----	-----------------------------	------	--

Service	Interrupt	Register		Notes
		Input	Output	
Set cursor size.	10H	AH = 01H CH = starting scan line CL = ending scan line	None	Useful values for CH and CL depend on video mode.
Set cursor position.	10H	AH = 02H BH = display page DH = row DL = column	None	
Read cursor position.	10H	AH = 03H BH = display page	CH = starting scan line CL = ending scan line DH = row DL = column	
Read light-pen position.	10H	AH = 04H	AH = pen trigger signal BX = pixel column CH = pixel row (CGA and EGA video modes 4, 5, and 6) CX = pixel row (EGA except modes 4, 5, and 6) DH = character row DL = character column	
Set active display page.	10H	AH = 05H AL = page number	None	
Scroll window up.	10H	AH = 06H AL = lines to scroll up BH = fill attribute CH = upper row CL = left column DH = lower row DL = right column	None	

(continued)

Service	Interrupt	Register		Notes
		Input	Output	
Scroll window down.	10H	AH = 07H AL = lines to scroll down BH = fill attribute CH = upper row CL = left column DH = lower row DL = right column	None	
Read character and attribute.	10H	AH = 08H BH = display page	AH = attribute AL = character	
Write character and attribute.	10H	AH = 09H AL = character BH = display page BL = attribute CX = number of characters to repeat	None	
Write character.	10H	AH = 0AH AL = character BH = page number BL = color in graphics mode CX = number of characters to repeat	None	
Set color palette.	10H	AH = 0BH BH = palette color ID BL = color to be used with palette ID	None	
Write pixel.	10H	AH = 0CH AL = color BH = display page CX = pixel column DX = pixel row	None	

(continued)

Service	Interrupt	Register		Notes
		Input	Output	
Read pixel.	10H	AH = 0DH BH = display page CX = pixel column DX = pixel row	AL = pixel value	
Write character in teletype mode.	10H	AH = 0EH AL = character BH = display page BL = color for graphics mode	None	Display page number required only for IBM PC ROM BIOS dated 10/19/81 and earlier.
Get current video mode.	10H	AH = 0FH	AH = width in characters AL = video mode BH = display page	
Set one palette register.	10H	AH = 10H AL = 00H BH = palette register value BL = palette register number	None	EGA, VGA.
Set border register.	10H	AH = 10H AL = 01H BH = border color	None	EGA, VGA.
Set all palette registers.	10H	AH = 10H AL = 02H ES:DX → table of palette values	None	EGA, VGA.
Select background intensity or blink attribute.	10H	AH = 10H AL = 03H To enable background intensity: BL = 00H To enable blinking: BL = 01H	None	EGA, VGA.

(continued)

Service	Interrupt	Register		Notes
		Input	Output	
Get character generator information.	10H	AH = 11H AL = 30H <i>Contents of interrupt 1FH vector:</i> BH = 00H <i>Contents of interrupt 43H vector:</i> BH = 01H <i>Address of ROM 8 x 14 characters:</i> BH = 02H <i>Address of ROM 8 x 8 characters:</i> 1BH = 03H <i>Address of second half of ROM 8 x 8 table:</i> BH = 04H <i>Address of ROM 9 x 14 alternate characters:</i> BH = 05H <i>Address of ROM 8 x 16 characters:</i> BH = 06H <i>Address of ROM 9 x 16 alternate characters:</i> BH = 07H	CX = points DL = displayed character rows - 1 ES:BP → character table	EGA, MCGA, VGA.
Return video configuration information.	10H	AH = 12H BL = 10H	BH = default BIOS video mode {00H = color, 01H = monochrome} BL = amount of video RAM {00H = 64 KB, 01H = 128 KB, 02H = 192 KB, 03H = 256 KB}	EGA, VGA.

(continued)

Service	Register			Notes
	Interrupt	Input	Output	
Return video configuration information. <i>(continued)</i>			CH = feature bits CL = configuration switches	
Select alternate print screen routine.	10H	AH = 12H BL = 20H	None	EGA, MCGA, VGA. Updates INT 05H vector.
Select scan lines for alphanumeric modes.	10H	AH = 12H BL = 30H 200 scan lines: AL = 00H 350 scan lines: AL = 01H 400 scan lines: AL = 02H	AL = 12H	VGA only.
Select default palette loading.	10H	AH = 12H BL = 31H <i>Enable default palette loading:</i> AL = 00H <i>Disable default palette loading:</i> AL = 01H	AL = 12H	MCGA, VGA.
Enable/disable video addressing.	10H	AH = 12H BL = 32H <i>Enable video addressing:</i> AL = 00H <i>Disable video addressing:</i> AL = 01H	AL = 12H	MCGA, VGA.

(continued)

Service	Interrupt	Register		Notes
		Input	Output	
Enable/disable gray-scale summing.	10H	AH = 12H BL = 33H <i>Enable gray-scale summing:</i> AL = 00H <i>Disable gray-scale summing:</i> AL = 01H	AL = 12H	MCGA, VGA.
Enable/disable BIOS cursor emulation.	10H	AH = 12H BL = 34H <i>Enable cursor emulation:</i> AL = 00H <i>Disable cursor emulation:</i> AL = 01H	AL = 12H	VGA only.
Display switch interface.	10H	AH = 12H BL = 35H <i>Initial adapter video off:</i> AL = 00H <i>Initial planar video on:</i> AL = 01H <i>Switch active video off:</i> AL = 02H <i>Switch inactive video on:</i> AL = 03H ES:DX → 128-byte save area	AL = 12H	MCGA, VGA.
Enable/disable video refresh.	10H	AH = 12H BL = 36H <i>Enable refresh:</i> AL = 00H <i>Disable refresh:</i> AL = 01H	AL = 12H	VGA only.

(continued)

Service	Interrupt	Register		Notes
		Input	Output	
Write string; don't move cursor.	10H	AH = 13H AL = 00H BL = attribute BH = display page DX = starting cursor position CX = length of string ES:BP → start of string	None	PC/AT, EGA, MCGA, VGA.
Write string; move cursor after string.	10H	AH = 13H AL = 01H BL = attribute BH = display page DX = starting cursor position CX = length of string ES:BP → start of string	None	PC/AT, EGA, MCGA, VGA.
Write string of alternating characters and attributes; don't move cursor.	10H	AH = 13H AL = 02H BH = display page DX = starting cursor position CX = length of string ES:BP → start of string	None	PC/AT, EGA, MCGA, VGA.
Write string of alternating characters and attributes; move cursor.	10H	AH = 13H AL = 03H BH = display page DX = starting cursor position CX = length of string ES:BP → start of string	None	PC/AT, EGA, MCGA, VGA.

(continued)

<i>Service</i>	<i>Interrupt</i>	<i>Input</i>	<i>Register</i>	<i>Output</i>	<i>Notes</i>
Keyboard Services					
Read next keystroke.	16H	AH = 00H	AH = scan code AL = ASCII character code		
Report whether keystroke ready.	16H	AH = 01H	ZF = 0 if keystroke available AH = scan code (if ZF = 0) AL = ASCII character code (if ZF = 0)		
Get shift status.	16H	AH = 02H	AL = shift status bits		<i>Shift status bits:</i> bit 7 = 1: Insert state active bit 6 = 1: Caps Lock active bit 5 = 1: Num Lock active bit 4 = 1: Scroll Lock active bit 3 = 1: Alt pressed bit 2 = 1: Ctrl pressed bit 1 = 1: left Shift pressed bit 0 = 1: right Shift pressed
Write to keyboard buffer.	16H	AH = 05H CH = scan code CL = ASCII character code	AL = 00H (success); AL = 01H (keyboard buffer full)		PC/XT (BIOS dated 01/10/86 and later), PC/AT (BIOS dated 11/15/85 and later), and PS/2s only.
Extended keyboard read.	16H	AH = 10H	AH = scan code AL = ASCII character code		PC/XT (BIOS dated 01/10/86 and later), PC/AT (BIOS dated 11/15/85 and later), and PS/2s only.
Extended keyboard status.	16H	AH = 11H	<i>If no keystroke available:</i> ZF = 1 <i>If keystroke available:</i> ZF = 0 AH = scan code AL = ASCII character code		PC/XT (BIOS dated 01/10/86 and later), PC/AT (BIOS dated 11/15/85 and later), and PS/2s only.
Extended shift status.	16H	AH = 12H	AL = shift status (as above) AH = extended shift status: bit 7: Sys Req is pressed bit 6: CapsLock is pressed bit 5: NumLock is pressed bit 4: ScrollLock is pressed bit 3: right Alt is pressed bit 2: right Ctrl is pressed		PC/XT (BIOS dated 01/10/86 and later), PC/AT (BIOS dated 11/15/85 and later), and PS/2s only.

(continued)

<i>Service</i>	<i>Interrupt</i>	<i>Input</i>	<i>Register</i>	<i>Output</i>	<i>Notes</i>
Extended shift status, continued.				bit 1: left Alt is pressed bit 0: left Ctrl is pressed	
Printer Services					
Send 1 byte to printer.	17H	AH = 00H AL = character DX = printer number	AH = success/ failure status flags		<i>Status bit settings:</i> bit 7 = 1: not busy bit 6 = 1: acknowledge bit 5 = 1: out of paper bit 4 = 1: selected bit 3 = 1: I/O error bit 2 = unused bit 1 = unused bit 0 = time out
Initialize printer.	17H	AH = 01H DX = printer number	AH = status code		Status code bit settings as above.
Get printer status.	17H	AH = 02H DX = printer number	AH = status code		Status code bit settings as above.
Miscellaneous Services					
Switch control to ROM BASIC.	18H	None	N/A		No return, so no possible output.
Reboot computer.	19H	None	N/A		No return, so no possible output.

<i>Service</i>	<i>Function (hex)</i>	<i>Register Input</i>	<i>Output</i>	<i>DOS Version</i>	<i>Notes</i>
Program Control Functions					
Terminate: End program.	00H	AH = 00H CS = segment of PSP		1.0	Obsolete: Use function 4CH instead.
Create new program segment.	26H	AH = 26H DX = segment where new PSP starts		1.0	Obsolete: Use function 4BH instead.
Terminate and stay resident.	31H	AH = 31H AL = return code DX = # of paragraphs to keep resident		2.0	
Get/set Ctrl-C flag.	33H	AH = 33H <i>To set flag:</i> AL = 01H DL = value <i>To get flag:</i> AL = 00H	AL = result code <i>If called with AL = 01H:</i> DL = current value of flag (0 = off, 1 = on)	2.0	
EXEC: Load and execute a program.	4BH	AH = 4BH DS:DX → ASCIIZ command line ES:BX → control block <i>To execute child program:</i> AL = 00H	<i>If no error:</i> CF clear <i>If error:</i> CF set AX = error code	2.0	Changes all registers, including SS:SP.

Figure 18-4. A summary of the DOS interrupt 21H functions.

(continued)

<i>Service</i>	<i>Function (hex)</i>	<i>Register Input</i>	<i>Output</i>	<i>DOS Version</i>	<i>Notes</i>
EXEC: Load and execute a program. (continued)		To load without executing: AL = 03H			
Terminate with return code.	4CH	AH = 4CH AL = return code		2.0	
Get return code.	4DH	AH = 4DH	AL = return code AH = termination method	2.0	Call only once after calling function 4CH.
Get PSP address.	62H	AH = 62H	BX = PSP segment	3.0	
Standard Input Functions					
Character input with echo.	01H	AH = 01H	AL = 8-bit character	1.0	
Direct character input without echo.	07H	AH = 07H	AL = 8-bit character	1.0	
Character input without echo.	08H	AH = 08H	AL = 8-bit character	1.0	
Buffered keyboard input.	0AH	AH = 0AH DS:DX → input buffer	Buffer contains keyboard input.	1.0	See Chapter 16 for input buffer format.
Check keyboard status.	0BH	AH = 0BH	If character available: AL = FFH If no character available: AL = 00H	1.0	
Flush keyboard buffer, read keyboard.	0CH	AH = 0CH AL = function number (01H, 06H, 07H, 08H, or 0AH)	(Depends on function specified in AL)	1.0	

(continued)

<i>Service</i>	<i>Function (hex)</i>	<i>Register Input</i>	<i>Output</i>	<i>DOS Version</i>	<i>Notes</i>
Get/set global code page.	66H	AH = 66H <i>To get current code page:</i> AL = 01H <i>To set code page:</i> AL = 02H BX = code page number	<i>If error:</i> CF set AX = error code <i>If no error:</i> CF clear <i>If called with AL = 01H:</i> BX = current code page DX = default code page	3.3	
Memory Functions					
Allocate memory block.	48H	AH = 48H BX = size of block in paragraphs	<i>If error:</i> CF set AX = error code BX = size of largest available block <i>If no error:</i> CF clear AX = paragraph address of allocated block	2.0	
Free memory block.	49H	AH = 49H ES = paragraph address of memory block	<i>If error:</i> CF set AX = error code <i>If no error:</i> CF clear	2.0	
Resize memory block.	4AH	AH = 4AH BX = new size of memory block in paragraphs ES = paragraph address of memory block	<i>If error:</i> CF set AX = error code BX = size of largest available block (if increased size was requested)	2.0	
Resize memory block. <i>(continued)</i>			<i>If no error:</i> CF clear		
Get/set memory allocation strategy.	58H	AH = 58H <i>To get allocation strategy:</i> AL = 00H <i>To set allocation strategy:</i> AL = 01H BX = strategy code	<i>If error:</i> CF set AX = error code <i>If no error:</i> CF clear <i>If called with AL = 00H:</i> AX = strategy code	3.0	See Chapter 17 for details.

APENDICE E

E. SISTEMAS DE RESPALDO CON BATERIA PARA EL 80C31BH

En el diseño de sistemas de control industrial es necesario considerar circuitos que en condiciones normales son alimentados por la línea de CA, pero que en el caso de una falla en el suministro conmutan a una operación con batería. Aquí asumimos que en operación con batería las demandas de alta corriente cesarán junto con la línea de CA. El sistema podrá continuar con limitada funcionalidad, probablemente monitoreando un transductor de control o manejando un LCD. O irá a un modo de supervivencia, en el cual se guarden datos críticos pero no sucederá algo más sino hasta que la línea de CA se reestablezca.

De cualquier modo es necesario tener un previo aviso de una eventual falla en el suministro, para que el sistema pueda realizar una transferencia ordenada a alimentación con batería. Los sistemas de aviso previo pueden operar ya sea monitoreando la línea de voltaje de CA o la salida no regulada del rectificador, o inclusive monitoreando el voltaje regulado de CD.

El monitorear la línea de CA da el aviso más temprano. De esa manera se puede saber si una falla ha ocurrido entre uno o dos medios ciclos de la frecuencia de línea. En la mayoría de los casos, se cuenta de por lo menos otro medio ciclo de la frecuencia de línea antes de que empiece a caer el VCC regulado. En un medio ciclo de la frecuencia de línea un 8031 puede ejecutar alrededor de 5,000 instrucciones, suficiente tiempo como para efectuar una ordenada transferencia de poder.

El circuito de la figura E.1 utiliza un diodo Zener para probar cada medio ciclo el voltaje de línea, y un transistor para pasar la información al 8031. (Obviamente para realizar la misma función se puede emplear un comparador de voltaje con una referencia adecuada.) Su operación se basa en que al presentarse un nivel aceptablemente alto, el Z1 satura al Q1 e interrumpe al 8031. En este caso la interrupción será activada por cambio de nivel. La rutina de atención a la interrupción recarga uno de los temporizadores del 8031 con un valor que lo hará regresar al valor inicial dentro de uno y dos medios ciclos de la frecuencia de línea. Mientras que la línea de voltaje esté sana, el temporizador nunca regresa al valor inicial, ya que se recarga cada medio ciclo. Si llega a haber un medio ciclo en el cual el voltaje de línea no alcance un nivel alto aceptable como para generar la interrupción, el temporizador regresa al valor inicial y genera una interrupción del temporizador.

Entonces la interrupción del temporizador comienza la transición al respaldo con batería. Los datos críticos necesitan ser copiados a una RAM protegida. Las señales a circuitos que perderán potencia deberán escribirse con nivel lógico bajo. Se deben deshabilitar los circuitos protegidos (aquellos alimentados por VCC2) que se comunican con circuitos desprotegidos. El microcontrolador en sí debe ser puesto en

APÉNDICE E

modo de espera (el modo de espera y el modo apagado son particulares de los dispositivos CHMOS de la familia MCS-51), para que pueda continuar con algún nivel de manejo por interrupción, o bien se puede poner en modo apagado.

Notese que si el CPU invocará el modo apagado, los registros de funciones especiales (SFRs) también tendrán que ser copiados a la RAM protegida, ya que el reset que termina con el modo apagado también inicializa todos los SFRs a su valor inicial.

El circuito de la figura E.1 no muestra un mecanismo restablecedor. Sin embargo, existen muchas opciones. Un botón se puede utilizar para generar una interrupción, si el CPU esta en modo de espera, o activar un reset, si el CPU está en modo apagado.

También es posible un restablecedor automático al restaurarse la alimentación. Si el CPU esta en modo de espera, puede seguir respondiendo a las interrupciones provenientes de Q1. La rutina de atención a la interrupción determina del estado de los bits de bandera GF0 y GF1 en PCON (Power Control Register) que está en modo de espera al haber una falla. Entonces puede muestrear VCC1 a través de un comparador similar a Z1, Q1 de la figura E.1. Un nivel satisfactorio de VCC1 será efectivo al estar saturado el transistor.

Pero probablemente no se pueda emplear el otro temporizador, que es la clave del funcionamiento del circuito de la figura E.1. En tal caso un one-shot redispensible, disparado por la línea de CA, puede en esencia realizar la misma función. La figura E.2 muestra un ejemplo de este tipo de detector de falla. Un one-shot redispensible (la mitad de un 74HC123) monitorea el voltaje de la línea de CA a través del transistor Q1. El Q1 redisparsa el one-shot cada medio ciclo de la frecuencia de línea. Si el ancho del pulso de salida se encuentra entre uno y dos medio ciclos de la frecuencia de línea, entonces un único medio ciclo faltante o muy bajo generará una bandera de un bajo activo, la cual puede ser empleada para interrumpir al microcontrolador.

La rutina de interrupción se encarga de la transición al respaldo con batería. A partir de entonces el VCC1 puede o no desaparecer. El medio ciclo faltante del voltaje de línea pudo haber sido no más que una pequeña irregularidad en la línea. Si la línea regresa con la fuerza suficiente para disparar el one-shot mientras que VCC1 sigue en alto (como lo indica el estado del transistor Q2), entonces la otra mitad del 74HC123 generará una señal de restablecimiento.

Una vez restablecido, el 8031 permanecerá así por lo menos por otro medio ciclo de la frecuencia de línea (otras 5,000 instrucciones) antes de que posiblemente se le diga que realice otra transferencia de poder. Consecuentemente, si la línea de voltaje está fallando constantemente alrededor del punto de conmutación (determinado por el diodo Z1), el sistema estará ejecutando en unidades de medio ciclo de la frecuencia de línea.

Por otro lado, si la falla es real y tardada, eventualmente VCC1 caerá al nivel donde se dá el cambio a la batería. La batería de respaldo mantendrá la alimentación

del 8031 y del 74HC123, y a todos los demás circuitos que se desee proteger. El voltaje de la batería debe ser lo suficientemente alto como para mantener el valor nominal de V_{CCmin} del 8031.

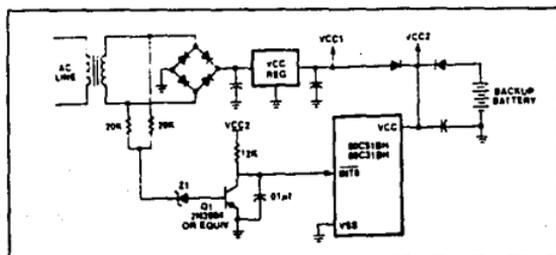


Fig. E.1 Detector de Falla en el Suministro con Respaldo de Batería

Para el caso específico del 8031, que ejecuta de una ROM externa, y si se establece en modo de espera al ocurrir la falla, entonces la ROM externa también debe de ser provista de alimentación por la batería. En caso contrario, si se establece el modo apagado, entonces se puede permitir que la ROM "muera" junto con la línea de CA. Es importante notar que a la hora de restablecimiento, la ROM debe restablecerse antes que el 8031, para que no se caiga en un estado indeterminado.

CIRCUITOS DE CONMUTACION DE POTENCIA

Los sistemas de respaldo con batería necesitan de un modo para que los circuitos protegidos consuman potencia de la fuente de alimentación cuando existe la línea de CA, y que conmuten a la potencia de la batería cuando se requiera. El circuito de conmutación es simple si el sistema completo se alimentará con la batería en el evento de una falla de línea. En tal caso un par de diodos es suficiente, como se muestra en la figura E.2, siempre que el valor nominal de V_{CCmin} se mantenga después de que la caída en el diodo se ha sustraído de su fuente de poder respectiva.

La situación se torna más complicada cuando sólo se quieren proteger algunos circuitos. En tal caso es difícil mantener VCCs iguales para los circuitos protegidos y los no protegidos (y probablemente también sea peligroso).

APENDICE F

F. EJEMPLO DE UN TRANSDUCTOR RESONANTE. EL SENSOR DE PRESION PIEZORESISTIVO MOTOROLA MPX100

INTRODUCCION

En la figura F.1 se muestra el corte transversal del transductor de presión MPX100. La pastilla sensora de presión se ubica dentro de un paquete termoplástico de 0.6" de diámetro y 0.2" de ancho. El montaje se hace con un material especial a una base de aluminio con cuatro alambres de oro que conectan la pastilla con las terminales externas. La cavidad está llena con un gel especial para proteger la pastilla de materiales peligrosos y los alambres de gases o líquidos conductores. Una tapa de aluminio cubre la parte superior. El elemento básico y las partes para su acoplamiento con una manguera de 1/8" se muestran en la figura F.2.

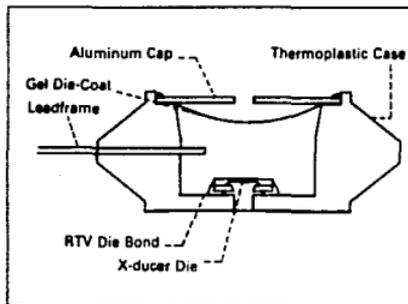


Fig.F.1 Sensor de Presión X-ducer de Motorola

Los sensores de presión de Motorola se pueden conseguir en rangos de presión de 0 a 1.5 PSI, 0 a 7.5 PSI, 0 a 15 PSI y de 0 a 30 PSI. Para todos los rangos se cuenta con el mismo tipo de paquete y técnica de construcción. En la figura F.3 se muestra la relación de presión v.s. voltaje de salida. El valor típico de voltaje de salida para el valor máximo es de 60 mV @ 25°C, el valor del voltaje de "offset" para presión cero es de 20 mV @ 25°C y el valor de su voltaje de excitación es de 3.0 Vdc. El voltaje de salida correspondiente al valor máximo de presión, se decrementa al incrementarse la temperatura e incrementa al decrementarse la temperatura relativa a 25°C.

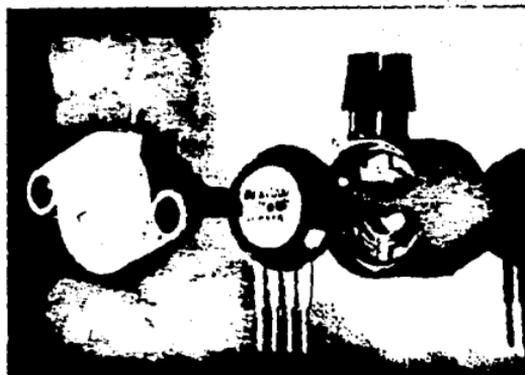


Fig.F.2 Elementos Basicos

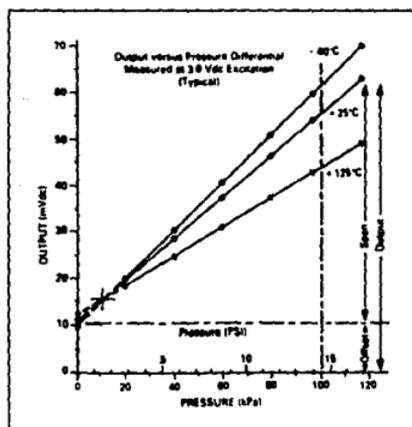


Fig.F.3 Sensor de Presión Piezoresistivo MPX100

A este transductor es necesario acoplarle circuitería adicional para compensación de temperatura y "offset", calibración, y ganancia de voltaje, así como un convertidor de voltaje a frecuencia. Para que sea un transductor resonante completo.

CIRCUITO DE COMPENSACION PARA TEMPERATURA Y "OFFSET"

Este circuito se muestra en la figura F.4, el cual emplea un LM258 amplificador operacional dual. La selección de la resistencia de compensación R_3 , para un rango de temperatura de 0°C a 80°C, se calcula mediante:

$$R_3 = R_x \times 3.57$$

donde:

R_3 = resistencia de compensación

R_x = resistencia del transductor a 25°C

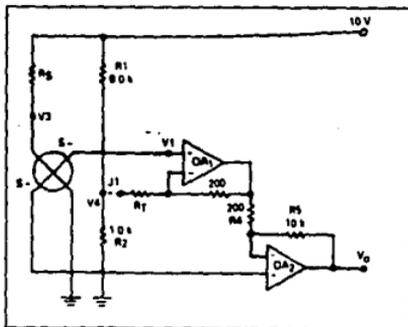


Fig.F.4 Circuito de Calibración de Offset y Temperatura

APENDICE F

Los transductores de presión se especifican con una resistencia mínima de 400 Ω y una máxima de 550 Ω .

Asumiendo que R_S es aproximadamente 4 veces la resistencia del transductor, el voltaje de excitación en V_3 es aproximadamente 2.0 Vdc, si el circuito opera con 10 V. Los voltajes de salida del transductor en las terminales S+ y S- son de aproximadamente 1.0 Vdc.

El divisor de voltaje formado por R_1 y R_2 fijan el voltaje en V_4 a aproximadamente 1.1 Vdc.

Como el transductor tiene un coeficiente de resistencia positivo a la temperatura, el voltaje en V_3 se incrementa con la temperatura, como también el voltaje en V_1 . El transductor tiene su propio coeficiente de temperatura de "offset". Por lo que es necesario calcular el valor de R_T tal que el cambio de V_1 con la temperatura, cancele el cambio de "offset".

Se puede calcular R_T con la siguiente ecuación:

$$R_T = \left(\frac{\Delta V_1}{\Delta V_0} \times R_S \right) - \frac{R_1 \times R_2}{R_1 + R_2}$$

donde:

$$V_0 = V_0 (\text{Temp. máx.}) - V_0 (@ 25^\circ\text{C})$$

$$V_1 = V_1 (\text{Temp. máx.}) - V_1 (@ 25^\circ\text{C})$$

La segunda etapa del amplificador operacional (OA_2) provee una ganancia de 50 (R_5 dividido por R_4). El efecto de R_T es mínimo en la ganancia total de OA_1 y OA_2 . El voltaje de salida V_0 estará compensado para el rango de temperaturas de 0°C a 80°C .

ETAPA DE CALIBRACION

Esta etapa se muestra en la figura F.5. El amplificador de ganancia variable, OA_4 , suma la señal que depende de la presión (V_0) con el voltaje de "offset" variable creado por el divisor de voltaje R_8 , R_9 y R_{off} . OA_3 funciona como "buffer" entre el voltaje de "offset" variable y el control de ganancia variable R_G .

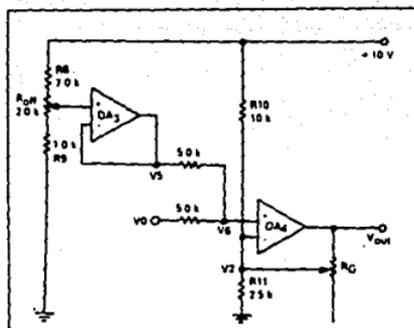


Fig.F.5 Calibración de OA3 y OA4

Si OA3 no estuviese en el circuito, el ajuste de "offset" interactuaría con el ajuste de ganancia, provocando una calibración repetida.

La calibración de temperatura y "offset" (figura F.4.) tiene un voltaje de salida (V_0) de 1.0 a 2.0 V, dependiendo del voltaje de "offset" del transductor a una diferencial de presión cero y del valor de R_8 . El voltaje V_5 se ajusta mediante el divisor de voltaje R_8 , R_9 y R_{0H} . V_6 es el promedio de V_0 y V_5 entre 1.0 y 3.0 Vdc.

Para los circuitos mostrados en las figuras F.4 y F.5, la salida de voltaje para el diferencial de presión cero a través del transductor es de 2.0 V.

Si R_{0H} se ajusta para que ninguna corriente fluya a través de R_G , V_6 será igual a V_{out} y V_2 . El voltaje de salida es 2.0 Vdc.

Usando un valor de 5 k Ω para R_G , se tendrá un voltaje de presión cero de 2.0 Vdc y un voltaje de presión máxima de 4.0 Vdc. Una resistencia de 10 k Ω dará un voltaje máximo de salida de 6.0 Vdc.

CONVERSION DE VOLTAJE DE SALIDA A FRECUENCIA

La conversión de voltaje a frecuencia se puede lograr con el circuito mostrado en la figura F.6. El circuito consiste de un amplificador operacional LM258 y un temporizador

APENDICE F

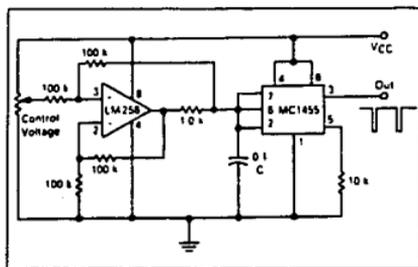


Fig.F.6 Voltaje a Frecuencia MC1455/LM258

555 como multivibrador astable. El LM258 se emplea como una fuente de corriente controlada por voltaje. Este circuito provee una corriente de salida hasta el valor máximo de corriente permitida por el amplificador operacional.

El capacitor C ($0.1 \mu F$) se carga por la fuente constante de corriente y se descarga por la terminal 7 del temporizador 555. La resistencia de $10 k\Omega$ sólo permite que el capacitor se cargue hasta la mitad de el voltaje de la fuente, resultando en un control bastante lineal para la conversión de voltaje a frecuencia.

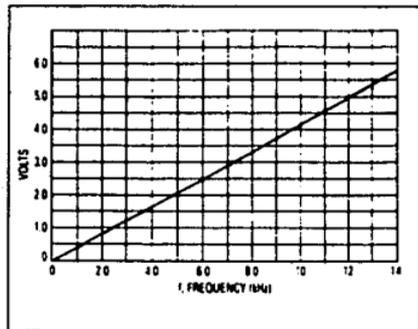


Fig.F.7 Voltaje a Frecuencia ($V_{CC} = 15 V.$)

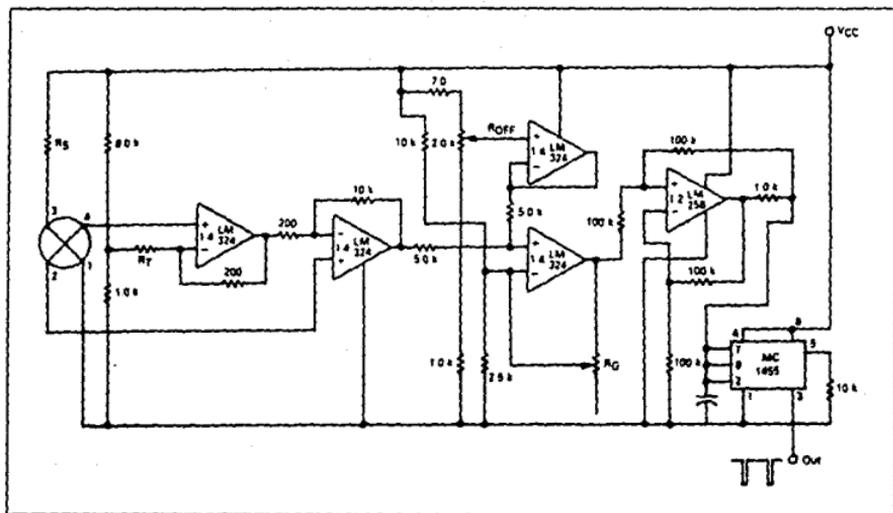


Fig.F.8 Conversión de Presión a Frecuencia

La figura F.7 ilustra gráficamente la relación medida entre el control de voltaje y la frecuencia de salida en el circuito de la figura F.6. La figura F.8 muestra el circuito completo para convertir presión a frecuencia.