



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN

DISEÑO DE UN SISTEMA AUTOMATICO DE  
SUMINISTRO DE GASOLINA EMPLEANDO UN  
MICROPROCESADOR Z-80

T E S I S  
QUE PARA OBTENER EL TITULO DE :  
INGENIERO MECANICO ELECTRICISTA  
PRESENTAN:  
AGUILAR CID LUIS ADRIAN  
BAUTISTA CRUZ GILBERTO  
GIMENEZ SCHERER ALEJANDRO

Cuatitlan Izcalli, Edo. de México

1984



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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 esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO  
FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN

DISEÑO DE UN SISTEMA AUTOMATICO DE SUMINISTRO  
DE GASOLINA EMPLEANDO UN MICROPROCESADOR Z-80

T E S I S

que para obtener el título de:

INGENIERO MECANICO ELECTRICISTA

presentan:

AGUILAR CID LUIS ADRIAN

BAUTISTA CRUZ GILBERTO

GIMENEZ SCHERER ALEJANDRO

Director de Tesis:

M. en I. JOSE FERNANDO GARCIA NUÑEZ CANO

CUAUTITLAN IZCALLI, EDO. DE MEXICO, 1984

J U R A D O

Presidente

IME. ANTONIO HERRERA MEJIA

Vocal

M. en C. JUAN ANTONIO NAVARRO MARTINEZ

Secretario

M. en I. JOSE FERNANDO GARCIA NUÑEZ CANO

Primer Suplente

IME. LOURDES CLARES FUENTES

Segundo Suplente

IME. RICARDO JOAQUIN RAMIREZ VERDEJA

J U R A D O

Presidente

IME. ANTONIO HERRERA MEJIA

Vocal

M. en C. JUAN ANTONIO NAVARRO MARTINEZ

Secretario

M. en I. JOSE FERNANDO GARCIA NUÑEZ CANO

Primer Suplente

IME. LOURDES CLARES FUENTES

Segundo Suplente

IME. RICARDO JOAQUIN RAMIREZ VERDEJA

A nuestros padres

# I N D I C E

---

	página
1.- INTRODUCCION	
La Electrónica	1
Los Semiconductores	1
Los Microprocesadores	2
Areas de aplicación	3
2.- EXPOSICION	
Elementos del sistema	7
Elección del sistema de control	11
Descripción del sistema de control	13
Bombas	16
Motores eléctricos	18
Medidores de flujo	20
Selección del medidor de flujo	29
3.- MICROPROCESADOR	
Microprocesadores	31
Unidad central de proceso	35
Registros del microprocesador	38
Descripción del circuito	40
Instrucciones del microprocesador	47
4.- SISTEMAS DE MICROCOMPUTADORA	
Componentes básicos	80
Circuito de conteo y temporización	83
Arquitectura del Z80 CTC	83
Vector de interrupción	89
Descripción del circuito	90
Modos de operación	96
Bus normalizado S-100	97
Descripción del bus normalizado S-100	98

5.-	PROGRAMACION	
	Programación estructurada	107
	Proceso de desarrollo	110
	Análisis de requerimientos	113
	Especificación funcional	116
	Diseño	118
	Codificación	144
	Mapa de memoria	163
6.-	INTEGRACION	
	Elementos	166
	Nanocomputador	168
	Exhibidores luminosos	175
	Transductor	178
	Interfase al motor	182
	Alimentación	185
7.-	CONCLUSIONES	189
8.-	BIBLIOGRAFIA	191
Anexo:		
	INSTRUCTIVO DE OPERACION	194

# INTRODUCCION

---

## LA ELECTRONICA

Entre las diversas disciplinas científicas y especialidades tecnológicas que han alcanzado un espectacular desarrollo en el presente siglo, la Electrónica debe ser mencionada en uno de los primeros lugares, ya que ha evolucionado rápidamente e influido en una gran variedad de actividades del hombre.

Es la Electrónica el estudio del comportamiento de los electrones, en especial, de sus movimientos en los diversos campos de fuerzas a los que pueden ser sometidos, el eléctrico, el magnético, o al efecto conjunto de ambos campos. De una forma más simplificada se dice que, la Electrónica es el manejo de conexiones eléctricas y su control.

El desarrollo práctico de la Electrónica se inicia con la Radio, que se empieza a comercializar a gran escala en la segunda década del siglo XX, y nuevas aplicaciones como la Televisión aceleran su desarrollo.

Pero el paso importante de la Electrónica se da con la aparición de los semiconductores y, en especial, con la creación del transistor, que genera una completa revolución en la técnica electrónica.

## LOS SEMICONDUCTORES

Un Semiconductor es un elemento cuya resistencia eléctrica es intermedia entre los aisladores y los conductores. En 1906 ya se utilizaban como detectores, en los aparatos de radio primitivos (antes de que se introdujeran las válvulas o bulbos), gran cantidad de cristales semiconductores; siendo estos semiconductores la galena, las piritas de hierro, el carborundum y el silicio.

Una evolución altamente perfeccionada de aquellos detectores de cristal, la constituye el Diodo semiconductor que puede detectar, mezclar, y rectificar señales de corriente alterna con una gran eficiencia. Y como resultado de investigaciones en el campo de los diodos semiconductores, en 1948 fué creado el Transistor, el primer dispositivo semiconductor capaz de amplificar señales eléctricas.

El desarrollo de la Electrónica prosiguió con la integración de varios de estos elementos, que junto a la miniaturización de elementos pasivos, como son las resistencias y los capacitores, fué conduciendo a la llamada Microelectrónica, en la que sobre un material común llamado sustrato, se van construyendo dispositivos que reúnen varios componentes, activos y pasivos, formados simultáneamente sobre este sustrato de silicio y conocidos como circuitos integrados.

Así pues, los circuitos integrados comprenden elementos activos y pasivos y representan la culminación de los esfuerzos tendientes a la miniaturización de los circuitos electrónicos, permitiendo su técnica de fabricación una gran reducción de costo y muchos progresos en lo que se refiere a la seguridad de funcionamiento, pues al carecer de cables o alambres soldados se ha convertido en un componente muy seguro.

Aumentando la cantidad de componentes en un circuito integrado, lograremos nuevas funciones, ya que el circuito integrado con cinco mil transistores nos permite elaborar un reloj digital, mientras que con uno de veinte mil transistores construiremos una calculadora de bolsillo, y con otro de cien mil transistores podremos construir una pequeña computadora con la capacidad de las antiguas computadoras de válvulas.

## LOS MICROPROCESADORES

El desarrollo de los circuitos integrados llevó, en noviembre de 1971, al inicio de una etapa sumamente importante al producirse el primer microprocesador, que es un circuito integrado programable por el usuario, que puede ser empleado para una gran variedad de funciones, desde ser el cerebro de una central telefónica de gran capacidad, a ser el cerebro de un pequeño conmutador para ocho o diez abonados, y tanto podemos encontrarlo en una báscula digital, como en una caja registradora, y por supuesto en la computadoras.

## AREAS DE APLICACION

En la actualidad los microprocesadores se emplean en las siguientes áreas:

- 1.- Computación
- 2.- Industria
- 3.- Sistemas sencillos para recreación y el hogar
- 4.- Aplicaciones especiales

Estas áreas se diferencian entre sí por los diferentes criterios de funcionalidad, seguridad, costo y por el tipo de equipo necesario para complementar al sistema de microprocesador.

## APLICACIONES EN LA COMPUTACION

La primera área de aplicación de los microprocesadores fué la relacionada con la computación y los sistemas de cómputo, siendo empleados como circuitos de soporte y programados para realizar funciones ahora implementados en circuitería no programable. Posteriormente se construyeron computadoras personales de bajo costo, con el microprocesador como unidad central.

La implementación de sistemas ha variado notablemente, ya que la unidad de proceso es ahora uno de los componentes más económicos de un sistema de cómputo, lo cual ha llevado a desarrollar sistemas con varios microprocesadores, que intercambian bloques de datos o palabras, y cada microprocesador es un controlador de proceso.

## APLICACIONES INDUSTRIALES

Los sistemas industriales son la segunda área de aplicación de los microprocesadores, reemplazando a las minicomputadoras o circuitos de lógica cableada compleja, disminuyendo en ambos casos los costos. El impacto principal de los microprocesadores en la industria, ha sido el proveer un gran número de nuevas funciones, que hacen que el control de procesos sea más simple, más poderoso, y más "inteligente", sin incrementar los costos o inclusive, reduciéndolos.

Los microprocesadores han introducido la programación a los sistemas industriales, haciendo posible el utilizar productos estandarizados por un período largo de tiempo.

Las aplicaciones de los microprocesadores en los procesos industriales están caracterizadas por dos puntos esenciales. El primero es que la mayor parte de las aplicaciones requieren procesadores con entradas y salidas analógicas, o sea, el equivalente de un controlador analógico tradicional, y la segunda es que las aplicaciones industriales, en general, están caracterizadas por sensores y mecanismos de control costosos y este alto costo de los sensores requeridos para las entradas y el costo de los dispositivos de control para las salidas, son normalmente mucho mayores que el costo de un sistema de microprocesador normal.

### APLICACIONES PARA EL PUBLICO

La tercera área de aplicación está enfocada al consumo general y está caracterizada por un alto volumen y el menor costo posible. Es el área de aplicación de los microcomputadores de un solo circuito integrado. Cualquier otra arquitectura de microprocesadores más desarrollados, no son adecuados, debido a que el mayor número de componentes encarece el producto. Por ejemplo, un microcomputador de un solo circuito integrado, es suficiente para funciones de control simple, como las que requiere una máquina de lavar o un horno electrónico.

Las ventajas son obvias ya que el microprocesador elimina la lógica cableada o los dispositivos electromecánicos y permite más funciones.

### APLICACIONES ESPECIALES

Las aplicaciones especiales incluyen, la medicina, el área militar, y el área aeroespacial, caracterizadas por una necesidad absoluta de bajo volumen y bajo consumo de potencia, no siendo afectadas en demasía por altos costos de programación.

Las aplicaciones médicas son sin lugar a dudas las más interesantes dentro de esta área, existiendo dos tipos de áreas en que se utilizan los microprocesadores, y son los sistemas implantados en el cuerpo y los sistemas para uso externo.

Recientemente se han desarrollado muchas aplicaciones que implican el uso de microprocesadores que son implantados en el cuerpo, por ejemplo los marcapasos, que proveen estimulación cardíaca en intervalos de tiempo establecidos y pueden ser sustituidos ventajosamente por otros más modernos, que pueden proveer estimulación cardíaca en proporción al ritmo respiratorio, utilizando sensores sencillos que suministran este dato al marcapasos, permitiendo con esto al usuario, desarrollar mayores esfuerzos ya que al aumentar su ritmo respiratorio, también aumentará su ritmo cardíaco.

A nivel experimental, pero que nos permiten vislumbrar las vastas posibilidades que se han abierto, podemos mencionar que se han desarrollado dispositivos para inducir una reacción de respuesta del sistema nervioso como consecuencia de una actividad anormal del cerebro.

Los sistemas para uso externo, utilizando microprocesadores en aplicaciones médicas, son similares a los equipos de control industriales, ofreciendo al paciente la ventaja de velocidad, eficiencia e "inteligencia", ya que las funciones vitales del paciente pueden ser monitoreadas directamente, con equipos basados en microprocesadores, colocados éstos junto a la cama del paciente, con lo cual se tiene control del ritmo cardíaco, presión de la sangre, y otras funciones vitales.

Como vemos, el campo de Aplicaciones de la Electrónica Digital es tan vasto, que sus posibilidades no han sido exploradas ni desarrolladas en toda su magnitud.

Un campo de aplicación de la electrónica digital, muy interesante, es el Control de Procesos, el cual con la introducción del microprocesador se ha desarrollado vigorosamente pues la gran flexibilidad de este dispositivo, permite su incorporación en una amplia gama de equipos, sean éstos industriales, o científicos, transformándose así en un valioso elemento de ayuda a la medicina, la química, la biología y otras áreas del conocimiento.

## EXPOSICION

---

Un sistema de suministro de gasolina típico consta de los siguientes elementos constitutivos:

- 1.- Motor eléctrico
- 2.- Bomba impulsora
- 3.- Medidor de caudal
- 4.- Indicadores de costo y cantidad

La disposición física de estos elementos se muestra en el diagrama de la página siguiente.

La función que desempeña cada uno de estos elementos se describe a continuación.

### Motor eléctrico

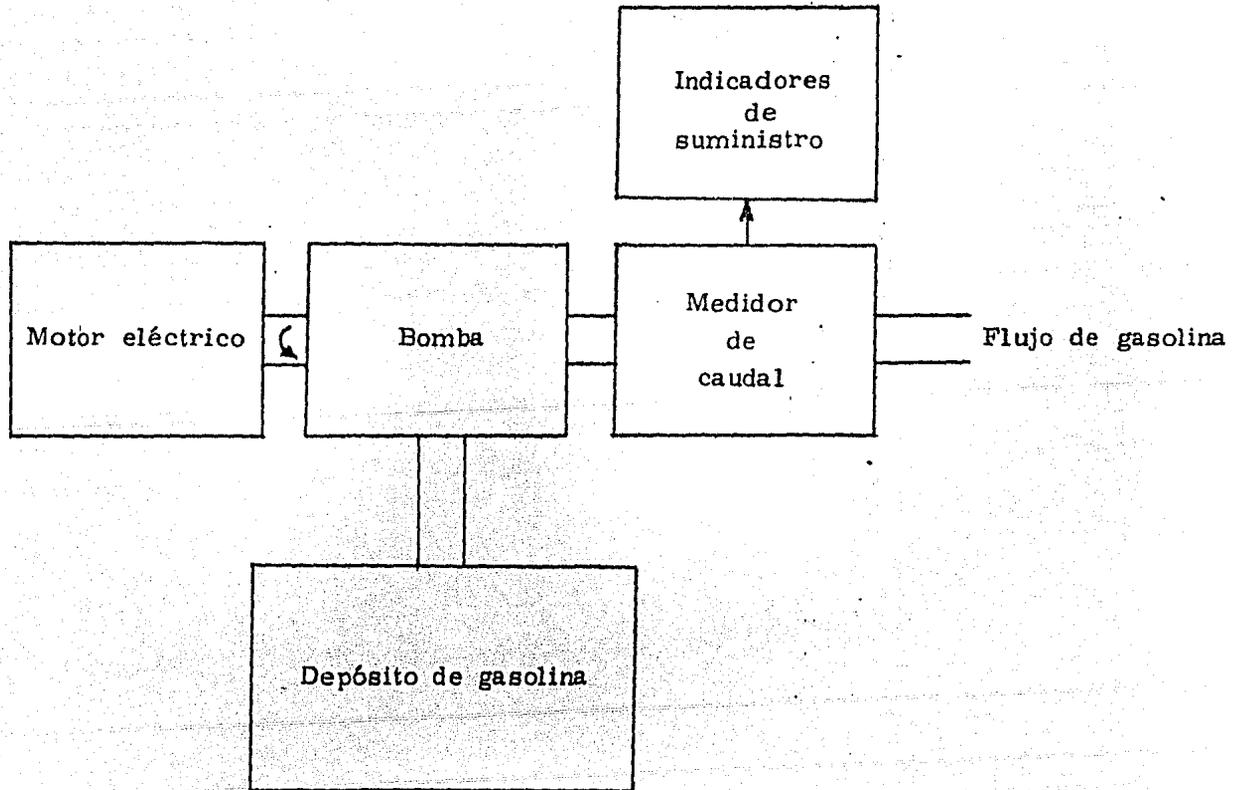
El motor eléctrico tiene la función de mover a la bomba impulsora de gasolina; este motor es de baja potencia y toma su energía de la red de distribución directamente.

### Bomba impulsora

La bomba es la encargada de succionar la gasolina que se encuentra en el depósito de la Estación de Servicio y transferirla al depósito de gasolina del cliente.

### Medidor de caudal

Este medidor tiene la función de registrar la cantidad de gasolina que es suministrada al usuario, transfiriendo esta información al sistema de despliegue de resultados.



ELEMENTOS DEL SISTEMA

## Indicadores de costo y cantidad

Los datos recogidos por el medidor de caudal son transferidos y procesados a esta unidad donde, por medios mecánicos, se convierte a su equivalente en litros y dinero.

El objetivo de este diseño es incorporar un sistema de control electrónico a este equipo, sustituyendo el sistema mecánico de cálculo por uno electrónico (digital) , que además nos permitirá controlar la operación del motor eléctrico y por lo tanto el suministro de gasolina y finalmente permitirá que el usuario defina previamente (si lo desea) el total de combustible que necesita para su vehículo.

Las ventajas que traería el integrar un sistema de control electrónico al actual sistema, serían entre otras las siguientes:

- 1.- Abatir los gastos de operación del sistema ya que un número menor de operarios podrían supervisar el funcionamiento de la Estación de Gasolina.
- 2.- Permitiría abatir costos por el mantenimiento que requiere el actual sistema mecánico de cálculo ya que las piezas mecánicas están sujetas a un desgaste al cual los componentes electrónicos no lo están.
- 3.- Facilita efectuar una supervisión centralizada del funcionamiento de todas las bombas de la Estación de Gasolina, permitiendo con esto que una sola persona maneje el dinero y ello, en un lugar adecuado.

- 4.- Al liberar a los operarios de la responsabilidad de efectuar los cobros y extender los recibos se conseguirá que éstos se encarguen de ofrecer una mejor atención al cliente, pudiéndose dedicar con una mayor eficiencia a sus demás tareas, como son revisión del aceite del motor, inflado de los neumáticos, etc., que ahora solo se realizan a petición expresa del cliente.
- 5.- Al automatizar el funcionamiento de las bombas de gasolina se tendrá la posibilidad de conectarlas a un computador central en la estación, lográndose con ello tener un control actualizado del total de gasolina disponible en el establecimiento y con ello solicitar los suministros de combustible a PEMEX de una forma más eficaz.
- 6.- Para concluir diremos que sería posible implementar en el futuro nuevos equipos a la bomba; como ejemplo mencionaremos lectoras de tarjetas de crédito y receptores de monedas, con lo cual estas bombas podrían ser operadas sin la necesidad de que estuviera presente un supervisor u operario de la estación de servicio.

Las desventajas que reportaría el automatizar una red de suministro de gasolina son las siguientes:

- 1.- Aumentar la complejidad del sistema al incluir un sistema electrónico dentro del mismo.
- 2.- Al no ser indispensables los operarios éstos podrían ser despedidos en lugar de realizar las funciones que en la actualidad desempeñan. Sin embargo este problema se podría evitar con una legislación adecuada para el funcionamiento de dichas estaciones de servicio.

## ELECCION DEL SISTEMA DE CONTROL

Un control electrónico puede ser diseñado de tres formas diferentes y son:

- 1.- Mediante lógica cableada
- 2.- Mediante el empleo de un microprocesador
- 3.- Empleando circuitos integrados especializados.

Un diseño empleando lógica cableada consiste esencialmente en la interconexión de un conjunto de componentes electrónicos discretos y circuitos integrados sencillos, de tal forma que el conjunto realice una función determinada.

Un diseño empleando un microprocesador simplifica grandemente el diseño del sistema en su configuración circuital aunque incorpora el uso de un programa, constituido por las instrucciones de dicho procesador, con lo cual, la complicación del diseño se transfiere a este campo.

Un diseño empleando un circuito integrado especializado implica la mayor simplificación circuital posible, consistiendo ésta, en un solo circuito integrado que realizará siempre las mismas funciones, sin embargo, no será posible modificar su funcionamiento en ningún aspecto, siendo además muy elevado el costo de fabricación de dicho circuito y solamente con grandes volúmenes de producción su empleo es rentable.

Para elegir el sistema que deberá ser empleado es necesario tomar en consideración los siguientes factores.

- 1.- Versatilidad
- 2.- Confiabilidad
- 3.- Costo

4.- Mantenimiento

5.- Eficiencia

### Versatilidad

Un sistema electrónico basado en un microprocesador puede ser programado para una función específica y posteriormente ser reprogramado para efectuar otras acciones o para mejorar su funcionamiento anterior. Esta versatilidad que tiene el sistema con microprocesador no se encuentra en sistemas de lógica cableada ya que, implementar una nueva función significa modificar la configuración circuital del equipo.

### Confiabilidad

Un sistema basado en la interconexión de muchos componentes electrónicos tiene una mayor probabilidad de fallar, ya que si uno de los componentes falla basta para que el dispositivo deje de trabajar adecuadamente. En un sistema con microprocesador se minimiza este riesgo, aumentando el grado de confiabilidad del sistema.

### Costo

El costo de un microprocesador se compara favorablemente con uno de lógica cableada por la sencillez, en comparación con la función equivalente implementada con lógica cableada.

### Mantenimiento

El mantenimiento del equipo es más económico, ya que existen menos componentes que en un diseño similar empleando lógica cableada.

## Eficiencia

La eficiencia de un sistema con microprocesador y la de un sistema empleando lógica cableada son, en ambos casos, muy elevados, siendo este un factor de poca ayuda para la selección del sistema a emplear.

Las ventajas que significan emplear un microprocesador se infieren inmediatamente de las consideraciones anteriormente expuestas, ya que el empleo de la lógica cableada está siendo rápidamente eliminado para dar lugar a equipos con microprocesadores, dejando su campo de aplicación reducido a funciones de apoyo.

## DESCRIPCION DEL SISTEMA

El sistema de suministro de gasolina incluye funcionalmente dos partes:

- 1.- Sistema de Suministro
- 2.- Sistema de Control

El sistema de Suministro incluye una bomba que extrae el líquido del depósito de combustible de la estación de servicio y un motor eléctrico que activa a la bomba; además se cuenta con otros elementos, como son la manguera a través de la cual se surte la gasolina.

El sistema de control incluye un medidor de caudal que registra la cantidad de fluido que lo atraviesa y un sistema de despliegue de información que indicará litros y costo; además contará con un panel de programación donde se le indicará al equipo las acciones a ejecutar.

El aspecto del equipo de suministro de gasolina podría ser el que se muestra en la página siguiente.

Los exhibidores y el teclado van conectados al procesador, el que a su vez va conectado mediante interfases adecuadas, a una motobomba y a un medidor de caudal, por lo que, los bloques básicos del equipo serán, microprocesador (incluyendo oscilador, memorias, etc.), teclado (entrada al sistema), exhibidores (salida del sistema), motobomba (elemento actuador) y medidor de caudal (transductor para controlar el flujo de gasolina).

Con esta configuración el equipo de suministro de gasolina podría funcionar de la siguiente forma:

El usuario por medio del teclado ordenará una acción al sistema.

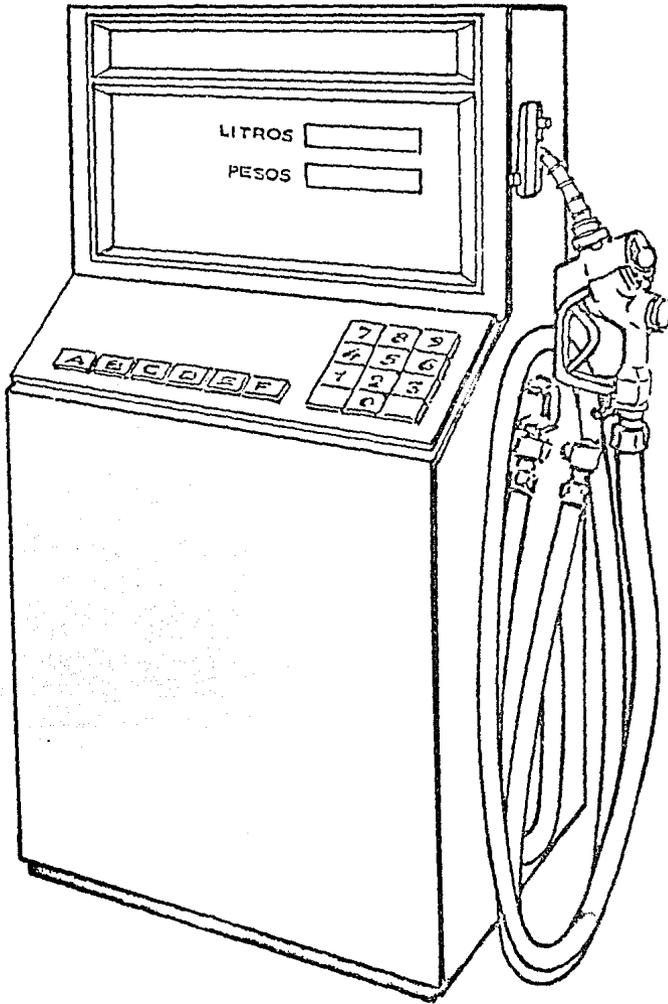
El procesador analizará esa información y la registrará.

El procesador mandará la orden de activación en el momento adecuado para que se inicie el flujo de gasolina.

El flujo de gasolina será monitoreado por medio del medidor de flujo, el cual transferirá la información al procesador.

El procesador analizará esta información y enviará los resultados al usuario.

En su caso apagará el motor eléctrico para finalizar el suministro de gasolina.



Como dijimos, el sistema de suministro de gasolina incluye una bomba, operada por un motor eléctrico acoplado a ella, que extrae el líquido de un depósito.

El equipo de Control se basa en un transductor, que monitorea el flujo del fluido y transforma este dato a un parámetro, que puede ser interpretado por el sistema de procesamiento.

A continuación veremos los diversos tipos de bombas que existen y los motores eléctricos que se pueden aplicar, seleccionando finalmente alguno de ellos, y veremos además los diversos métodos que se emplean para medir flujos, proponiendo los más adecuados para un funcionamiento eficaz y seguro del sistema.

La exposición del equipo electrónico que se empleará será tratado en el siguiente capítulo.

## BOMBAS

Una bomba es un dispositivo que absorbe energía mecánica y la transfiere a un fluido en forma de energía cinética.

Las bombas se utilizan para el transporte de fluidos, ya sean agua, aceite, petróleo o gasolina.

Las bombas debido a su principio de funcionamiento se clasifican de la siguiente manera :

- 1.- Bombas de Desplazamiento Positivo
- 2.- Bombas Rotatorias
- 3.- Bombas Centrífugas

## Bombas de Desplazamiento Positivo

Este tipo de bombas desarrollan una presión elevada gracias a la acción directa de un émbolo o pistón sobre el fluido encerrado en un cilindro, por lo que éste se ve obligado a salir con una elevada presión por las válvulas de descarga.

Estas bombas tienen como característica importante el proporcionar un volumen constante de fluido entre límites de presión muy altos, y su principal desventaja está en sus válvulas, las cuales no tienen un funcionamiento satisfactorio cuando trabajan con líquidos viscosos, y es necesario entonces instalar válvulas de seguridad y otros dispositivos para evitarles daños.

## Bombas Rotatorias

Las bombas rotatorias constituyen un grupo de máquinas que debido al movimiento rotatorio de sus piezas motrices provocan el desplazamiento del fluido que impulsan.

Las partes móviles de estas bombas giran dentro de la caja donde están construidas absorbiendo primero líquido procedente de una tubería de succión, y posteriormente forzando la salida del fluido a la presión necesaria, por la tubería de descarga.

Estas bombas que carecen de válvulas, pueden clasificarse en cinco grupos principales atendiendo a las características de sus rotores y son: las de engranes, lobulares, de aletas, de husillo y excéntricas.

Estos aparatos son lo suficientemente pequeños y mecánicamente equilibrados, para que no exijan una instalación especial. Su capacidad es casi constante dentro de los límites de velocidad del mecanismo propulsor, y para cualquier presión de impulsión.

## Bombas Centrífugas

Estas máquinas están constituidas por un impulsor o rodete, formado por una serie de aletas radiales de diversas formas y curvaturas, que giran dentro de una caja circular.

El fluido penetra por el eje de rotación y se descarga radialmente en una cámara periférica, a una elevada presión, (esta presión corresponde a la suma de la fuerza centrífuga de la rotación, más la energía cinética comunicada al fluido por las aletas).

La única parte móvil de la bomba es el rodete. Las aletas del rodete se extienden desde el centro de rotación hacia la periferia, y las bridas son los discos de refuerzo, a cada lado de las aletas, que las envuelven. Las aletas pueden ser radiales, curvadas hacia adelante, o estar curvadas al revés.

Entre los tipos de bombas centrífugas se encuentran las de Aspiración unilateral, Doble aspiración, Efecto múltiple, Sumergibles y Autocebantes.

En un sistema de suministro de gasolina real es usual utilizar bombas de tipo rotatorio, en particular del tipo de aspas o aletas, con capacidad de 56 litros/minuto, empleando una potencia de  $1/2$  H.P. (0.50 HP), o bien de una alta capacidad, suministrando 90 litros/minuto con una potencia de  $3/4$  H.P. (0.75 HP).

## MOTORES ELECTRICOS

El motor eléctrico es una máquina que convierte la electricidad en movimiento rotatorio, con el objeto de realizar un trabajo útil.

Un motor de corriente alterna es aquel que utiliza tal como lo dice su nombre, una fuente de corriente alterna para realizar su trabajo.

Debido a que la mayor parte de la energía eléctrica se recibe de distribución en forma alterna, es muy usual el empleo de este tipo de motores, pero además, esta energía es suministrada en forma monofásica, esto es, el usuario recibe solo una línea desde la red; en lugar de las tres que conforman el sistema de distribución.

Los motores eléctricos que utilizan energía alterna monofásica son de varios tipos, cada uno de ellos con características particulares, debido a que es necesario para el motor operar por lo menos con dos fases y no con una; ésto lleva a que los motores monofásicos generen una segunda fase artificialmente, que puede ser lograda de varias formas, dando origen a diversos tipos de motores; algunos de los más usuales son:

- 1.- Motor Universal
- 2.- Motor de Fase dividida
- 3.- Motor con Espira de sombra
- 4.- Motor de Arranque con capacitor

El motor universal es un tipo de motor muy utilizado debido a que puede funcionar no solo con corriente alterna sino también con corriente continua; se le encuentra en aspiradoras, licuadoras, taladros etc.

El motor de fase dividida tiene la característica de emplear un devanado especial que genera una segunda fase eléctrica y el cual se desconecta automáticamente cuando el motor alcanza una cierta velocidad.

El motor con espira de sombra es muy utilizado por ser muy simple de construcción. Desgraciadamente tiene poca potencia de arranque, lo cual limita sus aplicaciones. Se utiliza mucho en ventiladores y pequeños motores para tocadiscos y tocacintas.

En el motor de arranque con capacitor la diferencia de fases se genera empleando un capacitor, el cual se conecta en serie con el devanado de arranque del motor. Al alcanzar este motor una cierta velocidad el capacitor y el devanado de arranque se desconectan del sistema.

El empleo de un condensador tiene una serie de ventajas que incluye una alta potencia de arranque; entre otras, esta característica lo ha hecho el indicado para ser usado con las bombas de suministro de gasolina, contando con las siguientes características:

Potencia de 1/2 HP (0.50HP), 1/3 HP (0.33HP) ó 3/4 HP (0.75 HP), una Frecuencia de 60 Hz, una velocidad de 1800 RPM, además debe estar blindado para evitar explosiones.

## MEDIDORES DE FLUJO

Existen varios métodos para la determinación del caudal de un fluido. Entre los más usuales se encuentran:

- 1.- Medidores diferenciales de presión
- 2.- Medidores de desplazamiento positivo
- 3.- Medidores electromagnéticos de flujo
- 4.- Medidores ultrasónicos de flujo.

### Medidores diferenciales de presión

El principio de funcionamiento de estos medidores consiste en oponer una resistencia fija al flujo en el ducto de conducción. Esta resistencia provoca una caída de presión que varía en forma proporcional al flujo, mientras que la velocidad del flujo aumenta.

Después de atravesar la reducción de la sección, la corriente fluida recobra su sección original, y esta obstrucción puede estar diseñada de varias formas, dando origen a varios instrumentos de medición, los cuales son:

- 1.- Placas con orificio
- 2.- Tubos Venturi
- 3.- Toberas de flujo
- 4.- Tubos de flujo
- 5.- Tubos de Pitot

### Placas con orificio

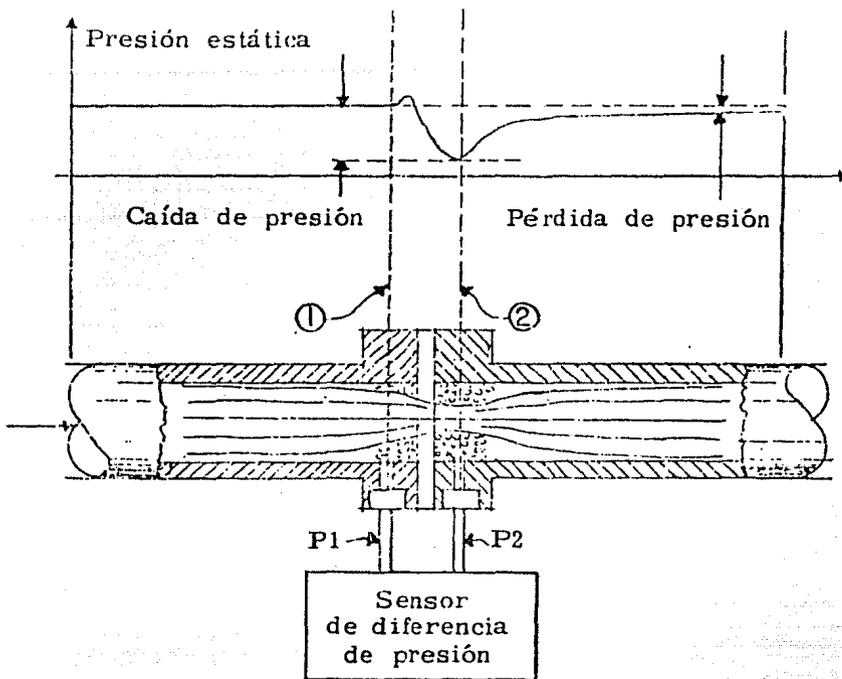
Es el dispositivo más sencillo de su tipo y el menos eficiente, debido a las pérdidas de presión que produce, las que pueden llegar a ser de un 60% a un 80%, consistiendo en un disco plano y delgado con un orificio, y se inserta entre dos bridas de la tubería.

### Tubos Venturi

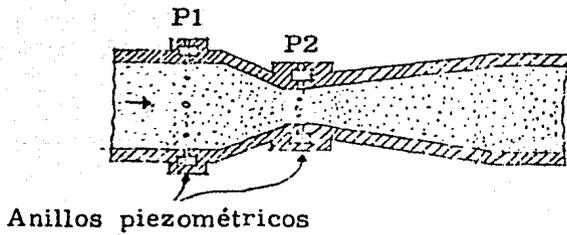
Este dispositivo se utiliza cuando el fluido a medir contiene grandes cantidades de sólidos en suspensión. Se utiliza también cuando se desea la máxima exactitud en la medición de fluidos altamente viscosos:

### Tobera de flujo

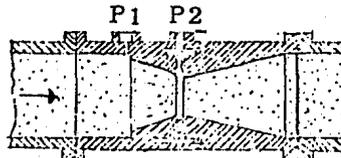
La tobera de flujo permite medidas de velocidad de flujo de 60% a 65% más elevadas que las permitidas por los tubos de diafragma, y hace que la tobera encuentre su principal aplicación cuando han de ser medidos grandes caudales en líneas de alta presión y poco diámetro. También se emplea debido a que los diafragmas requieren una cierta longitud de tubería recta, antes y después del elemento de medición. Esta longitud ha de ser tanto mayor cuanto mayor sea el diámetro del orificio del diafragma. Sin embargo, en la tobera, tenemos un menor diámetro del dispositivo, requiriéndose una menor longitud del tramo recto. Esto representa ciertas ventajas cuando la longitud de los tramos rectos es crítica.



PLACA CON ORIFICIO



TUBO DE VENTURI



TUBO DE DALL

### Tubo de flujo Dall

Este dispositivo tiene unas pérdidas de carga del orden del 5%, lo que es menor que en un tubo Venturi o en un diafragma. Este instrumento es más ligero, más barato y de menor longitud que un tubo Venturi.

### Tubo de Pitot

Este elemento es muy exacto para la medida de caudales y muy útil para la comprobación de instalaciones de medición de caudal, como sistema de calibración. Se utiliza mucho en las mediciones de caudal de tuberías de gran diámetro donde con frecuencia es el dispositivo más económico y eficaz.

Para emplear cualquiera de estos dispositivos en un sistema electrónico, es necesario emplear un transductor de presión como el fabricado por National Semiconductors series LX16XXD ó LX17XXD, que son transductores diferenciales de presión, y tienen por objeto transformar la diferencia de presión en una diferencia de voltaje y así transmitirla al sistema electrónico, teniendo que ser procesada en un convertidor analógico digital antes de poderse trabajar con ella en el sistema de microprocesador.

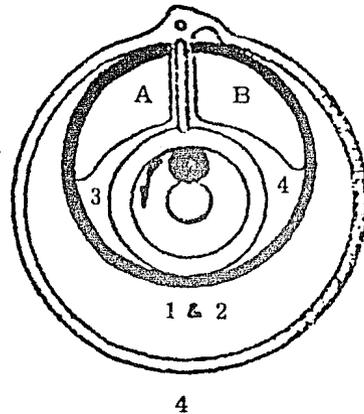
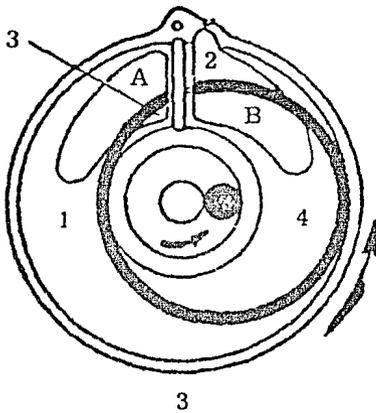
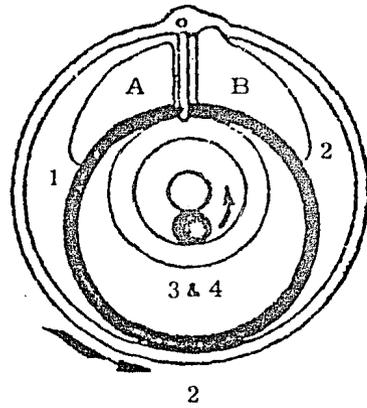
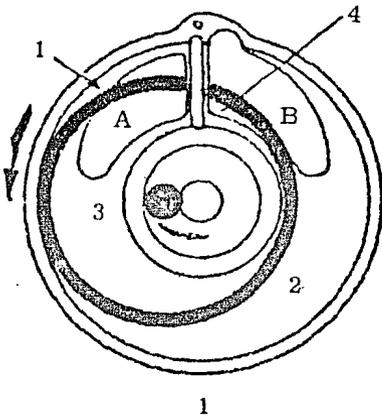
### Medidores de desplazamiento positivo

En un medidor de gasto por desplazamiento, el gasto se determina subdividiendo la corriente total del fluido en fracciones más o menos grandes de volumen conocido. La medición se efectúa entonces mediante el recuento del número de fracciones en la unidad de tiempo.

Los tipos más corrientes de medidores por desplazamiento son:

Medidor de pistón oscilante

Medidor de disco oscilante



En las posiciones 1 y 3 se recibe el fluido que entra por A

En las posiciones 2 y 4 se descarga el fluido por B

#### OPERACION DEL MEDIDOR DE PISTON OSCILANTE

### Medidor de pistón oscilante

En el medidor de pistón oscilante todo el fluido penetra por el orificio de entrada, y pasa alrededor del espacio anular, entre los anillos externos e internos hacia el orificio de descarga. Parte del fluido recorre esta distancia por el espacio existente entre el pistón y el anillo externo, y parte circula por el espacio entre la pared del pistón hueco y el anillo interior. El pistón va guiado por el eje, el cual sigue una trayectoria circular entre el anillo interno y un rodillo central. Una prolongación del eje que atraviesa la cubierta de la cámara de medida, comunica movimiento circular, el cual nos permite acoplarlo a un circuito detector.

### Medidor de disco oscilante

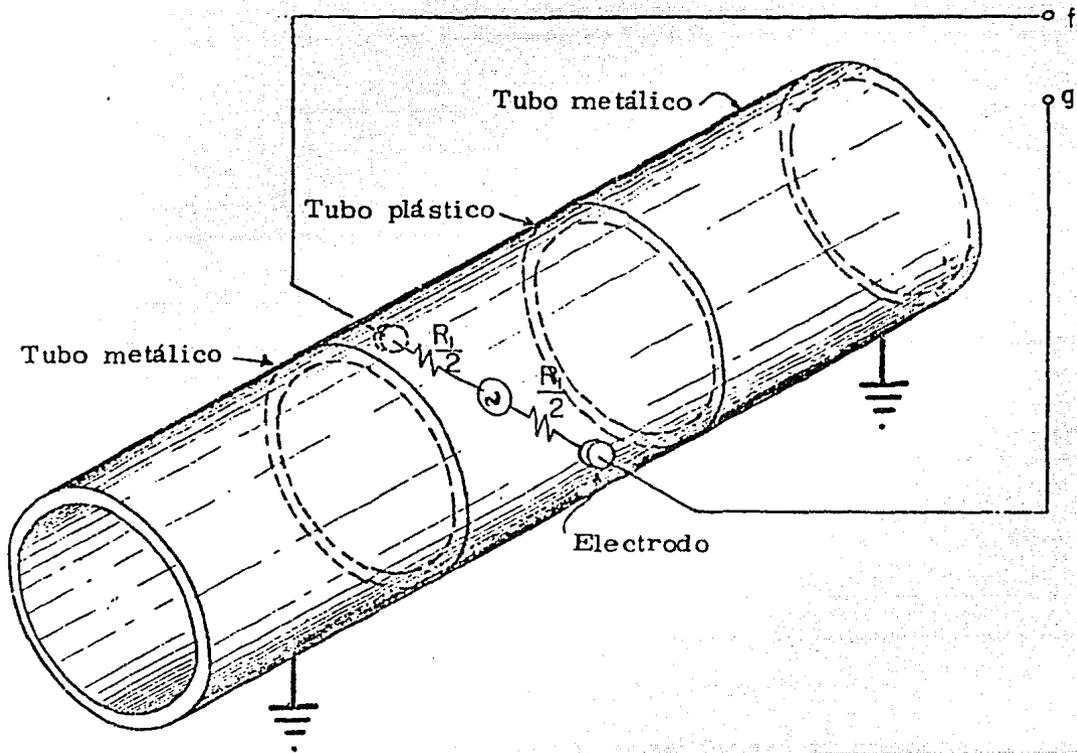
La cámara de medida de un medidor de disco oscilante posee una forma aproximadamente cilíndrica, y lleva un diafragma que se extiende radialmente, desde un lado de la pared de la cámara hasta el rodamiento esférico del centro, del mismo modo que ocurre en la cámara del contador de pistón oscilante.

Tan pronto como el fluido penetra en el espacio entre el disco y las paredes de la cámara, empuja al disco hacia adelante, imprimiéndole un movimiento de oscilación rotatoria. Este tipo de contador se utiliza mucho en las conducciones de agua de capacidad media.

### Medidores de flujo electromagnéticos

Este tipo de medidor de flujo mide el caudal de muchos líquidos y semilíquidos. No presenta obstrucción al paso del fluido y puede medirse con facilidad el flujo de líquidos, corrosivos o sulfurosos.

El medidor consiste en un cable eléctrico conectado a un transmisor de flujo y a un receptor. Para que el medidor funcione, la conductancia mínima del líquido a medir, ha de ser



MEDIDOR DE GASTO ELECTROMAGNETICO

200 micromhos para diámetros en el rango de 15.25 cm y de 500 micromhos para rangos del orden de 20 cm.

La salida del transmisor es lineal y directamente proporcional a la velocidad promedio del líquido, y por lo tanto, proporcional a su caudal y la turbulencia no afecta seriamente al transmisor, por lo que no son necesarios tramos rectos de tubería para amortiguarla, como en el caso de los medidores de obstrucción.

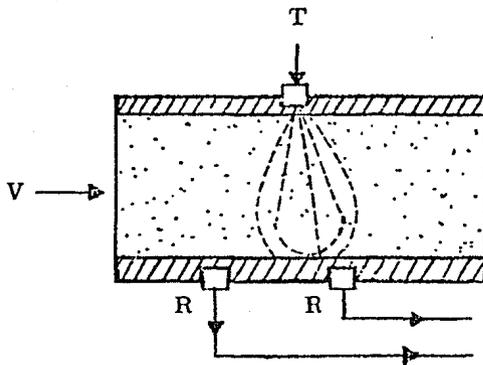
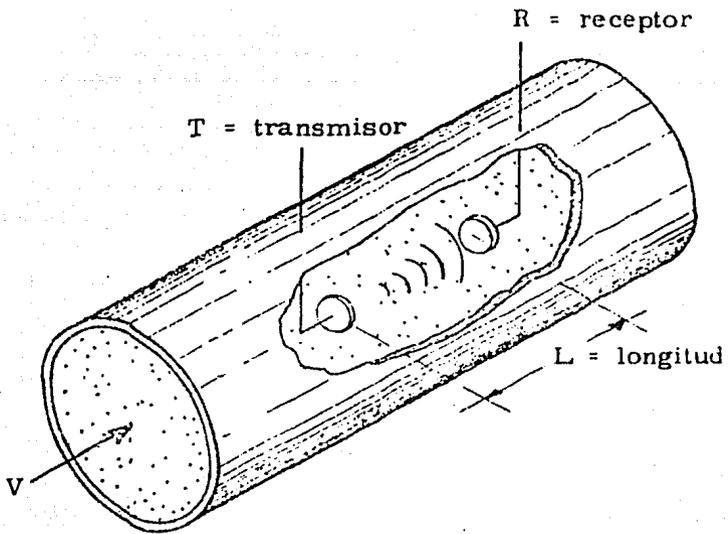
El funcionamiento del transmisor electromagnético está basado en la ley de Faraday de la inducción electromagnética, que establece que el voltaje  $E$  inducido en un conductor de longitud  $B$  que se mueve en un campo magnético  $H$ , es proporcional a la velocidad  $V$  del conductor. El voltaje se genera en una dirección perpendicular al plano, definido por la velocidad del conductor y la inducción del campo magnético.

En este medidor el líquido que está fluyendo es un conductor en movimiento que atraviesa un campo magnético, y debe imaginarse como un disco plano de líquido que se mueve entre dos electrodos, y la longitud del conductor  $D$  es la distancia entre ambos electrodos.

Estos medidores son muy exactos, teniendo una tolerancia del 1% para todo el dispositivo, incluyendo transmisor, cable, y receptor.

### Medidores de gasto ultrasónicos

Las perturbaciones de presión de pequeña magnitud se propagan, a través de un fluido, con una velocidad definida con relación al fluido igual a la velocidad del sonido. Si además el fluido tiene una velocidad, la velocidad absoluta de propagación de la perturbación es la suma algebraica de ambas. Como el gasto está relacionado con la velocidad del fluido, este efecto puede usarse para implementar un medidor de gasto.



MEDIDOR DE GASTO ULTRASONICO

Un procedimiento típico de construcción de este tipo de medidor consiste en utilizar transductores de material cristalino, que presentan un efecto piezoeléctrico. Aplicando en un transmisor hecho de este material un voltaje de alta frecuencia, y si el fluido está en contacto con el cristal, entonces la vibración se comunicará al fluido y se propagará a través de él, y el cristal del receptor estará expuesto a estas fluctuaciones de presión, respondiendo con una señal eléctrica proporcional a la presión que ha recibido.

### SELECCION DEL MEDIDOR DE FLUJO

Tomando en consideración las ventajas y desventajas que implica cada método de medición se considera aconsejable el empleo de un medidor de desplazamiento positivo, ya sea de pistón oscilante o de disco oscilante, por las siguientes razones:

- 1.- Este método es lo suficientemente preciso para ser empleado en los sistemas de Suministro de gasolina.
- 2.- Al ser un dispositivo mecánico no tiene ninguna fuente eléctrica que pudiera provocar un accidente en un fluido tan inflamable como lo es la gasolina.
- 3.- No es afectado por turbulencias, como en el caso de los medidores de obstrucción como son el Venturi y el Diafragma.
- 4.- Los transductores de presión como los que se mencionaron en el análisis anterior, son difíciles de encontrar en nuestro país y tienen un costo muy elevado, pudiéndose decir lo mismo de los receptores y transmisores ultrasónicos y los electromagnéticos.

5.- Es el único sistema que nos permite transferir la información, en forma de pulsos, al sistema de microcomputadora, sin necesidad de utilizar un convertidor analógico digital para tal fin, reduciéndose costos al evitar este elemento.

6.- Son los más indicados para un trato rudo, no siendo así, el caso de los demás medidores que pueden ser dañados fácilmente.

Un microprocesador es un circuito integrado, sumamente sofisticado, que puede ser programado para realizar un gran número de funciones ya sean sumas, restas, operaciones lógicas, etc., además de permitir la toma de decisiones en función de algún resultado o alguna condición interna.

Para poder realizar adecuadamente sus funciones es necesario que actúe junto con otros circuitos electrónicos, que lo complementan y le permitan operar adecuadamente.

En general cualquier sistema basado en un microprocesador cuenta con los siguientes elementos:

1. - Microprocesador
2. - Memoria para almacenar datos y programas
3. - Dispositivos de entrada
4. - Dispositivos de salida

Por ejemplo, una computadora digital está formada por estos elementos básicos. El microprocesador que forma la unidad central de proceso (CPU) es la parte principal de la computadora y por sus funciones se divide en dos unidades que son, la unidad aritmética y lógica y la unidad de control.

## Microprocesador

El microprocesador está constituido por dos partes principales, la unidad aritmética y lógica y la unidad de control.

La unidad aritmética y lógica es la que ejecuta el trabajo de procesamiento, o sea, es la encargada de recibir los datos y efectuar con ellos operaciones aritméticas, lógicas, de comparación, corrimientos de bits, incrementos, decrementos, etc. La unidad aritmética y lógica tiene registros para almacenar datos, siendo el más importante de ellos en el microprocesador Z80 CPU, el acumulador A y en aritmética de doble precisión, el par de registros HL.

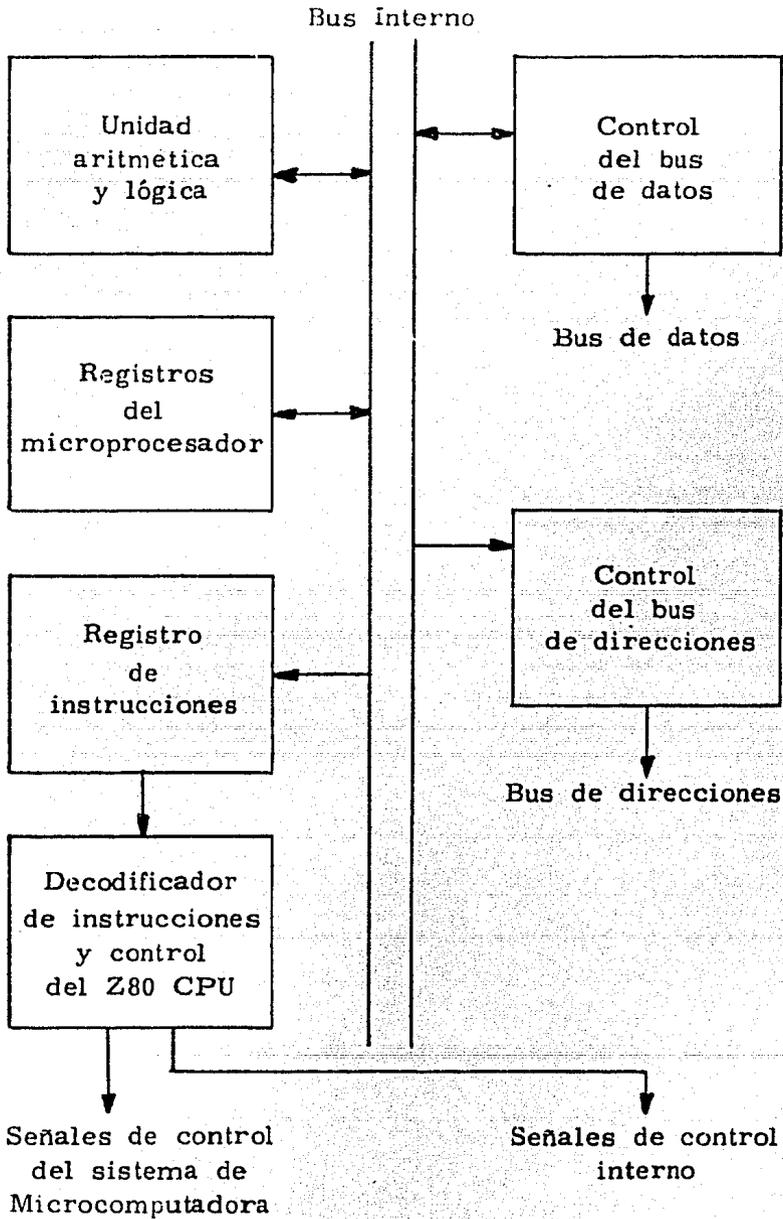
La unidad de control tiene como función el sincronizar todas las acciones que se realizan en la computadora. Esta unidad recibe las instrucciones codificadas en binario desde la memoria y decide cuándo y cómo se debe ejecutar cada instrucción del programa.

## Memoria

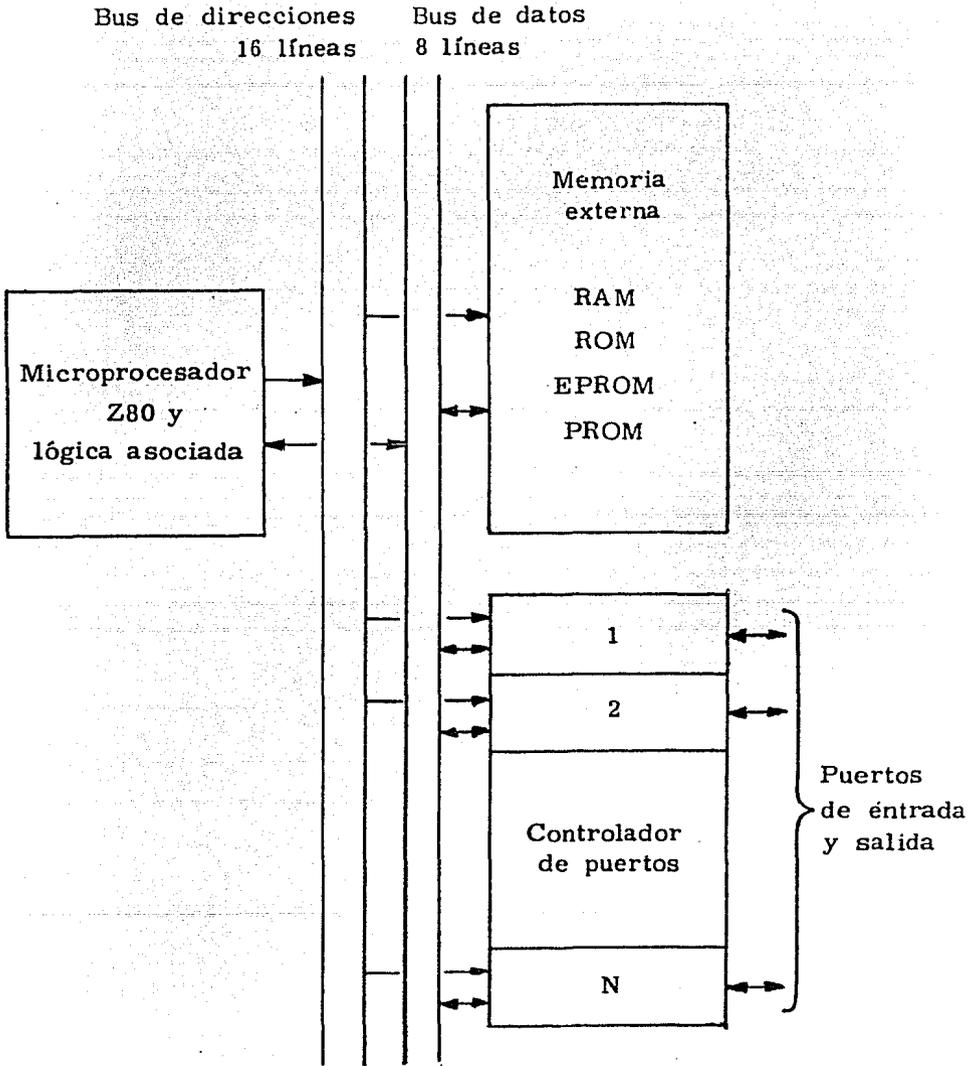
Esta unidad contiene las instrucciones y los datos que van a ser procesados en el CPU. La memoria se divide en dos áreas principales que son, memoria propia y memoria auxiliar. Las memorias propias están construidas con circuitos semiconductores, mientras que la auxiliar está formada por dispositivos de almacenamiento masivo de instrucciones y datos, como son discos, cintas magnéticas, tarjetas perforadas, etc.

## Dispositivos de entrada

Estos elementos se utilizan para transferir desde el mundo externo órdenes y datos, así como programas enteros al sistema de procesamiento.



### ARQUITECTURA DEL MICROPROCESADOR Z80 CPU



COMPONENTES DE UN SISTEMA DE MICROCOMPUTADORA

## Dispositivos de salida

Los dispositivos de salida nos permiten obtener los resultados del procesamiento informativo del sistema, y usualmente son tubos de rayos catódicos, cintas magnéticas o cintas perforadas.

## UNIDAD CENTRAL DE PROCESO

La unidad central de proceso es el cerebro del sistema de una microcomputadora. El circuito integrado Z80 CPU es un circuito programable construido con tecnología NMOS y viene encapsulado en una pastilla o circuito integrado con 40 conectores, requiriendo únicamente una fuente de alimentación de 5V de corriente directa y un reloj que le proporcione pulsos de forma cuadrada con una frecuencia en el rango de 2.5 MHz a 4.0 MHz.

El Z80 CPU incluye un juego de 158 instrucciones que le permiten cargar datos de 8 y 16 bits desde memoria y entre registros, transferir bloques de datos de una parte de la memoria a otra, realizar operaciones aritméticas y lógicas de 8 y 16 bits, realizar operaciones especiales como ajuste decimal, efectuar operaciones con bits individuales, ya sea en memoria o en registros, realizar saltos, ya sean incondicionales (tipo GO TO) o condicionales (tipo IF THEN), que permiten la toma de decisiones. Podemos además llamar subrutinas en el mismo programa, de acuerdo a ciertas condiciones que se presenten en el procesamiento y finalmente nos permite ejecutar operaciones de entrada y salida de datos ya sean individualmente o en bloque.

El Z80 CPU cuenta con 18 registros de 8 bits y con 4 registros de 16 bits. De los 18 registros de 8 bits, 14 de ellos son de propósito general, es decir, son registros que

realizan una gran cantidad de funciones, y se pueden direccionar individualmente, o en pares de registros cuando se requiere doble precisión, incluyendo además un registro con banderas para indicar ciertos resultados que permiten realizar saltos, y llamadas a Subrutinas.

El microprocesador Z80 CPU cuenta con tres buses (líneas para transferir información):

- 1.- Bus de datos
- 2.- Bus de direcciones
- 3.- Bus interno de transferencias.

#### Bus de datos

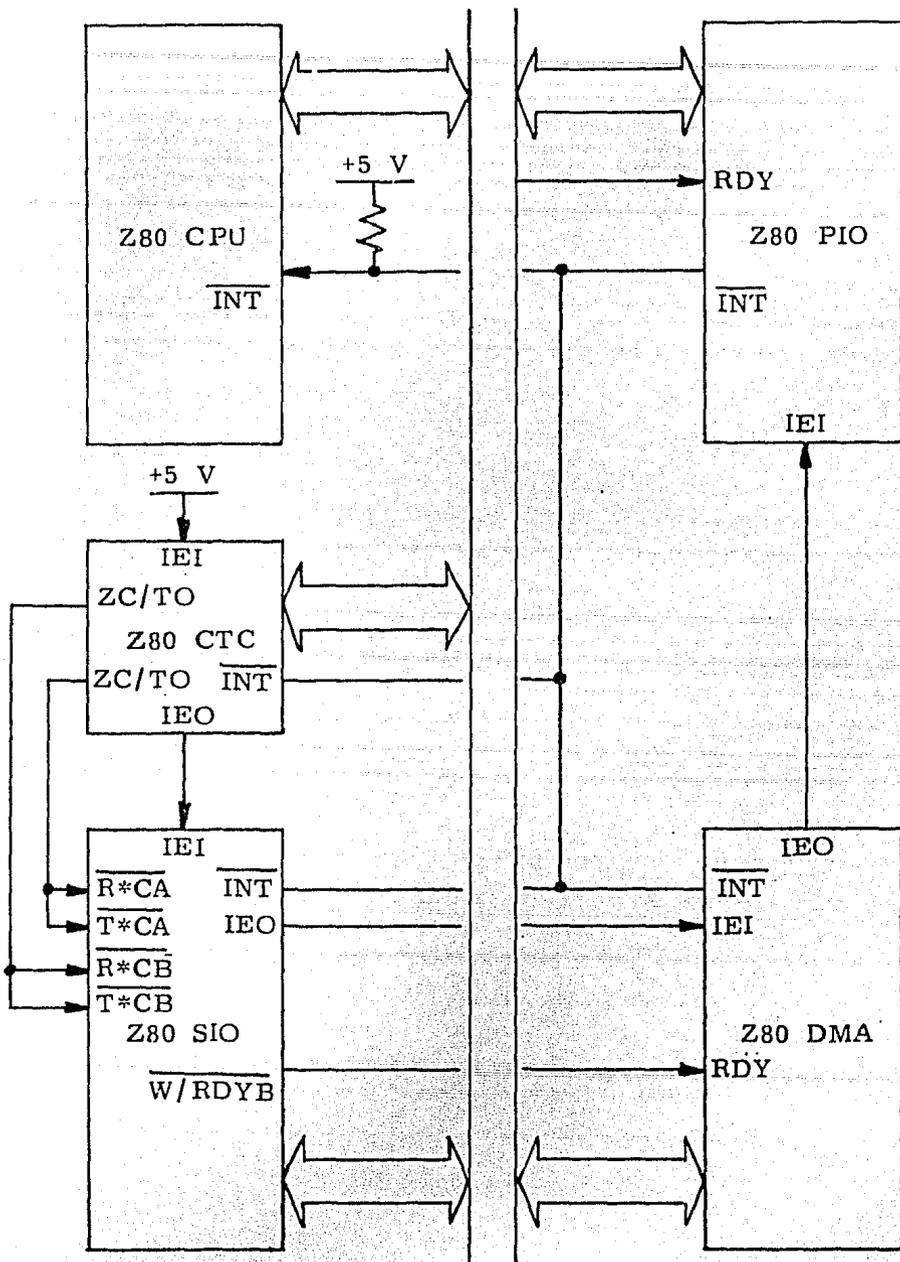
Cuenta con ocho líneas formando un bus de ocho bits. Este bus es el camino utilizado para realizar todas las transferencias de datos entre la memoria (externa al microprocesador), los puertos de entrada o salida y el microprocesador.

#### Bus de direcciones

Está constituido por dieciseis líneas, que forman un bus de 16 bits, y se utiliza para direccionar las diversas localidades de memoria, las cuales pueden ser la localidad cero, la uno, la dos, etc., hasta la 65535.

#### Bus interno de transferencias

Este bus es utilizado para las transferencias internas de información ya sean transferencias de datos entre registros, o entre registros y la unidad aritmética y lógica principalmente.



CONFIGURACION DE UN SISTEMA DE MICROCOMPUTADORA  
 CON CIRCUITOS DE LA FAMILIA Z80

El microprocesador cuenta además con 13 líneas que transmiten señales al microprocesador o señales que son generadas en él.

## REGISTROS DEL MICROPROCESADOR

El microprocesador cuenta con los siguientes registros:

- 1.- Contador de programa (PC , de 16 bits)
- 2.- Apuntador del apilador (stack) (SP, de 16 bits)
- 3.- Registros de propósito general (A, B, C, D, E, H, L, de 8 bits)
- 4.- Registros inactivos de propósito general (A', B', C', D', E', H', L', de 8 bits)
- 5.- Registros de índice (IX, IY, de 16 bits)
- 6.- Registros de banderas, activo e inactivo (F, F', de 8 bits)
- 7.- Registro para interrupciones (I, de 8 bits)
- 8.- Registro de regeneración de memorias dinámicas (R, de 8 bits).

### Contador de programa

El contador de programa tiene una longitud de 16 bits, cuya función es indicar la localidad de memoria donde está la orden que debe ser ejecutada.

## Apuntador del apilador (stack)

Este registro de 16 bits tiene la función de señalar el tope del apilador, entendiendo por apilador o (stack) una zona de memoria donde se pueden almacenar, de forma temporal, los contenidos de los registros, utilizando para ello las instrucciones PUSH para apilar, y POP para recuperar el dato.

## Registros de propósito general

Son el acumulador A y los registros B, C, D, E, H, L, que se utilizan para almacenar datos numéricos y realizar operaciones con ellos a través de la unidad aritmética y lógica, y además, recibir datos desde memoria, enviar datos a memoria, transferir datos entre ellos y formar los registros dobles BC, DE y HL.

## Registros inactivos de propósito general

Estos registros intercambian su contenido con el de los registros activos mediante las instrucciones EXX, que intercambia a B por B', C por C', D por D', E por E', H por H', L por L', y EX A, F, que intercambia acumuladores A por A' y banderas F por F'.

## Registros de índice

Estos dos registros son de 16 bits cada uno y tienen como función apuntar a una zona de memoria con la que se puede transferir datos y resultados. Esto se logra colocando a IX o IY en una posición específica y sumando o restando un número de 8 bits, con lo cual se especifica la localidad que será accesada.

## Registros de bandera activo e inactivo

Este registro de 8 bits tiene la función de almacenar algunas condiciones que se generan al realizar una operación, algunas de las cuales pueden ser empleadas para la toma de decisiones y son:

- 1.- Bandera de cero
- 2.- Bandera de signo
- 3.- Bandera de acarreo
- 4.- Bandera de substracción
- 5.- Bandera de paridad

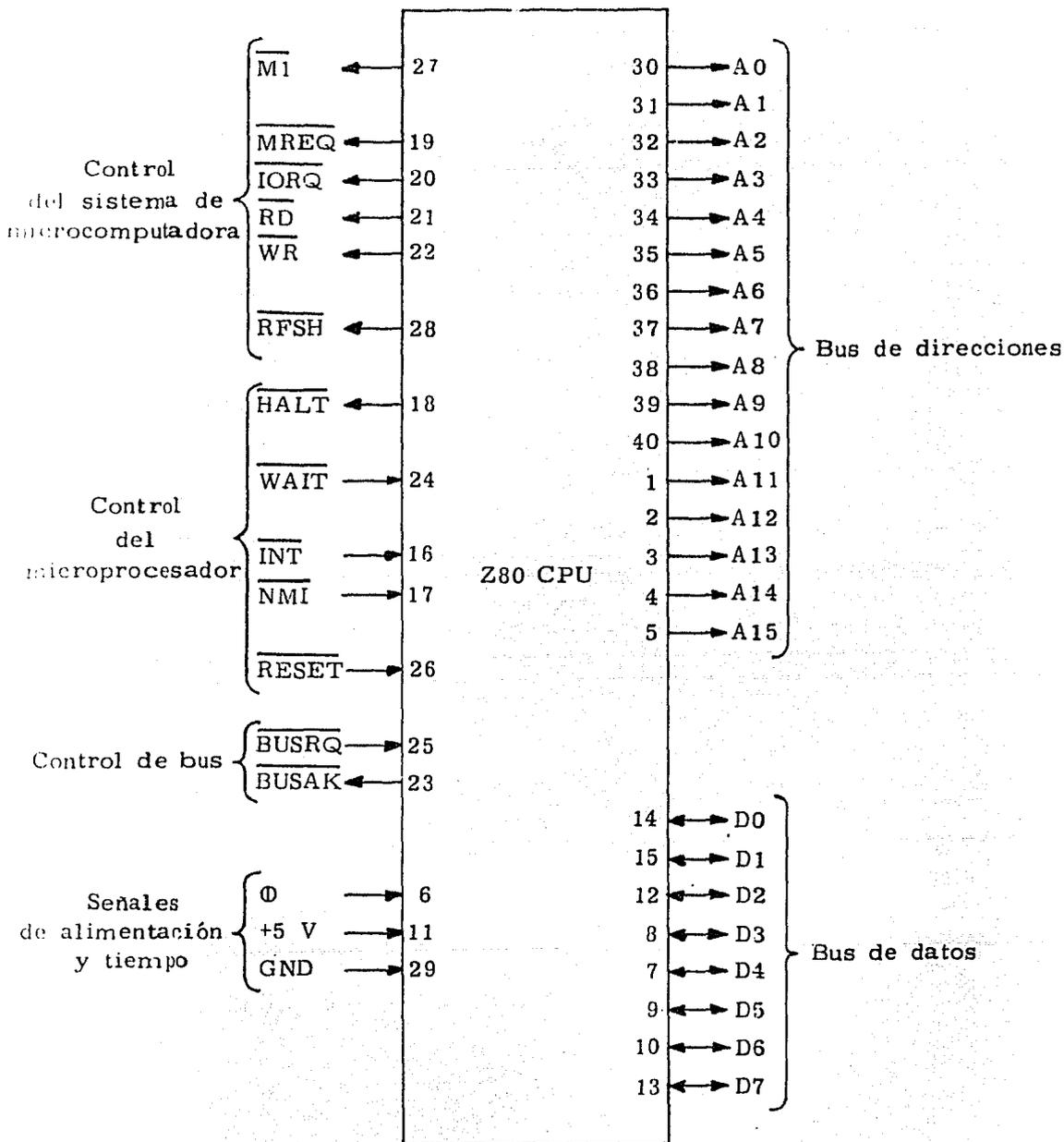
## DESCRIPCION DEL CIRCUITO

El Z80 CPU cuenta con 40 conectores que se agrupan funcionalmente en las siguientes categorías.

- 1.- Bus de direcciones
- 2.- Bus de datos
- 3.- Control del sistema
- 4.- Control de ejecución
- 5.- Control de buses
- 6.- Señales de reloj y alimentación

### Bus de direcciones

El bus de direcciones está compuesto por 16 líneas denominadas A0-A15, donde la línea de menor orden es A0. Las posiciones de estas líneas en la pastilla del Z80 CPU son, A15 posición 5, A14 posición 4, A13 posición 3, A12 posición 2, A11 posición 1, A10 posición 40, A9 posición 39, A8 posición 38, A7 posición 37, A6 posición 36, A5 posición



LINEAS DEL Z80 CPU

35, A4 posición 34, A3 posición 33, A2 posición 32, A1 posición 31 y A0 posición 30.

### Bus de datos

El bus de datos constituye el medio por el cual se intercambian datos entre el Z80 CPU y las memorias o los dispositivos de entrada y salida.

Las posiciones de estas líneas a partir de la línea D7, que es la de mayor orden o la más significativa, son D7, posición 13, D6 posición 10, D5 posición 9, D4 posición 7, D3 posición 8, D2 posición 12, D1 posición 15 y D0 posición 14.

### Control del sistema

Las señales de control del sistema se generan en el Z80 CPU y son las siguientes:

- 1.- Primer ciclo de máquina (M1)
- 2.- Solicitud de acceso a memoria (MREQ)
- 3.- Solicitud de acceso a dispositivos de entrada y salida (IORQ)
- 4.- Señal de lectura (RD)
- 5.- Señal de escritura (WR)
- 6.- Regeneración de memoria dinámica (RFSH)

### Primer ciclo de máquina (M1)

En este ciclo se obtiene el código de operación de la instrucción que se va a ejecutar. Esta señal además se utiliza durante el reconocimiento de una interrupción. Está localizada en la posición 27 del circuito integrado.

### Solicitud de acceso a memoria (MREQ)

Esta señal está localizada en la posición 19 e indica que el bus de dirección está indicando a una localidad de memoria, ya sea para escribir en ella, para leerla o para refrescarla y actúa cuando su nivel lógico es bajo.

### Solicitud de acceso a dispositivos de entrada y salida (IORQ)

Se localiza en la posición 20 del circuito y actúa cuando su nivel lógico es bajo (0 V) indicando que esta direccionando un puerto de entrada o salida con los 8 bits menos significativos del bus de direcciones, o esta reconociendo una interrupción.

### Señal de lectura (RD)

Esta señal está localizada en la posición 21 y se activa con nivel lógico bajo, indicando que la unidad central de proceso va a leer un dato, ya sea de memoria en conjunción con la señal MREQ, o de un dispositivo de entrada y salida cuando actúa junto con IORQ, o bien que la unidad central de proceso esta obteniendo el código de una instrucción.

### Señal de escritura (WR)

Está localizada en la posición 22 y trabaja con nivel lógico bajo, indicando que la unidad central de proceso va a escribir un dato en memoria o en un dispositivo de entrada y salida, actuando con MREQ o IORQ respectivamente.

### Regeneración de memoria dinámica (RFSH)

Se localiza en la posición 28 del circuito integrado, actuando con nivel lógico bajo e indicando que las siete líneas del bus de datos de más bajo orden, contienen una orden para regenerar el nivel lógico de las memorias dinámicas (éstas son memoria en las que se pierde la información que contienen si no son continuamente accesadas).

## Control de ejecución

Estas señales son cinco y solamente la primera de ellas (HALT) es generada internamente en el Z80 CPU; y son las que se mencionan a continuación:

- 1.- Señal de alto (HALT)
- 2.- Señal de espera (WAIT)
- 3.- Solicitud de interrupción (INT)
- 4.- Solicitud de interrupción prioritaria o no mascarable (NMI)
- 5.- Señal de reinicio (RESET)

### Señal de alto (HALT)

Se localiza en la posición 18, siendo una señal de salida del Z80 CPU, y actuando con nivel lógico bajo. Indica que la unidad central de proceso está esperando una solicitud de interrupción.

### Señal de espera (WAIT)

Entrada activa a nivel bajo, está localizada en la posición 18, indicando a la unidad central de proceso que la información que ha solicitado a un puerto o a una localidad de memoria no está lista todavía para serle transmitida.

Esta señal permite la sincronización de dispositivos lentos al Z80 CPU.

### Señal de interrupción (INT)

Esta señal también actúa con nivel lógico bajo y está localizada en la posición 16 del circuito integrado. Es generada por los dispositivos de entrada y salida, pero solamente si el Z80 CPU ha sido programado para aceptarla, ésta será reconocida. Al aceptarla contesta al dispositivo con la señal IORQ (solicitud de acceso al dispositivo de entrada o salida).

### Solicitud de interrupción prioritaria o no mascarable (NMI)

Localizada en la posición 17 actúa sobre el Z80 CPU con un cambio de nivel, pasando de uno lógico a cero lógico, y tiene una mayor importancia que la señal anterior, siendo atendida aunque no haya sido programada, obligando al Z80 CPU a continuar el programa a partir de una localidad de la memoria donde está localizado el programa de atención a este tipo de interrupción prioritaria.

### Señal de reinicio (RESET)

Es una entrada al Z80 CPU y actúa con nivel lógico bajo, estando localizada en la posición 26. Con esta señal se coloca el contador de programa PC en la primera instrucción del programa, esto es, se carga con cero; Deshabilita interrupciones que pudieran estar programadas en ese momento, y además coloca los registros I (interrupción vectorizada) y R (regeneración de señales dinámicas), en cero, dando lugar a que la información almacenada en memorias de ese tipo se pierda, y finalmente coloca el modo cero de interrupción.

## Control de buses

Para el control de las líneas de datos y direcciones y control del sistema (buses) se utilizan dos señales, que son:

- 1.- Solicitud de bus (BUSRQ).
- 2.- Reconocimiento de bus (BUSAK).

### Solicitud de bus (BUSRQ)

Tiene la posición 25, siendo una entrada al Z80 CPU, y se activa cuando alcanza un nivel lógico bajo. Esta señal es generada por un dispositivo externo cuando necesita tomar el control de los buses para transmitir información.

### Reconocimiento de bus (BUSAK)

Ocupa la posición 23. Es una salida al Z80 CPU y actúa con nivel lógico bajo. Esta señal es una respuesta a la anterior, y se genera cuando el Z80 CPU cede el control de los buses al dispositivo externo que lo solicitó previamente.

### Señales de reloj y alimentación

El Z80 CPU utiliza una sola fuente de alimentación que le debe proporcionar un voltaje constante de 5V a través de la línea número 11, y el nivel de tierra deberá ser conectado a la línea número 29.

La señal de reloj deberá ser conectada a la línea número 6, y esta señal debe ser compatible con los niveles TTL del sistema, o sea 5V, y con una frecuencia de hasta 4.0 MHz.

## INSTRUCCIONES DEL MICROPROCESADOR

Cada una de las instrucciones del Z80 CPU es un código que ordena al Z80 CPU la ejecución de una operación específica.

Las instrucciones del Z80 CPU están formadas por un grupo de unos y ceros que la identifican. A esta representación binaria de la instrucción se le denomina código de máquina. Sin embargo debido a la dificultad que presenta este tipo de código para el usuario, se utiliza para cada instrucción un código denominado nemónico, el que especifica el tipo de instrucción y los registros o las variables que complementan a esta instrucción.

Las instrucciones del microprocesador se agrupan por sus funciones en 11 categorías que son:

### 1.- Carga de registros de 8 bits

En esta categoría se incluyen instrucciones para cargar cualquier registro de 8 bits con el contenido de otro registro, de una constante, o bien de una localidad de memoria, y también se incluye en esta categoría el cargar datos en memoria.

### 2.- Carga de registros de 16 bits

En este grupo se incluyen las instrucciones que permiten cargar los registros pares BC, DE, HL, IX, IY, SP, ya sea con un número o con un dato de memoria.

Además se incluyen instrucciones para cargar al SP con el contenido de los registros HL, IX, e IY. Finalmente se incluyen instrucciones para almacenar y recuperar el contenido de los registros en el área del apilador (stack).

### 3.- Intercambios de registros y datos

Los intercambios de información pueden ser entre pares de registros o entre bloques de memoria. En este grupo se consideran además instrucciones para buscar un dato dentro de un bloque de ellos.

### 4.- Operaciones aritméticas y lógicas

En esta categoría se incluyen todas las operaciones aritméticas y lógicas, incrementos y decrementos, que puede realizar el microprocesador a nivel de ocho bits.

### 5.- Aritmética especial e instrucciones de control

En esta categoría se incluyen las instrucciones que nos permiten fijar modos de interrupción, permitir las o no. Además se incluyen instrucciones para realizar operaciones especiales en el registro A, como son complementarlo a uno, complementarlo a dos, y realizar directamente un ajuste decimal sobre éste. Además hay dos operaciones para controlar la bandera de acarreo.

### 6.- Aritmética de doble precisión

En este grupo se incluyen instrucciones para realizar sumas y restas entre pares de registros, y además se incluyen instrucciones para incrementar y decrementar en uno el contenido de éstos.

### 7.- Rotación y corrimientos

Aquí se incluyen las instrucciones que permiten realizar rotaciones de los bits que están almacenados en un registro o una localidad de memoria.

### 8.- Control a nivel de bit

En este grupo se incluyen las instrucciones que nos permiten conocer el estado de un bit, ponerlo en nivel lógico alto o bajo, ya sea de un registro particular o de una localidad de memoria.

### 9.- Instrucciones para transferencia de control

En esta categoría se incluyen las instrucciones que permiten realizar saltos condicionales e incondicionales dentro de un programa.

### 10.- Llamadas a subrutinas

Aquí se incluyen las instrucciones que permiten ir a una parte específica del programa y luego regresar al punto de llamada. Este tipo de llamadas puede ser incondicional o condicional. Además se incluyen las instrucciones para regresar de una interrupción o de una interrupción prioritaria.

### 11.- Operaciones de entrada y salida

Aquí se incluyen las instrucciones que nos permiten transferir información a los dispositivos externos y de éstos al microprocesador, ya sea a nivel de un dato o de todo un bloque de ellos.

A continuación se procede a efectuar un análisis de las instrucciones que contiene el microprocesador Z80 CPU.

## Instrucciones para carga de registros de 8 bits

- 1.- Cargar un registro con un número.

Formato: LD r, n

Donde r puede ser cualquiera de los registros B, C, D, E, H, L o el acumulador A y n un número en el rango 00H a FFH.

Ejemplo: LD B, 40H

cargar el registro B con el número 40H.

- 2.- Cargar un registro con el contenido de otro registro.

Formato: LD r, r'

r y r' pueden ser cualquiera de los registros A, B, C, D, E, H, L.

Ejemplo: LD B, C

cargar el contenido de registro C en el registro B.

- 3.- Cargar un registro con el contenido de una localidad de memoria.

Formato: LD r, (ss)

d es un número signado en complemento a dos; que permite apuntar 128 localidades decimales hacia atrás o 127 localidades decimales hacia adelante.

ss puede ser la memoria indicada por los registros: HL, IX e IY.

LD r, (HL)

LD r, (IX+d)

LD r, (IY+d)

Ejemplo: LD B, (HL)

cargar la memoria indicada por HL en el registro B (previamente se debe cargar un valor en HL, por ejemplo, LD H, 20H y LD L, 20H).

- 4.- Cargar el acumulador desde la memoria.

Formato: LD A, (ss)

ss puede ser el par de registros BC, DE o un número que especifique la localidad de memoria en cuestión:

LD A, (BC)

LD A, (DE)

LD A, (nn)

Ejemplo: LD A, (3030H)

cargar en el acumulador el contenido de la memoria 3030H.

#### 5.- Almacenar en memoria el contenido de un registro.

Formato: LD (ss),r

ss pueden ser únicamente HL, IX e IY. r puede ser A, B, C, D, E, H, L.

LD (HL), r

LD (IX+d), r

LD (IY+d), r

Ejemplo: LD (HL), B

almacenar B en la memoria indicada por HL; H y L deben estar cargados previamente, por ejemplo, con las instrucciones LD H,n y LD L,n.

Nota: Si se desea cargar IX o IY se deben analizar las instrucciones del grupo de carga de 16 bits.

#### 6.- Almacenar en memoria un número

Formato: LD (ss),n

n es un número en el rango 00H, FFH, donde ss puede ser HL, IX, IY.

LD (HL), n

LD (IX+d), n

LD (IY+d), n.

Ejemplo: LD (IX), 22H

almacenar en la memoria indicada por IX. El registro IX debe ser cargado previamente con LD IX,nn. (ver instrucciones del grupo de carga de 16 bits).

#### 7.- Almacenar en memoria el contenido del acumulador.

Formato: LD (ss), A

ss puede ser BC, DE o un número en el rango 0000H, FFFFH.

LD (BC), A

LD (DE), A

LD (nn), A

Ejemplo: LD (DE), A

memorizar el acumulador en la localidad especificada por DE.

8.- Cargar el registro para regenerar memoria dinámica.

Formato: LD A,R

Descargar regenerador.

9.- Descargar el registro para regenerar memoria dinámica

Formato: LD R,A

Cargar regenerador.

### Instrucciones para carga de registros de 16 bits y manejo del apilador (stack)

1.- Cargar un registro con un número.

Formato: LD rr,nn

rr puede ser BC, DE, HL, el SP o bien IX e IY.

Ejemplo: LD IX,2020H

cargar el registro de índice con el número 2020H.

2.- Cargar un registro desde la memoria.

Formato: LD rr,(nn)

rr puede ser BC, DE, HL, SP, IX e IY.

El contenido de la localidad de memoria indicada por nn pasa a los 8 bits menos significativos del registro, mientras que la localidad nn+1 pasa a la parte más significativa del registro.

Ejemplo: LD BC, (2020H)

El contenido de la memoria indicada pasa al registro C, y la siguiente localidad de memoria, que es 2021H, pasa al registro B.

3.- Memorizar el contenido de un registro par

Formato: LD (nn), rr

rr puede ser BC, DE, HL, SP, IX e IY. En esta instrucción la parte menos significativa del par de registros se deposita en la localidad nn, mientras que la parte más significativa se almacena en la localidad siguiente, nn+1.

Ejemplo: LD (2020), HL

El contenido de L se almacena en 2020H y el contenido de H se almacena en 2021.

4.- Cargar el apuntador del apilador (stack).

Formato: LD SP, dd

dd puede ser solamente HL, IX e IY. En esta instrucción el contenido del registro dd, o sea HL o IX o IY, pasa al SP.

5.- Almacenar en el apilador (stack).

Formato: PUSH qq

qq puede ser BC, DE, HL, IX, IY. También pueden serlo el acumulador A y el contenido del registro de banderas F, que forman el par AF. El valor de más alto orden se almacena en la localidad apuntada por SP-1, y el valor de más bajo orden en la localidad apuntada por SP-2. Finalmente el valor de SP queda decrementado en dos.

Ejemplo: PUSH BC

Se almacena el contenido de C en la localidad indicada por SP-2, y el contenido de B en la localidad apuntada por SP-1.

6.- Recuperar un dato almacenado en el apilador.

Formato: POP qq

qq puede ser BC, DE, HL, IX, IY y AF, y realiza la función inversa a la anterior. Esto es, el valor de la localidad apuntada por el SP+1, pasa al registro de más

Transferencias de bloques de datos entre localidades de memoria, automáticas y semiautomáticas.

5.- Transferencia semiautomática incrementando apuntadores.

Formato: LDI

Esta instrucción transfiere un dato, que está siendo indicado por el registro HL, a otra localidad de memoria, la cual está indicada por DE. Además se incrementan en uno los dos apuntadores, HL y BC. Esto es, apuntarán a la localidad de memoria inmediata superior, y finalmente se decrementa el contenido del registro BC que actúa como un contador.

6.- Transferencia automática en incremento.

Formato: LDIR

Esta instrucción es idéntica a la anterior, pero repite el procedimiento de transferencia hasta que el contador BC llegue al valor cero.

7.- Transferencia semiautomática decrementando apuntadores.

Formato: LDD

Esta instrucción transfiere el contenido del registro HL al BC, como ocurre en las instrucciones anteriores, sólo que éstos en lugar de ser incrementados son decrementados, esto es, apuntarán a la localidad de memoria inmediata inferior.

El contador BC es igualmente decrementado en una unidad.

8.- Transferencia automática decrementando apuntadores.

Formato: LDDR

Esta instrucción realiza la misma función que la anterior (LDD), solamente que continúa transfiriendo datos hasta que el registro BC, que actúa como contador, llegue a cero.

Instrucciones para la búsqueda de un dato dentro de un bloque de memoria.

9.- Búsqueda semiautomática incrementando apuntador.

Formato: CPI

Esta instrucción compara el contenido del acumulador con el contenido de la localidad de memoria, indicada por HL; posteriormente incrementa el apuntador HL y decrementa en uno al registro BC el cual actúa como un contador.

Si al realizar la comparación ambos datos son iguales, la bandera de cero (Z) será fijada a uno.

10.- Búsqueda automática incrementando apuntador.

Formato: CPR

Esta instrucción realiza la misma función que la anterior pero repite la acción tantas veces como el contador lo indique, o bien, si al comparar el acumulador con la memoria apuntada por HL, resultan iguales, también se termina la operación, ya que el dato buscado ha sido encontrado.

11.- Búsqueda semiautomática decrementando apuntador.

Formato: CPD

Esta instrucción compara el contenido del acumulador con el contenido de la memoria, indicada por HL, decrementando a este apuntador, y finalmente decrementa al contador BC.

Si al realizar la comparación ambos datos son iguales, la bandera de cero (Z) será fijada a uno.

12.- Búsqueda automática decrementando apuntador.

Formato: CPDR

Esta instrucción realiza la misma función que la anterior, sólo que continúa comparando datos hasta que el contador BC llegue a cero, o bien que se encuentre el dato buscado.

## Operaciones aritméticas y lógicas

### 1.- Suma simple de ocho bits.

Formato: ADD s

En esta instrucción se suma al acumulador el contenido de s.  $A = A + s$

s puede ser:

- a) Un registro: ADD r, ADD H, etc.
- b) Un número: ADD n, ADD FOH, etc.
- c) El contenido de una localidad de memoria indicada por:

Apuntador HL : ADD (HL)  
Apuntador IX : ADD (IX+d)  
Apuntador IY : ADD (IY+d)

### 2.- Suma incluyendo acarreo anterior (bandera CY).

Formato: ADC s

Con esta instrucción se suma al acumulador el valor de s y el valor de la bandera de acarreo (CY).  
 $A = A + s + CY$

s puede ser:

- a) Un registro : ADC r , ADC C.
- b) Un número : ADC n , ADC 20H.
- c) El contenido de una localidad de memoria indicada por:

Apuntador HL : ADC (HL)  
Apuntador IX : ADC (IX)  
Apuntador IY : ADC (IY)

### 3.- Resta simple de ocho bits.

Formato: SUB s

Restar al acumulador el contenido de s.

$$A = A - s.$$

s puede ser:

- a) Un registro : SUB r , SUB B.
- b) Un número : SUB n , SUB 04H.
- c) El contenido de una localidad de memoria indicada por:

Apuntador HL : SUB (HL)  
Apuntador IX : SUB (IX+d)  
Apuntador IY : SUB (IY+d)

#### 4.- Resta incluyendo acarreo anterior (bandera CY).

Formato: SBC s

Esta operación es igual a la anterior, pero el contenido de la bandera de acarreo es también restada al acumulador.  $A = A - s - CY$ .

s puede ser:

- a) Un registro: SBC r, SBC H.
- b) Un número: SBC n, SBC 20H.
- c) El contenido de una localidad de memoria indicada por:

Apuntador HL : SBC (HL)  
Apuntador IX : SBC (IX+d)  
Apuntador IY : SBC (IY+d)

#### 5.- Operación lógica AND.

Formato: AND s

El operador lógico AND realiza un análisis, bit por bit, entre el acumulador y s, y su función se define:

0 AND 0 = 0  
0 AND 1 = 0  
1 AND 0 = 0  
1 AND 1 = 1

s, como en los casos anteriores, puede ser:

- a) Un registro: AND B.
- b) Un número: AND 20H.
- c) El contenido de una localidad de memoria indicada por:

Apuntador HL : AND (HL)  
Apuntador IX : AND (IX+d)  
Apuntador IY : AND (IY+d)

## 6.- Operación lógica OR.

Formato: OR s

El operador lógico OR realiza un análisis, bit por bit, entre el acumulador y s, dejando el resultado en el acumulador. El operador OR realiza la siguiente función:

0 OR 0 = 0  
0 OR 1 = 1  
1 OR 0 = 1  
1 OR 1 = 1

s puede ser:

- a) Un registro: OR D
- b) Un número: OR 20H
- c) El contenido de una localidad de memoria indicada por:

Apuntador HL : OR (HL)  
Apuntador IX : OR (IX+d)  
Apuntador IY : OR (IY+d)

## 7.- Operación lógica XOR.

Formato: XOR s

El operador lógico XOR realiza también un análisis, bit por bit, entre el acumulador y s, dejando el resultado final en el acumulador. El operador XOR realiza la siguiente función:

0 XOR 0 = 0  
0 XOR 1 = 1  
1 XOR 0 = 1  
1 XOR 1 = 0

s puede ser:

- a) Un registro: XOR D
- b) Un número: XOR 20H
- c) El contenido de una localidad de memoria indicada por:

Apuntador HL : XOR (HL)  
Apuntador IX : XOR (IX+d)  
Apuntador IY : XOR (IY+d)

## 8.- Operación de comparación.

Formato: CP s

En esta operación se compara el contenido del acumulador con el contenido de s. La operación de comparación realizada es A - s, sin embargo el acumulador no modifica su valor, (tampoco s), y lo único afectado es el registro de banderas.

s puede ser:

- a) Un registro: CP B
- b) Un número: CP 20H
- c) El contenido de una localidad de memoria indicada por:

Apuntador HL : CP (HL)  
Apuntador IX : CP (IX+d)  
Apuntador IY : CP (IY+d)

## 9.- Operación de incremento.

Formato: INC d

En esta operación el valor de d es incrementado en una unidad.

d puede ser:

- a) Un registro: (se suma uno al contenido del registro).
- b) Una localidad de memoria indicada por:

Apuntador HL : INC (HL)  
Apuntador IX : INC (IX+d)  
Apuntador IY : INC (IY+d)

(al contenido de la memoria se le suma 1).

## 10.- Operación decremento.

Formato: DEC d

En esta operación el valor de d es decrementado en una unidad.

d puede ser:

- a) Un registro: (se resta 1 al contenido del registro).
- b) Una localidad de memoria indicada por:

Apuntador HL : DEC (HL)  
Apuntador IX : DEC (IX+d)  
Apuntador IY : DEC (IY+d)

(al contenido de la memoria se le resta 1).

## Aritmética especial e instrucciones de control

### 1.- Ajuste decimal

Formato : DAA

Esta instrucción, que actúa sobre el acumulador, transforma el resultado de una suma o resta a código BCD, esto es, transforma un número de hexadecimal a decimal.

### 2.- Complementar acumulador

Formato : CPL

Esta instrucción actúa sobre el acumulador e invierte el valor de éste, es decir, los unos pasan a cero y viceversa.

### 3.- Complemento a dos del acumulador

Formato : NEG

Esta instrucción modifica el valor del acumulador colocando en su lugar el valor de su complemento más uno.

### 4.- Complementar bandera de acarreo CY

Formato : CCF

Esta instrucción actúa sobre la bandera de acarreo, y su función es invertir su estado, por consiguiente, si su valor es cero pasa a uno y viceversa.

### 5.- Fijación de bandera de acarreo CY

Formato : SCF

Esta instrucción coloca a la bandera de acarreo en estado lógico alto, esto es, en uno.

## 6.- Operación blanca.

Formato: NOP

Esta instrucción no realiza ninguna función. Es utilizada para dejar un área de programa en blanco, o es generada automáticamente por el Z80 CPU durante la ejecución de una instrucción HALT.

## 7.- Alto.

Formato: HALT

Esta instrucción impide que el programa siga siendo ejecutado; esto se logra realizando operaciones blancas NOP, sin que se incremente el valor del contador de programa.

## 8.- Desacoplar interrupciones.

Formato: DI

Esta instrucción fija el biestable IFF en cero, que impide que el CPU acepte interrupciones de dispositivos periféricos.

## 9.- Acoplar interrupciones.

Formato EI

Esta instrucción fija el biestable IFF en 1, permitiendo que los dispositivos periféricos interrumpen al Z80 CPU.

## 10.- Fijar modo de interrupción cero.

Formato: IM 0

Cuando se fija el modo de interrupción cero, y el biestable IFF 1 se ha fijado en nivel lógico alto (por medio de la instrucción EI); las solicitudes de interrupción generadas por un dispositivo periférico son reconocidas por el microprocesador; el dispositivo periférico que solicita la interrupción deberá cargar en el bus de datos del microprocesador el código de una instrucción de reinicio en página cero ya sea 1.- RST 00H, 2.- RST 08H, 3.- RST 10H, 4.- RST 18H, 5.- RST 20H, 6.- RST 28H.

RST 30H, RST 38H, localidad a partir de la cual se ejecutará el programa de servicio a interrupción, almacenándose previamente el contenido del contador de programa en el apilador (stack) y recuperándose éste al final de la rutina de servicio a interrupción, por medio de la instrucción RETI.

#### 11.- Fijar modo de interrupción uno.

Formato: IM 1

Cuando se emplea este modo de interrupción el microprocesador responde a una solicitud de interrupción en la línea INT ejecutando la instrucción RST 38 que corresponde a la dirección hexadecimal 0038H; el contenido del contador de programa PC pasa al apilador (stack); en este tipo de interrupción el dispositivo que interrumpe no requiere de circuitos especiales para generar la instrucción RST 07. Sin embargo este modo de interrupción solamente permite un nivel de interrupción, en comparación con los ocho niveles del modo de interrupción cero.

#### 12.- Fijar modo de interrupción dos.

Formato: IM 2

Este es el modo de interrupción más poderoso con que cuenta el microprocesador Z80, ya que permite realizar 128 niveles de interrupción, realizando una llamada indirecta a una localidad de memoria. Con este modo de interrupción se forma una dirección de memoria de 16 bits en donde los ocho bits de más alto orden se encuentran en el registro I del microprocesador y los ocho bits de más bajo orden son proporcionados por el dispositivo que interrumpe; esta dirección de memoria de 16 bits indica la posición en la memoria a partir de la cual se encuentra la rutina de servicio a interrupción, esto es, se realiza un salto a la localidad de memoria especificada en la localidad de memoria indicada por I y por el dispositivo periférico.

## Aritmética de doble precisión

### 1.- Suma de registros de 16 bits

Formato: ADD HL, ss

ss puede ser los registros BC, DE, HL, SP.  
El resultado de la suma queda en el registro HL.

$$HL = HL + ss$$

### 2.- Suma de registros de 16 bits, incluyendo acarreo

Formato: ADC HL, ss

ss puede ser los registros BC, DE, HL, SP.  
El resultado de la suma de HL, ss y CY queda en HL.

$$HL = HL + ss + CY$$

### 3.- Suma especial sobre el registro indexado IX

Formato: ADD IX, pp

pp puede ser los registros BC, DE, IX, SP.  
No pueden ser HL ni IY.

$$IX = IX + pp$$

### 4.- Suma especial sobre el registro indexado IY

Formato: ADD IY, rr

rr puede ser los registros BC, DE, IX y SP.  
No pueden ser HL ni IX.

$$IY = IY + rr$$

5.- Resta de registros de 16 bits, incluyendo acarreo.

Formato : SBC HL, ss

ss puede ser BC, DE, HL, SP.

Se resta de HL el valor de ss y el valor del acarreo.

$HL = HL - ss - CY$

6.- Incrementar registros de 16 bits

Formato : INC ss

ss puede ser BC, DE, HL, SP, y también IX e IY.

La función de esta instrucción es incrementar en 1 el valor del registro.

7.- Decrementar registros de 16 bits

Formato : DEC ss

ss puede ser BC, DE, HL, SP, además IX e IY.

La función de esta instrucción es decrementar en 1 el valor del registro de 16 bits en cuestión.

### Rotación y corrimientos

1.- Rotación a la izquierda sobre el acumulador

Formato : RLCA

Se realiza una rotación de todos los bits del acumulador una localidad a la izquierda, llevando el bit de más alto orden a la bandera CY.

2.- Rotación a la izquierda

Formato : RLC s

Se realiza una rotación de todos los bits de s una localidad a la izquierda, llevando el bit de más alto orden a la bandera CY.

la bandera de acarreo en dicha rotación. El bit de más alto orden del acumulador, pasa a la bandera de acarreo, y el contenido de ésta, al bit de más bajo orden del acumulador.

- 6.- Rotación a la izquierda incluyendo a la bandera de acarreo.

Formato: RL s

Se realiza una rotación de todos los bits de s una posición a la izquierda, incluyendo a la bandera de acarreo en dicha rotación, pasando el bit de más alto orden de s a la bandera de acarreo y el contenido de ésta, al bit de más bajo orden de s.

s puede ser:

- a) Un registro
- b) El contenido de una localidad de memoria apuntada por (HL), (IX+d), (IY+d).

- 7.- Rotación a la derecha sobre el acumulador incluyendo la bandera de acarreo.

Formato: RRA

En esta instrucción se realiza una rotación de todos los bits del acumulador una posición a la derecha, incluyendo en este corrimiento a la bandera de acarreo, pasando el bit de más bajo orden a la bandera de acarreo y el contenido de ésta al bit de más alto orden del acumulador.

- 8.- Rotación a la derecha, incluyendo la bandera de acarreo

Formato: RR s

En esta instrucción se realiza lo mismo que en la anterior, pero no está restringida al acumulador.

s puede ser un registro o bien el contenido de una localidad de memoria apuntada, ya sea (HL), (IX+d) o (IY+d)

la bandera de acarreo en dicha rotación. El bit de más alto orden del acumulador, pasa a la bandera de acarreo, y el contenido de ésta, al bit de más bajo orden del acumulador.

- 6.- Rotación a la izquierda incluyendo a la bandera de acarreo.

Formato : RL s

Se realiza una rotación de todos los bits de s una posición a la izquierda, incluyendo a la bandera de acarreo en dicha rotación, pasando el bit de más alto orden de s a la bandera de acarreo y el contenido de ésta, al bit de más bajo orden de s.

s puede ser:

a) Un registro

b) El contenido de una localidad de memoria apuntada por (HL), (IX+d), (IY+d).

- 7.- Rotación a la derecha sobre el acumulador incluyendo la bandera de acarreo.

Formato : RRA

En esta instrucción se realiza una rotación de todos los bits del acumulador una posición a la derecha, incluyendo en este corrimiento a la bandera de acarreo, pasando el bit de más bajo orden a la bandera de acarreo y el contenido de ésta al bit de más alto orden del acumulador.

- 8.- Rotación a la derecha, incluyendo la bandera de acarreo

Formato : RR s

En esta instrucción se realiza lo mismo que en la anterior, pero no está restringida al acumulador.

s puede ser un registro o bien el contenido de una localidad de memoria apuntada, ya sea (HL), (IX+d) o (IY+d)

## 9.- Corrimiento aritmético a la izquierda

Formato: SLA s

En esta instrucción todos los bits de s se desplazan una posición a la izquierda. El último de ellos pasa a la bandera de carry (el de más alto orden) y en el bit de más bajo orden se coloca un cero.

s puede ser:

- a) Un registro
- b) Una localidad de memoria apuntada por:

HL; SLA (HL)  
IX; SLA (IX+d)  
IY; SLA (IY+d)

## 10.- Corrimiento aritmético a la derecha

Formato: SRA s

En esta instrucción todos los bits son desplazados una posición a la derecha. El bit de más bajo orden pasa a la bandera de acarreo y en la posición de más alto orden queda el mismo valor que se tenía antes de realizar la operación.

s puede ser:

- a) Un registro
- b) Una localidad de memoria apuntada por:

HL; SRA (HL)  
IX; SRA (IX+d)  
IY; SRA (IY+d)

## 11.- Corrimiento lógico a la derecha

Formato: SRL

En esta instrucción todos los bits de  $s$  son desplazados una posición a la derecha, y el bit de más alto orden quedará automáticamente con un cero mientras que el bit de más bajo orden pasará a la bandera de acarreo.

$s$  puede ser:

a) Un registro

b) Una localidad de memoria apuntada por:

HL; SRL (HL)

IX; SRL (IX+d)

IY; SRL (IY+d)

12.- Rotación de dígitos a la izquierda entre (HL) y A.

Formato: RLD

En esta instrucción los cuatro bits de más bajo orden del acumulador pasan a las localidades de más bajo orden de la memoria, apuntada por HL. Los 4 bits de más bajo orden de (HL) son colocados en las posiciones de más alto orden y finalmente los 4 bits de más alto orden de (HL), son pasados a la posición de más bajo orden del acumulador A.

13.- Rotación de dígitos a la derecha entre (HL) y A.

Formato: RRD

En esta instrucción los cuatro bits de más bajo orden del acumulador pasan a las localidades de más alto orden de la memoria, apuntada por HL. Los 4 bits de más alto orden de (HL) son colocados en las posiciones de más bajo orden del mismo (HL), y finalmente los 4 bits de más bajo orden de (HL), son pasados a la posición de más alto orden del acumulador A.

## Control a nivel de bit

### 1.- Probar un bit

Formato: BIT b, s

Cuando se realiza esta función el valor del bit solicitado pasa a la bandera de cero(Z), pero con su valor invertido, esto es, si su valor es "1" pasará como "0".

b deberá ser un número de 0 a 7 correspondiente al bit que debe ser probado.

s puede ser un registro o una localidad de memoria apuntada ya sea por (HL) o un registro de índice.

### 2.- Fijar un bit de un registro o de una localidad de memoria.

Formato: SET b, s

Esta instrucción tiene por objeto colocar un "1" en el bit indicado, ya sea de un registro o de una localidad de memoria.

b deberá ser un número en el rango 0 a 7.

s puede ser un registro o una localidad de memoria apuntada por (HL) o un registro de índice.

### 3.- Reponer un bit a cero

Formato: RES b, s

Esta instrucción tiene por objeto colocar un "0" en el bit indicado, ya sea de un registro o de una localidad de memoria.

b deberá indicar el bit en cuestión, su valor puede ser de 0 a 7.

s puede ser un registro o una localidad de memoria apuntada por (HL) o un registro de índice.

## Instrucciones para transferencia de control

### 1.- Salto incondicional

Formato: JP nn

En esta instrucción se carga al contador de programa con el número de instrucción nn, esto es, la siguiente instrucción que realizará el Z80 CPU no será la que esté a continuación de esta instrucción, sino a otra, la cual será la indicada por nn.

### 2.- Salto condicional

Formato: JP cc, nn

Esta instrucción realiza un salto si se cumple una condición cc.

La condición cc está relacionada con el estado del registro de banderas:

Las condiciones CC pueden ser:

NZ tener "0" en bandera de cero  
Z tener "1" en bandera de cero  
NC tener "0" en bandera de acarreo  
C tener "1" en bandera de acarreo  
PO tener "1" en bandera de paridad impar  
PE tener "1" en bandera de paridad par  
P tener "1" en bandera de signo positivo  
M tener "1" en bandera de signo negativo

### 3.- Salto relativo incondicional

Formato: JR e

Este salto se realiza a una localidad del programa próxima a la instrucción, y puede ser un salto 129 localidades adelante, o 126 localidades atrás.

Tiene la ventaja de ser más rápido que el salto incondicional, pero está limitado a saltos pequeños dentro del programa.

En esta instrucción se le suma el número e (-126, 129) al contenido del contador de programa.

#### 4.- Saltos relativos condicionales

Formato: JR gg, e

Estos saltos se realizan también en el rango (-126, 129) solamente, y son más rápidos que los saltos condicionales, pero solamente pueden realizarse en las siguientes condiciones:

NZ tener "0" en bandera de cero  
Z tener "1" en bandera de cero  
NC tener "0" en bandera de acarreo  
C tener "1" en bandera de acarreo

#### 5.- Saltar y decrementar un contador hasta cero

Formato: DJNZ, e

Este salto relativo se realiza n veces hasta que el contador B, previamente cargado con n llega a cero y entonces se prosigue con el programa.

#### 6.- Saltos indirectos

Formato: JP (ss)

(ss) puede ser (HL), (IX), e (IY).

En esta instrucción se carga al contador de programa con el contenido de una localidad apuntada por HL, IX, o IY; los registros de índice no permiten en este caso un desplazamiento.

HL; JP (HL)  
IX; JP (IX)  
IY; JP (IY)

## Llamadas a subrutinas

### 1.- Llamada a subrutina

Formato: CALL nn

Una subrutina es un segmento de programa que se utiliza varias veces.

Al utilizar la instrucción CALL nn, se hace un salto a la localidad nn, donde está almacenada la subrutina; al terminar ésta se regresa al punto de llamada.

Al utilizar CALL nn, se almacena el contenido del contador de programa en el apilador de la siguiente forma:

El byte de más alto orden del contador de programa se almacena en la localidad apuntada por el SP - 1 y el byte de más alto orden en la localidad apuntada por el SP-2; por supuesto el apuntador del apilador SP quedará decrementado en dos y finalmente el contador de programa quedará cargado con el valor nn.

### 2.- Regreso de subrutina

Formato: RET

Esta instrucción se utiliza al final de una subrutina, y sirve para regresar al programa principal. El contenido del apilador, indicado por su apuntador SP, pasará a las localidades de más bajo orden del contador de programa, y los ocho bits almacenados en la localidad apuntada por SP+1, pasarán a las localidades de más alto orden del contador del programa, siguiendo así la ejecución del programa principal.

### 3.- Llamada condicional a subrutina

Formato: CALL cc,nn

Esta llamada a subrutina se basa en una condición, y si esta condición se cumple, se llama a la subrutina.

La condición cc puede ser:

NZ tener "0" en bandera de cero  
Z tener "1" en bandera de cero  
NC tener "0" en bandera de acarreo  
C tener "1" en bandera de acarreo  
PE tener "1" en bandera de paridad par  
PO tener "1" en bandera de paridad impar  
P tener "1" en bandera de signo positivo  
M tener "1" en bandera de signo negativo

#### 4.- Regreso condicional de subrutina

Formato: RET cc

Esta instrucción se utiliza para regresar al programa principal si la condición cc se cumple, y estas condiciones son las mismas que para la instrucción anterior CALL cc, nn.

#### 5.- Regreso de interrupción

Formato: RETI

Esta es una instrucción que permite regresar al programa que se estaba desarrollando antes de que éste fuera interrumpido por una señal de interrupción.

#### 6.- Regreso de una interrupción prioritaria

Formato: RETN

Esta es una instrucción que permite regresar al programa que se estaba desarrollando antes de que se presentara la interrupción prioritaria (no mascarable).

#### 7.- Reinicio en página cero

Formato: RST p

Esta instrucción permite realizar saltos a una de ocho localidades del área de memoria de más bajo nivel, las cuales pueden ser indicadas con ocho bits.

Su función es almacenar el byte de más alto orden del contador de programa en la localidad apuntada por SP-1, y el byte de más bajo orden del contador de programa en la localidad apuntada por SP-2.

Finalmente el contador de programa es cargado con cero en su byte de más alto orden y con p en su byte de más bajo orden, pudiendo tomar uno de los valores siguientes:

00H, 08H, 10H, 18H, 20H, 28H, 30H, y 38H.

Esta instrucción es empleada por ciertos dispositivos periféricos para generar una interrupción vectorizada.

## Operaciones de entrada y salida

### 1.- Entrada de información al acumulador

Formato: IN A, (n)

En esta instrucción el contenido del puerto n pasará al acumulador.

n puede tener valores desde 0 a 255, correspondientes a los puertos que puede manejar el Z80 CPU.

Cuando se ejecuta esta instrucción, la dirección del puerto n se coloca en las líneas de dirección A7 a A0, y el contenido del puerto pasará al acumulador.

### 2.- Entrada de información indirecta a un registro.

Formato: IN r, (c)

r puede ser cualquier registro de un byte del Z80 CPU.

El contenido del registro C indicará el puerto con el cual se va a realizar la transferencia de datos.

En esta instrucción el puerto indicado por C es leído y su contenido colocado en el registro r.

La dirección indicada por C se coloca en las líneas A7 a A0 del bus de direcciones y el contenido del puerto pasará al registro r a través de las líneas A8 a A15

- 3.- Entrada semiautomática de bloques de información incrementando apuntador de memoria.

Formato: INI

En esta instrucción se transfiere un dato del puerto, apuntado por C, a la localidad de memoria apuntada por HL.

Posteriormente se incrementa el valor del apuntador HL y se decrementa el contenido del registro B, que actúa como un contador de eventos.

- 4.- Entrada automática de bloques de información incrementando apuntador de memoria.

Formato: INIR

Esta instrucción es similar a la anterior, solamente que la acción se realiza n veces hasta que el contador B llegue a cero.

- 5.- Entrada semiautomática de bloques de información decrementando apuntador de memoria.

Formato: IND

Esta instrucción transfiere un dato del puerto, apuntado por el registro C, a la localidad de memoria apuntada por HL, y a continuación se decrementa en uno al apuntador HL y se decrementa en uno al contador B.

- 6.- Entrada automática de bloques de información decrementado apuntador de memoria.

Formato: INDR

Esta instrucción es similar a la anterior, pero se realiza n veces hasta que el contador B llegue al valor cero.

#### 7.- Salida de información desde el acumulador

Formato: OUT (n), A

En esta instrucción se transfiere el contenido del acumulador a un puerto especificado por n.

n puede tomar cualquier valor en el rango 0-255 correspondiente a los puertos que el Z80 CPU puede manejar.

Cuando se ejecuta esta instrucción la dirección del puerto n se coloca en las líneas A7-A0 del bus de direcciones, y el contenido del acumulador pasará al puerto a través de las líneas A8 a A15 del bus de direcciones.

#### 8.- Salida de información indirecta desde un registro

Formato: OUT (c), r

r puede ser cualquier registro del Z80 CPU.

El contenido del registro C indicará el puerto con el que se va a realizar la transferencia de datos.

En esta instrucción el contenido del registro r es transferido al puerto indicado por el registro C.

La dirección indicada por el registro C se coloca en las líneas A7 a A0 del bus de direcciones, y el contenido del registro r pasará al puerto apuntado por el registro C a través de las líneas A8 a A15.

#### 9.- Salida semiautomática de bloques de información incrementando apuntador de memoria

Formato: OUTI

En esta instrucción se transfiere un dato de la localidad de memoria apuntada por HL, al puerto apuntado por el registro C, y a continuación se incrementa en uno al apuntador HL, y finalmente se decrementa en uno el contenido del registro B, que actúa como un contador.

- 10.- Salida automática de bloques de información incrementando apuntador de memoria.

Formato: OUTIR

Esta instrucción es similar a la anterior (OUTI), sólo que se realiza n veces hasta que el contador B llegue al valor cero.

- 11.- Salida semiautomática de bloques de información decrementando apuntador de memoria.

Formato: OUTD

En esta instrucción se transfiere un dato de la localidad de memoria apuntada por HL, al puerto apuntado por el registro C, y a continuación se decrementa en uno al apuntador HL, y también se decrementa en uno al registro B, que actúa como un contador.

- 12.- Salida automática de bloques de información decrementando al apuntador de memoria.

Formato: OUTFDR

Esta instrucción es similar a la anterior (OUTD), sólo que se realiza n veces hasta que el contenido del contador B sea cero.

## SISTEMAS DE MICROCOMPUTADORA

---

Para que sea posible utilizar al microprocesador Z80 CPU requerimos del empleo de algunos elementos complementarios, que son:

- 1.- Fuente de alimentación de 5V.
- 2.- Oscilador de señal cuadrada.
- 3.- Memoria para almacenar el programa.
- 4.- Control para inicializar el sistema.
- 5.- Dispositivos de entrada y salida.

La fuente de alimentación debe proporcionar un voltaje constante de 5V, para alimentar al Z80 CPU y a los demás circuitos asociados, siendo éste el nivel de voltaje típico para circuitos TTL.

El oscilador deberá generar pulsos uniformes, con una amplitud de 5V y una frecuencia comprendida entre 2.5 MHz y 4.0 MHz.

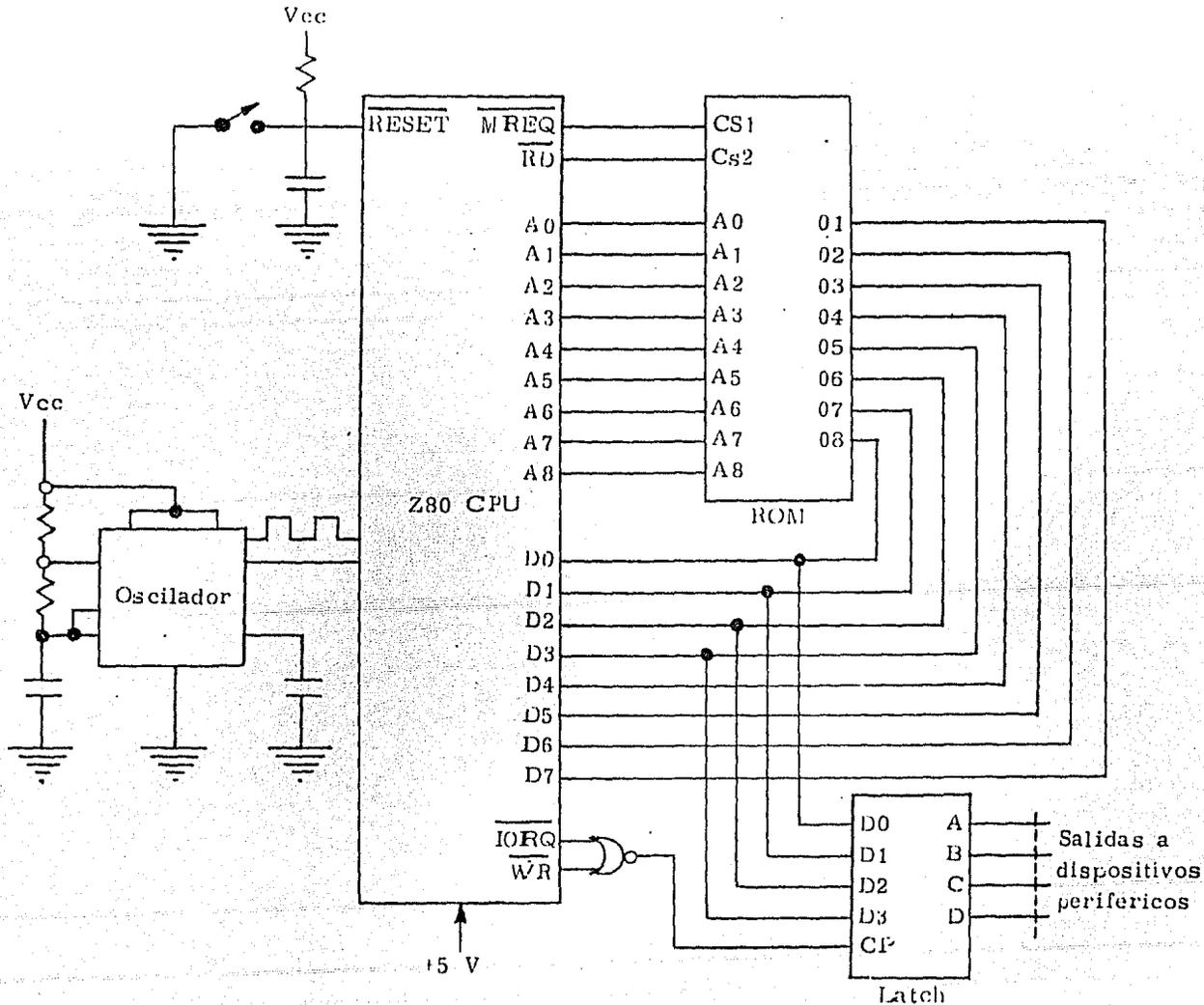
La memoria que se utiliza para almacenar el programa es de sólo lectura, ya sea del tipo preprogramada ROM o del tipo programable PROM, EPROM o EEPROM.

El control para inicializar el programa actúa sobre la línea RESET del Z80 CPU y puede ser un simple interruptor.

Los dispositivos de entrada y salida tienen como función el transferir información entre el Z80 CPU y el mundo exterior y pueden ser multivibradores biestables, como se muestra en la página siguiente, o pueden ser circuitos mucho más sofisticados, como el Z80 PIO o el Z80 CTC.

La forma en que opera el sistema es la siguiente:

Un pulso sobre la línea RESET con el control de inicialización del sistema, hace que el contador de programa se coloque en la primera localidad de la memoria, incrementándose posteriormente de acuerdo a los pulsos generados por el oscilador.



SISTEMA MINIMO DE MICROCOMPUTADORA

La memoria en la que está contenido el programa es del tipo ROM, de 8 bits y tiene 512 localidades; cualquier localidad de las 512 es seleccionada a través de las líneas A0 - A8 del Z80 CPU; en este sistema la memoria deberá ser de acceso rápido y con ésto simplificar la operación del equipo; por acceso rápido se entiende un tiempo no mayor de 250 ns. Las salidas de la memoria son de tres estados, así es, que todas las salidas están en un nivel de alta impedancia, con excepción de la localidad accesada, con lo cual se asegura que la información leída corresponde únicamente a dicha localidad.

Las ocho líneas de datos del microprocesador se conectan directamente a las líneas correspondientes de la memoria, y además, se sacan en paralelo hacia el dispositivo de entrada y salida, ya que las líneas D0 - D7 son bidireccionales.

El dispositivo de entrada y salida consiste en cuatro multivibradores biestables, los cuales son controlados por D0 - D7 cuando se realiza una operación de salida, o bien son leídos desde D0 - D7 por el microprocesador durante una operación de entrada.

En un sistema como éste no podemos almacenar datos o resultados en ningún tipo de memoria, ya que ROM no permite almacenar información, sin embargo, pueden ser utilizados algunos registros del Z80 CPU para este fin, aún cuando las funciones de los registros se desaprovechan al emplearlos de este modo.

En la práctica no se construyen sistemas como el anterior, ya que no se aprovecha eficazmente la capacidad del microprocesador.

Básicamente un sistema lleva memoria para almacenar datos, variables y resultados, y esta memoria debe ser, por lo tanto, de lectura y escritura, y se denomina memoria de acceso aleatoria (RAM).

El intercambio de datos con el mundo exterior es una función muy importante y en general se realiza a través de teclados para proporcionar datos y órdenes al sistema, y de registros visuales para obtener los resultados.

Otros dispositivos que se emplean con el microprocesador Z80 CPU son el circuito de conteo y temporización Z80 CTC, la interfase de entrada y salida en paralelo Z80 PIO y la interfase de entrada y salida en serie Z80 SIO. El Z80 CTC por ejemplo nos permite ampliar la capacidad del microprocesador al permitirnos generar funciones de tiempo, programadas por el usuario y también nos permite realizar funciones de conteo de eventos.

## CIRCUITO DE CONTEO Y TEMPORIZACION

El circuito de conteo y temporización Z80 CTC es un componente programable con cuatro canales independientes, los cuales nos permiten realizar o efectuar funciones de conteo y funciones de temporización en un sistema de microcomputadora, basado primariamente en un Z80 CPU.

El Z80 CPU puede operar los canales del Z80 CTC para que funcionen bajo varios modos y condiciones, dependiendo de las características de los circuitos o dispositivos externos.

El Z80 CTC es un circuito de tecnología NMOS y está encapsulado en una pastilla con 28 conectores, requiriendo únicamente una fuente de alimentación de 5V y un reloj con pulsos de 5V de amplitud.

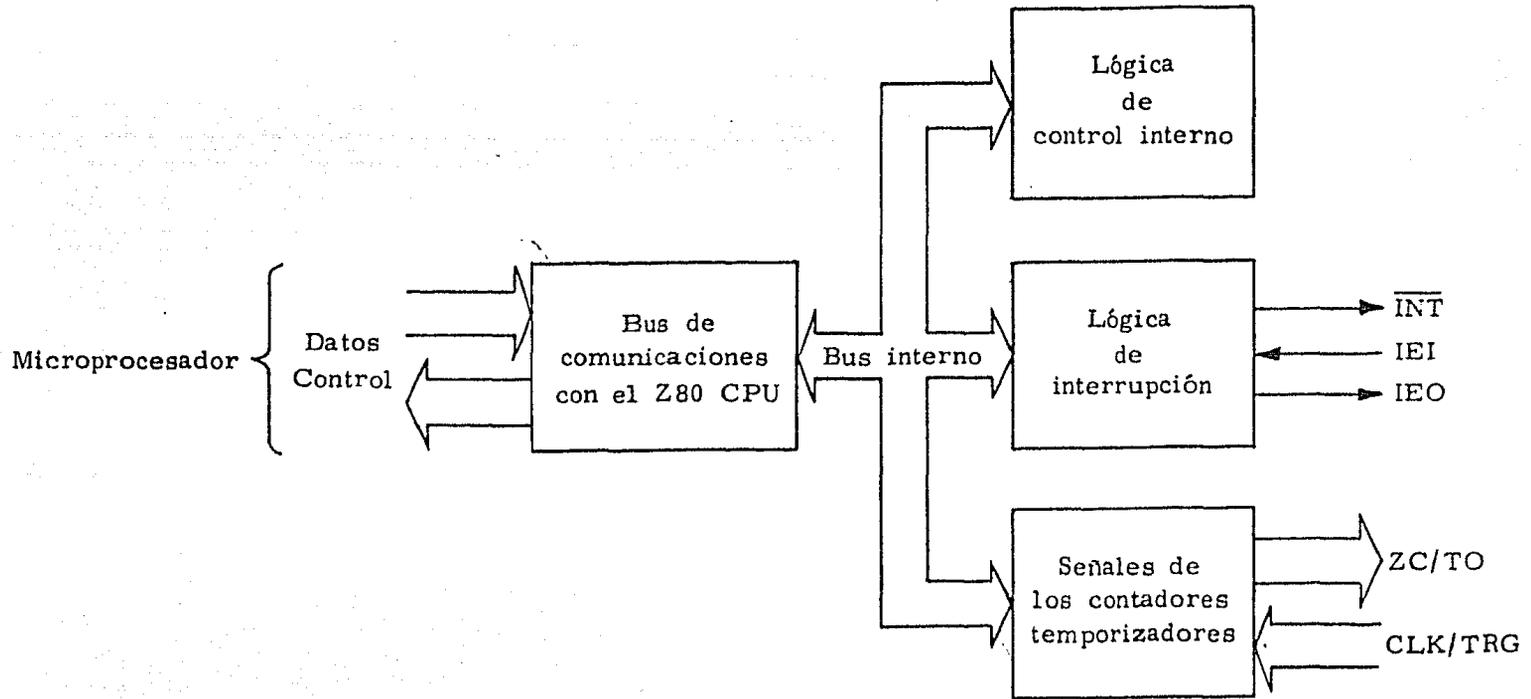
## ARQUITECTURA DEL Z80 CTC

El circuito Z80 CTC está formado por cuatro elementos principales:

- 1.- Bus de entrada y salida al Z80 CPU
- 2.- Control interno de canales
- 3.- Lógica de control de interrupción
- 4.- Circuitos contadores temporizadores

### Bus de entrada y salida al Z80 CPU

El circuito de entrada y salida decodifica las direcciones de entrada y forma la interfase para las señales de control y del bus de datos entre los circuitos Z80 CPU y el Z80 CTC.



ARQUITECTURA DEL Z80 CTC

### Control interno de canales

Los circuitos de control interno, como su nombre lo indica, permiten controlar la operación de todo el circuito integrado; estos circuitos controlan las funciones del Z80 CTC como son las siguientes: reinicializarlo, activarlo y permitir las funciones de lectura o escritura.

### Lógica de control de interrupción

Esta parte del circuito integrado asegura que las señales de interrupción funcionen adecuadamente entre este circuito y el Z80 CPU.

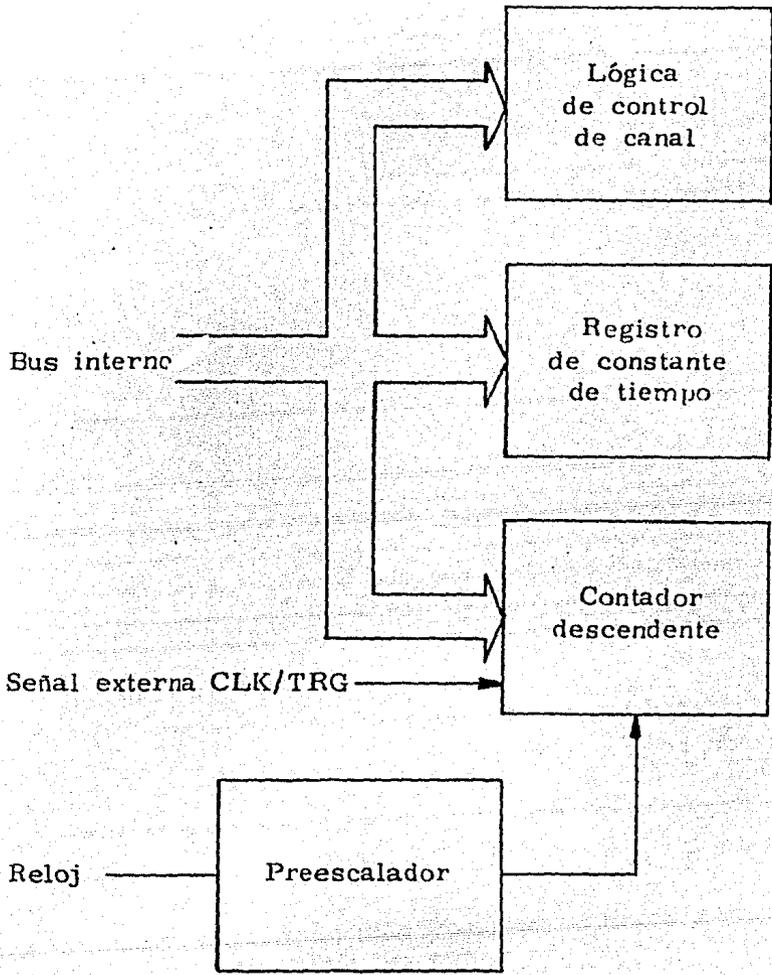
### Circuitos contadores temporizadores

El Z80 CTC tiene cuatro circuitos independientes de conteo y temporización, contando cada uno de ellos con los siguientes elementos:

- 1.- Lógica para control de canal
- 2.- Registro de constante de tiempo
- 3.- Preescalador
- 4.- Contador descendente

### Lógica para control de canal

La lógica para control de canal recibe una palabra de control de canal de ocho bits cuando el circuito es programado y la decodifica fijando las condiciones de operación del circuito; el bit cero de la palabra de control de canal es el indicativo de la instrucción; si este bit toma el valor de uno, indicamos que la palabra seleccionará el modo de operación del circuito, el rango, el flanco de disparo, etc., de la siguiente forma:



ARQUITECTURA DE UN CONTADOR TEMPORIZADOR

Bit siete es el habilitador de interrupción con nivel lógico alto, y deshabilitador en caso contrario.

Bit seis es el control de operación; con cero lógico activa al Z80 CTC como temporizador y con uno lógico como contador de eventos.

Bit cinco. Permite programar el factor de preescala; con uno lógico se da un valor de 256 y con cero lógico un valor de 16 (sólo cuando el circuito es programado como temporizador).

Bit cuatro. Selecciona el momento del disparo de la señal de reloj o pulso de disparo CLK/TRG; con cero lógico y operando como contador, el flanco de caída de la señal decremента el contador, operando como temporizador, el flanco de caída inicia la operación de temporización. Un cero lógico indica que es el pulso de subida de la señal (CLK/TRG) el que decremента al contador e inicia la operación de temporización si está programado el canal como temporizador.

Bit tres. Indica el disparo del sistema cuando opera como temporizador; en cero lógico indica que el disparo del temporizador se inicia cuando se termina de programar el circuito, y un uno lógico indica que el circuito operará con la señal de reloj o pulso de disparo.

Bit dos. Indica que el Z80 CTC será cargado con una constante a partir de la cual contará hasta cero, si es programado con uno lógico; si es programado un cero lógico no se cargará ninguna constante de tiempo, este caso, se presenta cuando un canal está en operación y la palabra de control de canal sólo modifica alguna característica de operación del circuito.

Bit uno. Indica con uno lógico que se desea reprogramar al circuito con una nueva constante de tiempo o con otra característica sin que el circuito deje de operar; cuando el valor de este bit es cero se da una reinicialización (RESET) al circuito.

Bit cero. Indica con un cero lógico que estamos cargando un vector de interrupción y con nivel uno estamos programando el Z80 CTC.

### Registro de constante de tiempo

Se utiliza en ambos modos de operación, programándose justo después de la palabra de control de canal con un valor constante de 1 a 256. Este registro carga el valor programado en el contador descendente cuando el CTC es inicializado, y recargado con el mismo valor automáticamente, siempre que pasa la cuenta por cero. Si se carga una nueva constante de tiempo mientras un canal está funcionando, la cuenta presente se completará antes de que el nuevo valor sea cargado en el registro.

### Preescalador

El preescalador solamente se utiliza en modo de temporización. Es un registro de 8 bits, el cual puede ser programado por el Z80 CPU a través del registro de control de canal para dividir su entrada al reloj del sistema por 16 ó 256. La salida del modificador de escala, es llevada como entrada para acompasar el contador descendente, divide la señal del reloj del sistema por un factor adicional que es el de la constante de tiempo. Cada vez que el contador descendente pasa por cero, su salida ZC/TO es pulsada a nivel alto.

### Contador descendente

Este es un registro de 8 bits utilizado en ambos modos de operación, cargado inicialmente y recargado cada vez que cuenta hasta cero, con el valor del registro de constante de tiempo. Es decrementado por cada pulso del reloj externo en modo de conteo o en modo de temporización por la salida del modificador de escala.

En los canales 0, 1, 2, cuando se alcanza la condición de cuenta en cero, un pulso aparece en la salida ZC/TO correspondiente (el canal 3 del Z80 CTC no tiene esta salida).

## VECTOR DE INTERRUPCION

El vector de interrupción es usado para formar un puntero a una dirección de la memoria, donde la dirección de la rutina de servicio a interrupción está almacenada en una tabla.

Existe una convención en el sistema del Z80 CTC. Todas las direcciones en la tabla de la rutina de servicio a interrupción deben tener su byte de más bajo orden en una localidad par de la memoria, y el byte de más alto orden a la próxima localidad de la memoria. El bit menos significativo de cualquier vector de interrupción será siempre cero, indicando que se envía la programación del vector de interrupción.

La instrucción RETI es usada al final de cualquier rutina de interrupción para inicializar la línea de acople del sistema de prioridad sucesiva (IEO) para el manejo apropiado de interrupciones anidadas. El Z80 CTC monitorea el sistema del bus de datos y decodifica esta instrucción cuando ocurre. Por lo que la lógica de control de canal del Z80 CTC siempre conocerá cuando el Z80 CPU ha completado una rutina de servicio de interrupción.

Un canal del Z80 CTC puede ser programado para solicitar una interrupción cada vez que el contador descendente alcance la cuenta en cero (utilizar esta característica requiere programación del modo 2 de interrupción en el Z80 CPU). Después de la solicitud el Z80 CPU enviará un reconocimiento de interrupción y la lógica de control del Z80 CTC determinará cual canal está solicitando la interrupción.

Los 5 bits de más alto orden del vector deberán ser escritos en el Z80 CTC previamente, como parte de la programación. Los otros bits serán dados por la parte de lógica de control de interrupción del Z80 CTC.

## DESCRIPCION DEL CIRCUITO

Funcionalmente los conectores del Z80 CTC se clasifican en cinco grupos que son:

- 1.- Bus de datos
- 2.- Control del circuito
- 3.- Control de interrupciones
- 4.- Señales de control de canal
- 5.- Señales de reloj y alimentación

A continuación analizaremos las señales de cada grupo e indicaremos su posición en la pastilla del circuito.

### Bus de datos

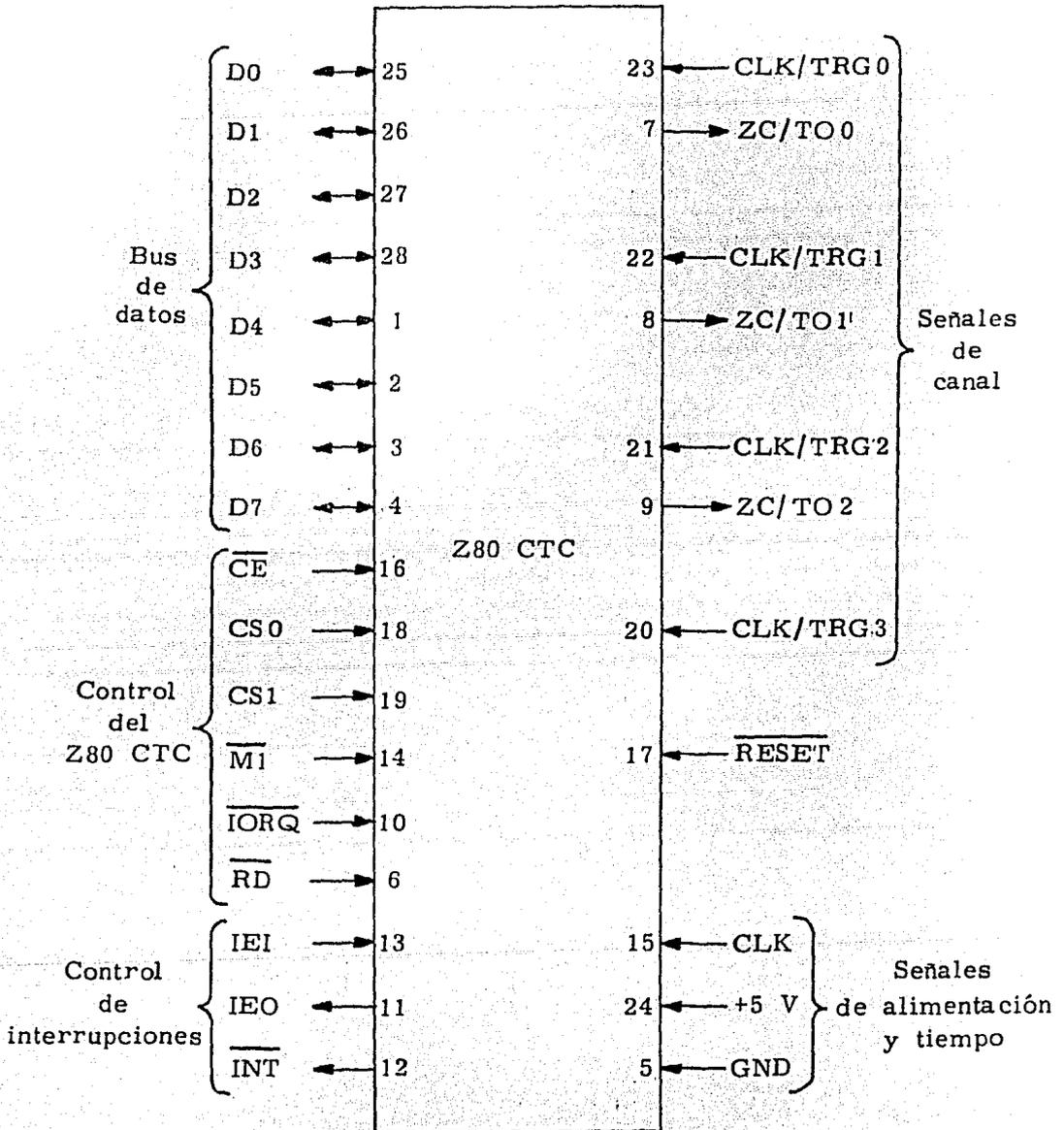
Este bus es utilizado para transferir todos los comandos y datos entre el Z80 CPU y el Z80 CTC. Este bus es de ocho bits y por lo tanto utiliza ocho conectores de la pastilla siendo el más significativo el D7.

Las posiciones que guardan estos conectores en el circuito son: D 7 posición 4, D 6 posición 3, D 5 posición 2, D 4 posición 1, D 3 posición 28, D 2 posición 27, D 1 posición 26, D 0 posición 25.

### Control del circuito

En esta sección se agrupan siete señales del Z80 CTC las cuales se detallan a continuación:

- 1.- Selección de canal (CS1) - (CS0)
- 2.- Acoplamiento de circuito (CE)
- 3.- Primer ciclo de máquina (M1)
- 4.- Solicitud de entrada y salida (IORQ)
- 5.- Ciclo de lectura (RD)
- 6.- Señal de reposición (RESET)



LINEAS DEL Z80 CTC

### Selección de canal (CS1) - (CS0)

Estos dos conectores forman una dirección de dos bits y generan un código de dirección para seleccionar uno de los cuatro canales del Z80 CTC de la siguiente forma:

Señal CS1	Señal CS0	Canal seleccionado
0	0	Canal 0
0	1	Canal 1
1	0	Canal 2
1	1	Canal 3

Los conectores correspondientes son:

CS1 posición 18 y CS0. posición 19.

### Acoplamiento de circuito (CE)

Cuando este conector toma nivel bajo permite que el Z80 CTC acepte palabras de control, vectores de interrupción o datos de constante de tiempo, desde el bus de datos durante un ciclo de escritura de entrada o salida, o también para transmitir los contenidos del contador descendente al Z80 CPU durante un ciclo de lectura de entrada o salida. Esta señal tiene la posición 16 en el circuito integrado.

### Primer ciclo de máquina (M1)

Cuando se activa esta señal sin activarse la señal IORQ, el Z80 CPU está buscando y registrando una instrucción desde memoria. Cuando la señal M1 y la señal IORQ son activas el Z80 CPU está reconociendo una interrupción, alertando al Z80 CTC para que coloque un vector de interrupción en el bus de datos, si es que tiene preferencia en la configuración de prioridad sucesiva.

### Solicitud de entrada y salida (IORQ)

Esta señal es usada en conjunto con las señales CE y RD para transferir datos y palabras de control de canal entre el Z80 CPU y el Z80 CTC.

Durante un ciclo de escritura la señal IORQ y la señal CE deben estar activas y la señal RD inactiva.

El Z80 CTC no genera señales de escritura, pues la implementa internamente a partir de la inversión de la señal RD.

En el ciclo de lectura las señales IORQ, CE y RD, deberán estar activas para colocar el contenido del contador descendente en el bus de datos.

### Ciclo de lectura (RD)

La señal RD es utilizada en conjunción con las señales IORQ y CE para transferir datos y palabras de control de canal entre el Z80 CPU y el Z80 CTC.

Durante un ciclo de escritura IORQ y CE deben estar activas y RD inactiva (el Z80 CTC no recibe señal de escritura ya que la genera internamente de la inversión de la señal RD). En el ciclo de lectura del Z80 CTC, las señales IORQ, CE, RD, deben ser activas para colocar el contenido del contador descendente en el bus de datos.

### Señal de reposición (RESET)

Esta señal tiene por objeto finalizar las acciones del Z80 CTC; repone el bit de acoplamiento de interrupción de canal en todos los registros de control, con lo que inhibe las interrupciones generadas por el Z80 CTC.

### Control de interrupciones

El Z80 CTC cuenta con tres líneas para el manejo adecuado del sistema de interrupción siendo estas señales las siguientes:

- 1.- Solicitud de interrupción (INT)
- 2.- Entrada habilitadora de interrupción (IEI)
- 3.- Salida habilitadora de interrupción (IEO)

### Solicitud de interrupción (INT)

Esta señal que actúa con un nivel lógico bajo es utilizada para solicitar una interrupción al Z80 CPU desde cualquier canal del Z80 CTC; esta señal tiene la posición 12 en el circuito integrado.

### Entrada habilitadora de interrupción (IEI)

Esta línea de entrada al Z80 CTC actúa con nivel lógico alto y es utilizada en la configuración de prioridad sucesiva cuando se utiliza más de un dispositivo con capacidad de interrumpir al Z80 CPU de forma vectorizada.

Esta línea tiene la posición 13 en el circuito integrado Z80 CTC.

### Salida habilitadora de interrupción (IEO)

Esta señal que actúa con nivel lógico bajo es utilizada en la configuración de prioridad sucesiva, teniendo la función de bloquear la capacidad de interrupción de los dispositivos de más baja prioridad que él; esta línea ocupa la posición número 11 en el circuito integrado.

### Señales de control de canal

Para el control de los cuatro canales del Z80 CTC se cuenta con dos líneas para cada uno de los tres primeros canales, y una línea para el control del cuarto canal.

1.- Sincronía externa y disparo (CLK/TRG)

2.- Cuenta en cero y tiempo fuera (ZC/TO)

### Señales de sincronía externa y disparo (CLK/TRG)

Estas señales que entran al sistema del Z80 CTC son cuatro, una para cada canal, y son activadas con un nivel lógico bajo.

Las posiciones que ocupan en el circuito son las siguientes:

CLK/TRG<sub>0</sub> posición 23, CLK/TRG<sub>1</sub> posición 22, CLK/TRG<sub>2</sub> posición 21, y CLK/TRG<sub>3</sub> posición 20.

Cuando un canal opera como contador de eventos, cada pulso en estas líneas decrementa en uno al contador descendente en cuestión. Cuando un canal opera como temporizador un pulso en estas líneas lo activa.

### Cuenta en cero y tiempo fuera (ZC/TO)

Estas son tres líneas que parten del Z80 CTC hacia el exterior faltando una para el canal tres del circuito; esto quiere decir que los tres primeros canales (el cero, el uno, y el dos) pueden indicar que la cuenta ha llegado a cero, activando esta línea; el canal tres no tiene esta capacidad.

Las posiciones que ocupan estas líneas en el circuito integrado son las siguientes: ZC/TO<sub>0</sub> posición 7, ZC/TO<sub>1</sub> posición 8 y ZC/TO<sub>2</sub> posición 9.

### Señales de reloj y alimentación

- 1.- Señal de reloj
- 2 - Señales de alimentación

### Señal de reloj

Esta entrada al sistema es aceptada en la línea 15 del circuito integrado Z80 CTC y es la misma que se utiliza para el Z80 CPU; sirve para sincronizar internamente ciertas señales en el circuito.

### Señales de alimentación

El circuito Z80 CTC deberá ser alimentado con una señal de cinco volts en el conector 24, y el nivel de tierra deberá ser referido al conector 5 del Z80 CTC.

## MODOS DE OPERACION DEL Z80 CTC

La señal RESET pone al Z80 CTC en un estado conocido para seleccionar el modo de operación, ya sea conteo o temporización. Una palabra de control de canal y una constante de tiempo deberán ser escritas en sus respectivos registros de canal.

Si cualquier canal ha sido programado para aceptar interrupciones se deberá escribir una palabra de vector de interrupción en el control de lógica de interrupciones del Z80 CTC. Cuando el Z80 CPU ha escrito todas las palabras en el Z80 CTC todos los canales pueden ser programados para operación inmediata en ambos modos, temporización y conteo.

### Modo de conteo del Z80 CTC

En este modo el Z80 CTC cuenta los flancos de la entrada CLK/TRG. El modo de conteo es programado para un canal cuando la palabra de control de canal es escrita con el bit 0 fijado a 1. La entrada del canal para reloj externo CLK/TRG es monitoreada para una serie de flancos de disparo, después de los cuales y en sincronía con el siguiente flanco de subida del reloj, el contador descendente que fue inicializado con el dato de constante de tiempo al inicio de la secuencia de conteo, es decrementado.

Una entrada del reloj externo del canal es programada por el bit 4 de la palabra de control de canal para disparar la secuencia de decremento con cualquier flanco, ya sea el de subida o el de bajada.

En cualquiera de los canales 0, 1 y 2 cuando el contador descendente es sucesivamente decrementado desde el valor de la constante de tiempo hasta que finalmente alcanza el cero, la salida de cuenta en cero ZC/TO es activada, si el canal ha sido programado para ésto (por el bit 7 de la palabra de control de canal).

La condición de cuenta en cero, además, resulta en una recarga automática del contador descendente con la constante de tiempo original, en el registro de constante de tiempo.

No existe interrupción en la secuencia de conteo descendente si el registro de constante de tiempo es escrito con un nuevo valor mientras el contador descendente es decrementado; la cuenta presente se complementará antes que la nueva constante sea cargada en el contador descendente.

### Modo de temporización

Si se selecciona el modo de temporización la palabra de control debe fijar también un modificador de escala, el cual divide la señal de reloj del sistema por un factor de 16 ó 256. La constante de tiempo es un valor entre 1 y 256.

En este modo se determinan intervalos de tiempo tan pequeños como 6.4 ms sin necesidad de lógica externa. El intervalo de tiempo es generado dividiendo la señal de reloj del sistema empleando para ello el modificador de escala. El temporizador es disparado automáticamente cuando su valor de constante de tiempo es programado para ello.

### Aplicaciones de los contadores y temporizadores

Estos circuitos programables se pueden aplicar a una amplia gama de propósitos, como son contadores de eventos, generadores de frecuencia, relojes de tiempo real, controladores para motores complejos, para medir velocidades, etc.

En el presente trabajo el Z80 CTC será empleado como un contador de eventos en conjunción con un arreglo de emisor y sensor de luz, con el cual se detectará cada pulso luminoso que pase a través del arreglo y esta señal será transferida a través de un puerto al circuito Z80 CTC.

### BUS NORMALIZADO

En muchas aplicaciones es necesario conectar dispositivos periféricos al sistema de microcomputadora y en muchos casos estos periféricos no son compatibles, conector por conector, con los del microprocesador.

Para conectar dispositivos periféricos al sistema de microcomputadora, es usual utilizar buses normalizados para lograr que estos sistemas sean lo más versátiles posible; el nanocomputador Z30 Starter Kit utiliza un tipo de bus normalizado denominado Bus S-100.

## DESCRIPCION DEL BUS

El Bus S-100 está constituido por un conjunto de cien contactos, montados en dos filas de cincuenta conectores por fila; generalmente el bus está montado en un bastidor de tarjetas, quedando las terminales 1 hasta la 50 en el lado de la tarjeta donde se ubican los componentes electrónicos, y las terminales 51 a 100 del lado de la tarjeta donde se encuentran las conexiones entre los componentes electrónicos:

Las cien líneas de señales del bus se dividen funcionalmente en ocho grupos que son los siguientes:

- 1.- Bus de direcciones.
- 2.- Bus de datos.
- 3.- Bus de estados.
- 4.- Bus de control para señales de salida.
- 5.- Bus de control para señales de entrada.
- 6.- Bus de interrupciones.
- 7.- Bus de acceso directo a memoria.
- 8.- Bus de servicio general.

### Bus de direcciones

Este bus está formado por 24 líneas denominadas A23, A22, A21, etc. hasta A0. Es utilizado para seleccionar ya sea un dispositivo periférico, o bien para especificar una cierta localidad de memoria.

Las líneas de este bus guardan las posiciones siguientes:

La línea A0 tiene la posición 79, A1 la posición 80,  
 A 2 la posición 81, A 3 la posición 31,  
 A 4 la posición 30, A 5 la posición 29,  
 A 6 la posición 82, A 7 la posición 83,  
 A 8 la posición 84, A 9 la posición 34,  
 A10 la posición 37, A11 la posición 87,  
 A12 la posición 33, A13 la posición 85,  
 A14 la posición 86, A15 la posición 32,  
 A16 la posición 16, A17 la posición 17,  
 A18 la posición 15, A19 la posición 59,  
 A20 la posición 61, A21 la posición 62,  
 A22 la posición 63, A23 la posición 64.

### Bus de datos

El bus de datos cuenta con dieciseis líneas, permitiendo con ésto que se transfieran datos con dos bytes de longitud; para ello debe ser utilizada una línea de control denominada XTRQ ya que el S-100 permite el uso del bus como un bus de entrada de datos de ocho bits y bus de salida de datos de ocho bits.

El bus de datos de entrada se denomina DI y el bus de datos de salida se denomina DO; la entrada y salida es referida a la dirección del flujo de datos del dispositivo principal o maestro.

Las posiciones en el S-100 del bus de datos de entrada DI son las siguientes: DI0 la posición 95, DI1 la posición 94, DI2 la posición 41, DI3 la posición 42, DI4 la posición 91, DI5 la posición 92, DI6 la posición 93 y DI7 la posición 43.

Las posiciones en el S-100 del bus de datos de salida DO son las siguientes: DO0 la posición 36, DO1 la posición 35, DO2 la posición 88, DO3 la posición 89, DO4 la posición 38, DO5 la posición 39, DO6 la posición 40, y DO7 la posición 90.

Para transferencias de información de 16 bits o 2 bytes, tenemos las correspondencias siguientes:

dato 0 posición 36,	dato 1 posición 35,
dato 2 posición 38,	dato 3 posición 39,
dato 4 posición 38,	dato 5 posición 39,
dato 6 posición 40,	dato 7 posición 90,
dato 8 posición 95,	dato 9 posición 94,
dato 10 posición 41,	dato 11 posición 42,
dato 12 posición 91,	dato 13 posición 92,
dato 14 posición 93, y	dato 15 posición 43.

### Bus de estados

La información que pasa a través de estas líneas, es utilizada por los dispositivos conectados a él para determinar el tipo de ciclo que se está ejecutando, ya sea un ciclo de lectura en memoria, o un ciclo de escritura en dispositivos de entrada y salida. Las líneas de este bus son las siguientes:

- 1.- Primer ciclo de máquina (sM1).
- 2.- Lectura de memoria (sMEMR).
- 3.- Señal de entrada (sINP).
- 4.- Señal de salida (sOUT).
- 5.- Señal de escritura (sWO).
- 6.- Reconocimiento de interrupción (sINTA).
- 7.- Señal de alto (sHLTA).
- 8.- Solicitud de transferencia de 16 bits (sXTRQ).

### Primer ciclo de máquina (sM1)

Esta señal es la misma del Z80 CPU y tiene por objeto indicar que se ejecuta el ciclo de búsqueda del código de operación de la instrucción a ser ejecutada; esta señal se localiza en la posición 44 del bus S-100 y actúa con nivel lógico alto.

### Lectura de memoria (sMEMR)

Esta señal indica que el Z80 CPU o algún otro microprocesador está leyendo un dato desde una localidad de memoria. Tiene la posición 47 en el bus y es activa con nivel lógico alto.

### Señal de entrada (sINP)

Esta señal se activa en nivel lógico alto cuando el microprocesador realiza un ciclo de entrada leyendo datos, desde un puerto. Tiene la posición 46 en el bus S-100.

### Señal de salida (sOUT)

Esta señal se activa cuando el microprocesador está ejecutando un ciclo de salida y escribe datos en un puerto. Tiene la posición 45 y actúa con nivel lógico alto.

### Señal de escritura (sWO)

Esta señal de estado indica, cuando está activa (con nivel lógico bajo), que el microprocesador está ejecutando una escritura en memoria o bien un ciclo de escritura en un dispositivo periférico.

### Reconocimiento de interrupción (sINTA)

Cuando el microprocesador acepta una interrupción por un dispositivo periférico se hace activa esta señal con un nivel lógico alto. Tiene la posición 96 en el S-100.

### Señal de alto (sHLTA)

Esta línea se activa con un nivel lógico alto cuando el Z80 CPU entra en un estado de alto (HALT), y está localizada en la posición número 48 del S-100.

### Señal de transferencia de 16 bits (sXTRQ)

Esta señal se activa con un nivel lógico bajo y es utilizada para solicitar las dieciséis líneas del bus de datos; en caso de ser inactiva el bus de datos se comportará como dos buses unidireccionales de ocho bits cada uno. La posición de esta línea en el S-100 es la 58.

## Bus de control para señales de salida

En este bus se agrupan cinco señales que son utilizadas para identificar memorias y puertos y el tipo de ciclo que ocurre en el sistema; las señales son las siguientes:

- 1.- Sincronizar (pSYNC)
- 2.- Estado válido (pSTVAL)
- 3.- Bus de datos de entrada (pDBIN)
- 4.- Escritura (pWR)
- 5.- Reconocimiento de solicitud de bus (pHLDA)

### Sincronizar (pSYNC)

Esta es una señal de muestreo que indica el inicio de cada ciclo del bus. Se activa con nivel lógico alto y se encuentra localizada en la posición 76.

### Estado válido (pSTVAL)

Esta señal indica que la información en el bus de direcciones es válida, y también la información en el bus de estados.

Se encuentra localizada en la posición 25 y actúa con un nivel bajo de voltaje.

### Entrada a bus de datos (pDBIN)

Esta señal muestreadora se activa con nivel lógico alto en los ciclos de lectura en memoria, lectura en puertos de entrada y en ciclos de reconocimiento de interrupción.

Está localizada en la posición 78 del S-100.

### Escritura (pWR)

Esta señal se activa en los ciclos de escritura en memoria y escritura en puertos de salida, y está localizada en la posición 77 activándose con nivel lógico bajo.

## Reconocimiento de solicitud de bus (pHLDA)

Esta señal indica que el microprocesador ha cedido el control del bus S-100 a algún dispositivo periférico; actúa con nivel lógico alto y tiene la posición 26 en el bus S-100.

## Bus de entrada de control

Este bus consta de cuatro líneas las cuales son controladas por los dispositivos periféricos, y tienen por objeto indicarle al dispositivo principal alguna condición de operación.

Las líneas de este bus son las siguientes:

- 1.- Señal de disponibilidad (RDY)
- 2.- Señal auxiliar de disponibilidad (XRDY)
- 3.- Señal de abandono de bus (HOLD)
- 4.- Señal de reconocimiento para 16 bits (SIXTN)

## Señal de disponibilidad (RDY)

Si esta señal toma un nivel lógico bajo se indica al microprocesador que el dispositivo periférico no está listo para realizar su función. Esta señal tiene la posición 72 del bus S-100.

## Señal auxiliar de disponibilidad (XRDY)

Esta señal es idéntica a la anterior y su función es la misma: su posición en el bus S-100 es la número 3.

## Señal de abandono de bus (HOLD)

Con esta señal se solicita al microprocesador el control del bus y se desactiva solamente al recibir el control del bus.

Actúa con nivel lógico bajo y se encuentra en la posición 74 del S-100.

## Señal de reconocimiento para 16 bits (SIXFN)

Esta señal es utilizada en transferencias de información de 16 bits y actúa con nivel lógico bajo teniendo la posición número 60 en el bus S-100.

## Bus de interrupciones

Este bus está constituido por 10 líneas que son utilizadas por los dispositivos periféricos para interrumpir la ejecución del programa que está realizando el microprocesador, obligando a éste a que proceda a realizar una rutina de servicio a interrupción.

Las líneas de interrupción corresponden a las señales INT y NMI del microprocesador Z80 CPU contando además el bus con ocho líneas para solicitar interrupciones vectorizadas, clasificándose estas líneas en tres grupos:

- 1.- Interrupción (INT)
- 2.- Interrupción no mascarable (NMI)
- 3.- Bus de líneas de interrupción (VIX)

## Interrupción (INT)

Esta señal se hace activa con nivel lógico bajo, permaneciendo así hasta que sea reconocida la interrupción por el microprocesador. La posición de esta señal es la 73 del bus S-100.

## Interrupción no mascarable (NMI)

Esta línea de interrupción es de alta prioridad utilizándose cuando el sistema tiene un problema serio, usualmente una pérdida de voltaje; su posición en el bus S-100 es la 12 y se activa en la transición de uno lógico a cero lógico del ciclo de reloj.

## Bus de líneas de interrupción (VIX)

Estas líneas permiten solicitar interrupciones vectorizadas con una prioridad sucesiva, ésto es, una línea de menor prioridad no es atendida mientras una de más alta prioridad esté interrumpiendo al dispositivo principal.

Las posiciones de estas líneas en el bus S-100 son las siguientes:

VI0 posición 4,	VI1 posición 5,
VI2 posición 6,	VI3 posición 7,
VI4 posición 8,	VI5 posición 9,
VI6 posición 10,	VI7 posición 11.

### Bus de acceso directo a memoria

Este bus proporciona cuatro señales que permiten a un dispositivo externo tomar el control del sistema para recibir y solicitar el control del bus.

El bus S-100 proporciona cuatro líneas que permiten a 16 dispositivos externos realizar funciones de acceso directo a memoria; estas líneas se denominan líneas del control TMA.

Las líneas que conforman este bus son:

- 1.- Deshabilitar bus de direcciones (ADSB)
- 2.- Deshabilitar bus de salida de datos (DODSB)
- 3.- Deshabilitar bus de estados (SDSB)
- 4.- Deshabilitar bus de salida de control (CDSB)
- 5.- Líneas de control (TMA)

Las posiciones de estas señales en el bus S-100 y los niveles lógicos con que actúan, son los siguientes:

Deshabilitar bus de direcciones (ADSB), posición 22, y se activa con nivel lógico bajo; Deshabilitar bus de salida de datos (DODSB), tiene la posición 23 en el S-100 y se activa con nivel lógico bajo; Deshabilitar bus de estados (SDSB), tiene la posición 18 del S-100 y actúa con nivel lógico bajo; Deshabilitar bus de salida de control (CDSB), tiene la posición 19 del S-100 y actúa con nivel lógico bajo.

Las líneas de control guardan las siguientes posiciones: TMA0 tiene la posición 55, TMA1 la posición 56, TMA2 la posición 57, y TMA3 la posición 14.

## Bus de servicio general

En este bus se agrupan conectores para recibir alimentación, pulsos de sincronía, etc., y estas señales son:

- 1.- Reloj del sistema ( $\phi$ )
- 2.- Reloj de tiempo real (CLOCK)
- 3.- Habilitador de dispositivo alterno (PHANTOM)
- 4.- Eliminador de transitorios (POC)
- 5.- Reposición (RESET)
- 6.- Reposición de dispositivos subordinados (SLAVE CLR)
- 7.- Señal de error (ERROR)
- 8.- Alarma de alimentación (PWFAIL)
- 9.- Señal de 8 volts
- 10.- Señal de 16 volts
- 11.- Señal de -16 volts
- 12.- Señal de tierra (GND)
- 13.- Líneas libres (NDEF)
- 14.- Líneas reservadas no utilizadas (RFU)

Las posiciones que guardan estas líneas en el sistema son las siguientes:

Reloj del sistema ( $\phi$ ), tiene la posición 24; Reloj de tiempo real (CLOCK), tiene la posición 49; Habilitador de dispositivo alterno (PHANTOM), tiene la posición 67; Eliminador de transitorios (POC), tiene la posición 99; Reposición (RESET), tiene la posición 75; Reposición de dispositivos subordinados (SLAVE CLR), tiene la posición 54; Señal de error (ERROR), tiene la posición 98; Alarma de alimentación (PWFAIL), tiene la posición 13.

La señal de 8V es conectada a las terminales 1 y 51, la señal de +16V a la terminal 2 y la de -16V a la terminal 52, mientras que la señal de tierra (GND) va a las terminales 20, 50, 53, 70 y 100.

Las líneas libres (NDEF) del bus S-100 son la 21, la 65 y la 66, mientras que las líneas reservadas no utilizadas (RFU) tienen asignados los conectores 27, 28, 69 y 71 del bus S-100.

## PROGRAMACION

El diseñar el programa de control para un equipo requiere de una sistematización adecuada, que emplearemos en el caso de este diseño. Primeramente serán consideradas las características generales que debe satisfacer un programa, y se expondrá a continuación el proceso de desarrollo que será empleado en el diseño del programa de control.

La programación de un sistema va transformándose, día a día, en un proceso más elaborado y más complejo, y por supuesto más costoso, y ésto obliga a sistematizar, es decir, es necesario seguir un proceso que nos lleve de las condiciones externas y de los requerimientos básicos, hasta la parte de codificación y pruebas como punto final del desarrollo.

Al realizar programas, uno de los problemas básicos que se presentan es la incidencia de errores, que ocurren muchas veces en las primeras etapas del desarrollo de un proyecto, debido a que los requerimientos de operación no están adecuadamente formulados; estos errores son localizados y corregidos, generalmente, cuando el trabajo de programación está muy adelantado, lo cual provoca que se deba descartar mucho del trabajo realizado y comenzar nuevamente, con todas las implicaciones de pérdida de tiempo y aumento de costos que significa.

La metodología que se aplica para sistematizar el desarrollo de un programa y así minimizar errores y reducir el impacto de éstos sobre el trabajo subsecuente, se denomina Ingeniería de Programación, la cual consta de una serie de procedimientos y métodos para mejorar la calidad y confiabilidad del programa, así como para optimizar los tiempos de desarrollo de un proyecto de programación.

La Ingeniería de Programación tiene como objetivos básicos, el mejorar la calidad de un programa, satisfaciendo los siguientes puntos:

- 1.- Aumentar el control del proceso de desarrollo
- 2.- Disminuir los costos de desarrollo
- 3.- Disminuir los costos de mantenimiento
- 4.- Aumentar la movilidad del sistema

Para que tengamos un programa bien diseñado es fundamental que, al realizarlo, se tomen en cuenta las siguientes características:

- 1.- Confiable
- 2.- Modular
- 3.- Estandar
- 4.- Eficiente
- 5.- Verificable
- 6.- Funcional

A continuación se explica que significa cada uno de los puntos anteriores.

### Confiable

Para que un programa sea confiable es necesaria una concordancia entre los requerimientos con los que está concebido el programa y el comportamiento real de éste, considerando las posibles violaciones a sus reglas de operación y su comportamiento en tales casos.

### Modular

Que un programa sea modular implica el aislar las funciones que realiza en bloques diferenciados, cada uno de los cuales debe contener información para su lectura y con ello poder realizar una modificación al bloque, evitando así tener que modificar otros bloques del sistema.

## Estandar

La estandarización es una característica a la que debe tenderse siempre en el caso de diseñar un programa; implica básicamente la independencia del programa respecto al sistema físico de procesamiento que se utilice, es decir, que funcione con equipos distintos. Este criterio en la actualidad no siempre es posible satisfacerlo ya que los componentes electrónicos van ganando en complejidad y aumentando sus funciones respecto a sistemas más antiguos.

## Eficiente

El programa debe utilizar en forma adecuada la capacidad del sistema, utilizando las posibilidades máximas de la circuitería (hardware); ésto parece oponerse en cierto modo al criterio anterior, ya que al estandarizar reducimos la eficiencia del nuevo sistema, puesto que no será posible hacerlo compatible con otros equipos ya existentes.

## Verificable

El programa debe tener una estructura lógica y simple, la cual nos debe permitir localizar fácilmente sus diversas funciones y permitir el análisis y la prueba de su comportamiento, es decir, permitirá la prueba operativa del sistema a cualquier nivel de jerarquía.

## Funcional

Implica que el programa de operación debe poder comunicarse con el usuario, permitiendo un control adecuado por parte de éste.

## PROCESO DE DESARROLLO

El proceso de desarrollo que se emplea para diseñar un programa estructurado, consta de seis puntos principales:

- 1.- Analizar requerimientos
- 2.- Especificación funcional
- 3.- Diseño
- 4.- Programación
- 5.- Pruebas
- 6.- Aceptación

### Analizar requerimientos

En esta primera etapa de desarrollo se busca entender el problema desde el punto de vista del usuario y el mercado, esto es, obtener las características que debe satisfacer el programa.

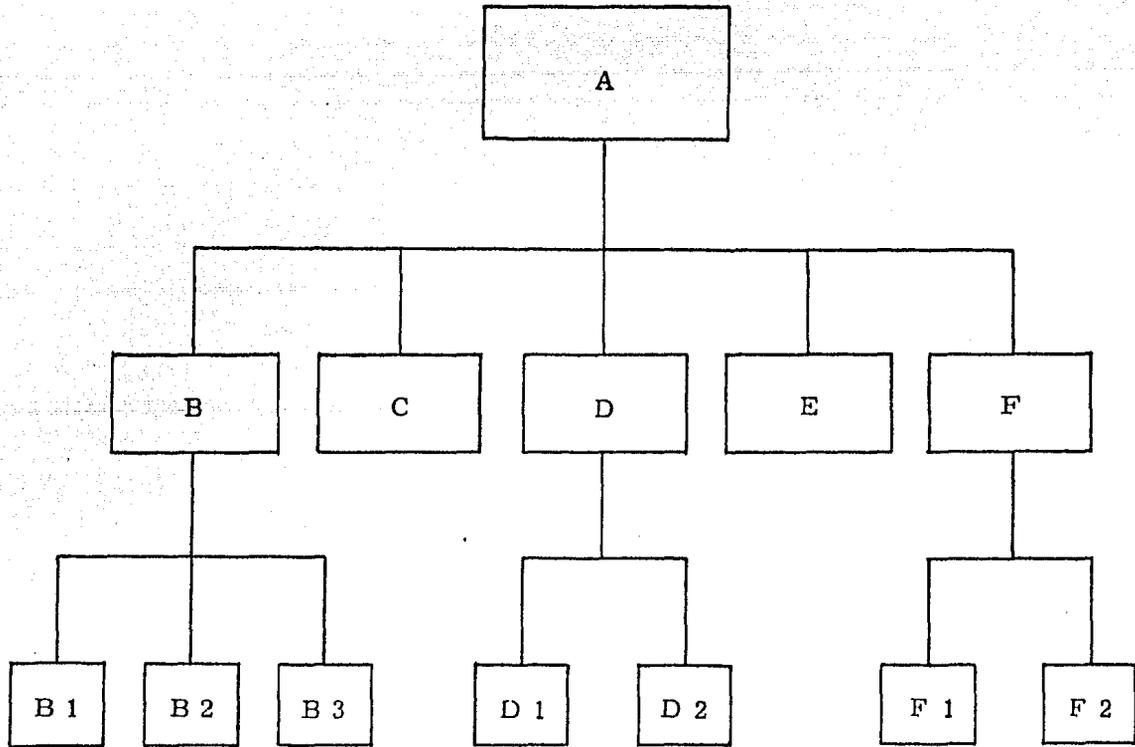
En esta etapa, además, es necesario documentar modularmente el programa descomponiendo el problema en sus partes básicas.

### Especificación funcional

En esta segunda etapa del desarrollo es necesario transformar las especificaciones del problema en una descripción de funciones implementables, definiendo que se va a realizar en cada una de ellas. Es necesario efectuar nuevamente un proceso de análisis de los puntos básicos del problema, para localizar desviaciones o precisar puntos indeterminados.

### Diseño

En esta etapa del proceso de desarrollo se define la arquitectura del sistema asignando cada función a un módulo determinado; los módulos así generados presentan el aspecto que se muestra en la página siguiente.



ESTRUCTURACION MODULAR DE UN PROGRAMA

El programa a su nivel más alto debe manejar, idealmente, siete rutinas, con una variación máxima de dos, o sea, con cinco rutinas como mínimo adecuado o nueve rutinas como máximo adecuado.

Estos módulos deben estar interrelacionados entre sí de una manera simple, entendiéndose por ésto que tengan una sola entrada con llamadas directas a subrutinas y con una sola salida. Los módulos deben estar definidos en términos de sus características de entrada y salida.

### Programación

En esta cuarta etapa de desarrollo debemos transformar el diseño en algoritmos, es decir, debemos especificar la forma en que cada módulo va a desempeñar su función. Además, en esta etapa debemos proceder a codificar el algoritmo, empleando una organización consistente, la cual debe incluir en el encabezado de cada módulo, una descripción de su objetivo, las entradas que tiene, las salidas que genera, los datos que emplea y las subrutinas que utiliza.

### Pruebas

En esta fase se verifica el funcionamiento del equipo, esto es, los programas deben actuar correctamente y desempeñar las funciones que tienen asignadas en el sistema. Las pruebas se realizan siempre en forma jerárquica.

### Aceptación

En esta última fase se debe tener el sistema funcionando en conjunto y además, tener un manual de operación del sistema.

## ANALISIS DE REQUERIMIENTOS

El equipo de suministro de gasolina debe cumplir con las características de funcionamiento de los equipos existentes en la actualidad, que son:

- 1.- Control de flujo desde la boquilla de la manguera
- 2.- Control de inicio para activar el motor
- 3.- Control para poner en cero el exhibidor
- 4.- Exhibidor que indique costo y consumo

Además de estos puntos nuestro sistema deberá tener un teclado con el que se pueda programar el consumo que se va a realizar, y deberá contar con una serie de botones para ordenar ciertas actividades como son, activar al sistema, programar consumo, cortar suministro, etc.

Con esta serie de condiciones se propone constituir un tablero de control con los siguientes elementos que se mencionan a continuación:

- 1.- Panel de comandos
- 2.- Panel de programación
- 3.- Exhibidores luminosos

### Panel de comandos

El panel de comandos consta de varias teclas, colocadas en la parte inferior del tablero de control, que son:

- 1.- Iniciar proceso (tecla A)
- 2.- Programación de caudal (tecla B)
- 3.- Programación de costo (tecla C)
- 4.- Activación del sistema (tecla D)
- 5.- Reactivación del sistema (tecla E)
- 6.- Verificación de datos o corte de suministro (tecla F)

### Iniciar proceso (tecla A)

Prepara al sistema para ser utilizado por el usuario, realizando varias acciones que son, colocar en "cero" los exhibidores

luminosos, borrar los registros de datos programados y resultados obtenidos por el usuario anterior, y finalmente habilitar a los demás comandos para ser utilizados por el nuevo usuario del sistema.

#### Programación de caudal (tecla B)

Con este comando se indica al sistema de suministro, que se desea programar cierta cantidad de litros.

#### Programación de costo (tecla C)

Con este comando se indica al sistema de suministro que se desea obtener una cantidad de gasolina equivalente a una determinada cantidad de dinero.

#### Activación del sistema (tecla D)

Con este comando se activa el motor eléctrico que maneja a la bomba de suministro de gasolina y se registran los datos programados (si es que los hay) y posteriormente va monitoreando el flujo de gasolina hasta llegar a la condición impuesta por el usuario o hasta que se le indique, desactivando a continuación el motor eléctrico.

#### Reactivación del sistema (tecla E)

Este comando permite activar el sistema de suministro de gasolina después de haber sido desactivado, sin que se pierdan los resultados que se tenían hasta ese momento, quedando inhabilitados los comandos para programar litros o costo.

#### Verificación de datos o corte de suministro (tecla F)

Cuando el sistema no está suministrando gasolina este comando nos permite ver la cantidad y el costo de la gasolina que se ha programado; además nos muestra el contenido del registro de resultados. Cuando el sistema está suministrando gasolina este comando nos permite terminar con la operación.

## Panel de programación

El panel de programación está constituido por un teclado con diez botones numerados del cero al nueve.

En el panel de programación el usuario ordena la cantidad de litros o la cantidad de pesos que desea, debiendo introducir un número entero de cuatro cifras como máximo.

## Exhibidores luminosos

Dos grupos de exhibidores permiten ver la cantidad de litros y pesos suministrados al usuario o programados por él.

Cada grupo de exhibidores consta de cinco dígitos. El exhibidor superior muestra la cantidad de litros, mientras que el inferior la cantidad de pesos.

Los exhibidores luminosos permiten también visualizar diversos mensajes en caso de una utilización errónea del sistema.

## Modos de operación

El sistema de suministro de gasolina puede ser operado de dos formas diferentes que son:

- 1.- Modo directo
- 2.- Modo programado

### Modo directo

Al ser operado de esta forma, el sistema de suministro de gasolina actúa del mismo modo en que operan los actuales sistemas de suministro, esto es, se activa el sistema y se controla el flujo de gasolina manualmente, o bien se espera a que el tanque de gasolina se llene.

## Modo programado

Al ser operado en esta forma, el sistema es programado con un valor de caudal o un costo, dejando de operar precisamente cuando dicho valor programado es alcanzado.

## ESPECIFICACION FUNCIONAL

Las funciones que debe realizar el sistema de suministro de gasolina son las siguientes:

- 1.- Iniciar proceso
- 2.- Programación de caudal
- 3.- Programación de costo
- 4.- Activación y supervisión del sistema
- 5.- Reactivación y supervisión del sistema
- 6.- Verificación de registros

El usuario tiene a su disposición las seis teclas del panel de comandos y con ellas se controlan los diferentes programas a los que tiene acceso.

La primera función a la que tiene acceso es la representada por el comando Iniciar proceso (tecla A), con el que se logra el borrado de los registros visuales o exhibidores que contenían las cantidades consumidas anteriormente. Además se borran los registros internos de datos y resultados, y se permite que el usuario pueda activar el sistema y programarlo libremente.

Finalmente debe indicar que el sistema está disponible. Esto lo indica colocando ceros en los exhibidores luminosos, tanto en el de litros como en el de dinero, apareciendo "00000", "00000" en los registros visuales.

Este control puede estar directamente a disposición del usuario o puede localizarse en un lugar aparte, donde un cajero puede supervisar el inicio y final de la operación, siendo él, el responsable del control.

La segunda función que realiza el sistema es la de aceptar un número si se presiona el comando adecuado en el teclado, interpretándolo como la cantidad total de litros que el usuario desea que le sean suministrados. Esta función es llamada mediante el comando para programar caudal (tecla B).

La tercera función que se puede realizar es como la anterior, y tiene la función de interpretar el dato numérico que se escriba en el teclado como una cantidad de dinero para que su equivalente en litros sea suministrado. A esta función se tiene acceso con el comando para programar costo (tecla C).

La cuarta función que tiene el sistema es la que se accesa mediante el comando de activación del sistema (tecla D), y sus funciones son registrar los datos que fueron programados. Muestra ceros en los dos registros visuales y procede a activar el motor eléctrico que maneja la bomba; además se monitorea el sistema para saber si la condición programada (en caso de existir) se ha cumplido.

También se observa si el sistema no se ha bloqueado a nivel de la boquilla de la manguera que el usuario controla; si este bloqueo dura un tiempo suficientemente prolongado el sistema se desconecta aunque la condición programada no se haya cumplido. Se tiene, además, la posibilidad de interrumpir la operación del sistema mediante el empleo del comando para cortar suministro (tecla F), que sirve también para verificar los registros de datos y resultados cuando el sistema no está operando.

La quinta función que realiza el sistema es la reactivación. Se tiene acceso a ella con el comando para reactivación del sistema (tecla E), y sólo debe ser empleado después de haber cortado el suministro de gasolina; al utilizar este comando el sistema operará en modo directo únicamente.

La sexta función que realiza el sistema es accesada con el comando de verificación de datos y resultados, siendo utilizada cuando el sistema no está suministrando gasolina y se desea consultar el contenido de los registros de programación y resultados, apareciendo en los registros visuales la cantidad programada, si es que la hay, y posteriormente aparece el contenido de los registros de resultados. Este comando es solicitado con la tecla F.

Cuando el sistema está operando el oprimir este comando provoca que el sistema termine de funcionar.

## DISEÑO

El programa que controlará el sistema de suministro estará estructurado de la siguiente forma; el programa principal controlará siete subrutinas que desempeñarán las funciones siguientes:

- 1.- Supervisar teclado de comandos (SUPERV)
- 2.- Inicializar el sistema (INICIA)
- 3.- Programación de litros (LITROS)
- 4.- Programación de costo (DINERO)
- 5.- Activación y monitoreo (ACTIVA)
- 6.- Reactivación y monitoreo (REACTI)
- 7.- Verificación de resultados (VERIFI)

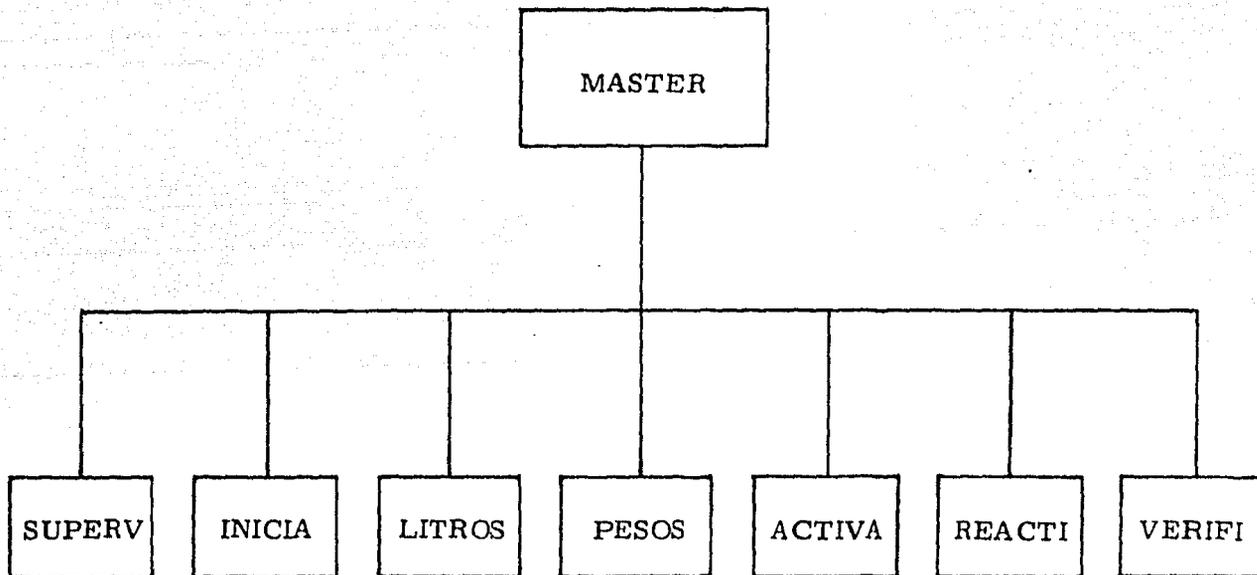
Cada uno de estos módulos es independiente de los demás.

En la página siguiente se muestra la estructura del programa con sus siete subrutinas básicas, las cuales también realizan llamadas a subrutinas de más bajo nivel, como son las subrutinas de apoyo y las subrutinas de utilería.

Las subrutinas de apoyo se requieren varias veces en la ejecución de los subprogramas principales y tienen la característica de emplear al menos una subrutina de utilería.

Las subrutinas de utilería tienen la función de realizar una operación básica, como puede ser enviar un mensaje al usuario, realizar una multiplicación, transformar un número de sistema decimal a sistema hexadecimal, etc.

El programa principal (MASTER) realiza las siguientes funciones:



ESTRUCTURACION DEL PROGRAMA

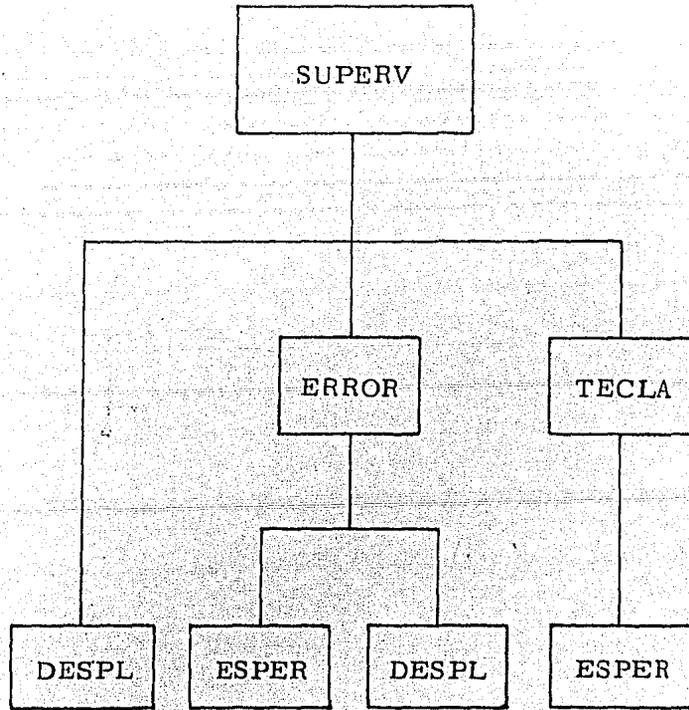
- 1.- Llamar a la subrutina de supervisión (SUPERV) la cual le proporcionará un código de comando cuando el usuario lo solicite.
- 2.- Al retomar el control cedido por la subrutina de supervisión (SUPERV) procede a realizar un análisis de la información proporcionada.
- 3.- De acuerdo al resultado de este análisis se transfiere el control a la subrutina elegida, la cual al finalizar su función le devuelve el control del sistema.
- 4.- Llama nuevamente a la subrutina de supervisión (SUPERV) repitiéndose nuevamente el ciclo.

#### Subrutina de supervisión (SUPERV)

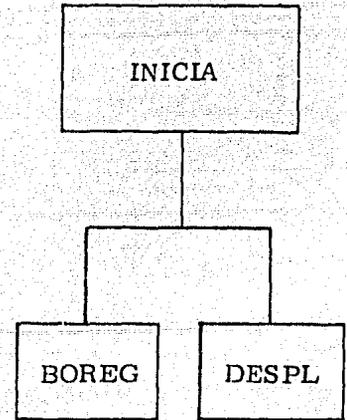
Tiene como función esperar un comando por parte del usuario, que al ser recibido a través del panel de comandos, lo transmite al programa principal (MASTER).

Las acciones que realiza son:

- 1.- Llama a la subrutina de apoyo para obtener datos (TECLA) y así recibir el código de la tecla de comando que ha sido oprimida por el usuario.
- 2.- Si esta tecla es un dato y no un comando, indicará al usuario su error y volverá a esperar un comando, retirando el mensaje de error y colocando en los registros visuales el mensaje "-----", "-----".
- 3.- Devuélve el control al programa principal (MASTER) enviándole el código de la tecla que ha sido oprimida.



SUBROUTINA DE INICIALIZACION



SUBROUTINA DE SUPERVISION

### Subrutina de inicialización (INICIA)

Este programa se ejecuta por orden del programa principal (MASTER) cada vez que el sistema de suministro de gasolina es empleado por un nuevo usuario.

Cuando se oprime el comando para iniciar proceso (tecla A), se llama a esta subrutina, que tiene por objeto poner al sistema con ceros en la pantalla y con ceros en los registros internos de datos y resultados, permitiendo además la subsecuente programación de litros o dinero y la activación del sistema.

La subrutina de inicialización (INICIA) hace uso de las subrutinas de utilería siguientes: subrutina para borrado de registros (BOREG) y de la subrutina para despliegue de mensajes (DESPL).

Las funciones que realiza son:

- 1.- Llama a la subrutina de utilería para borrado de registros (BOREG), para borrar las localidades de memoria donde se almacenan los datos y resultados.
- 2.- Llama a la subrutina de utilería para despliegue de mensajes (DESPL), para indicar ceros en ambos sistemas de despliegue, es decir, en exhibidor litros y exhibidor dinero.
- 3.- Habilita programación de litros y dinero con el indicador número 0, del registro de condiciones (BANDER).
- 4.- Devuelve el control al programa principal (MASTER).

### Subrutina de programación de caudal (LITROS)

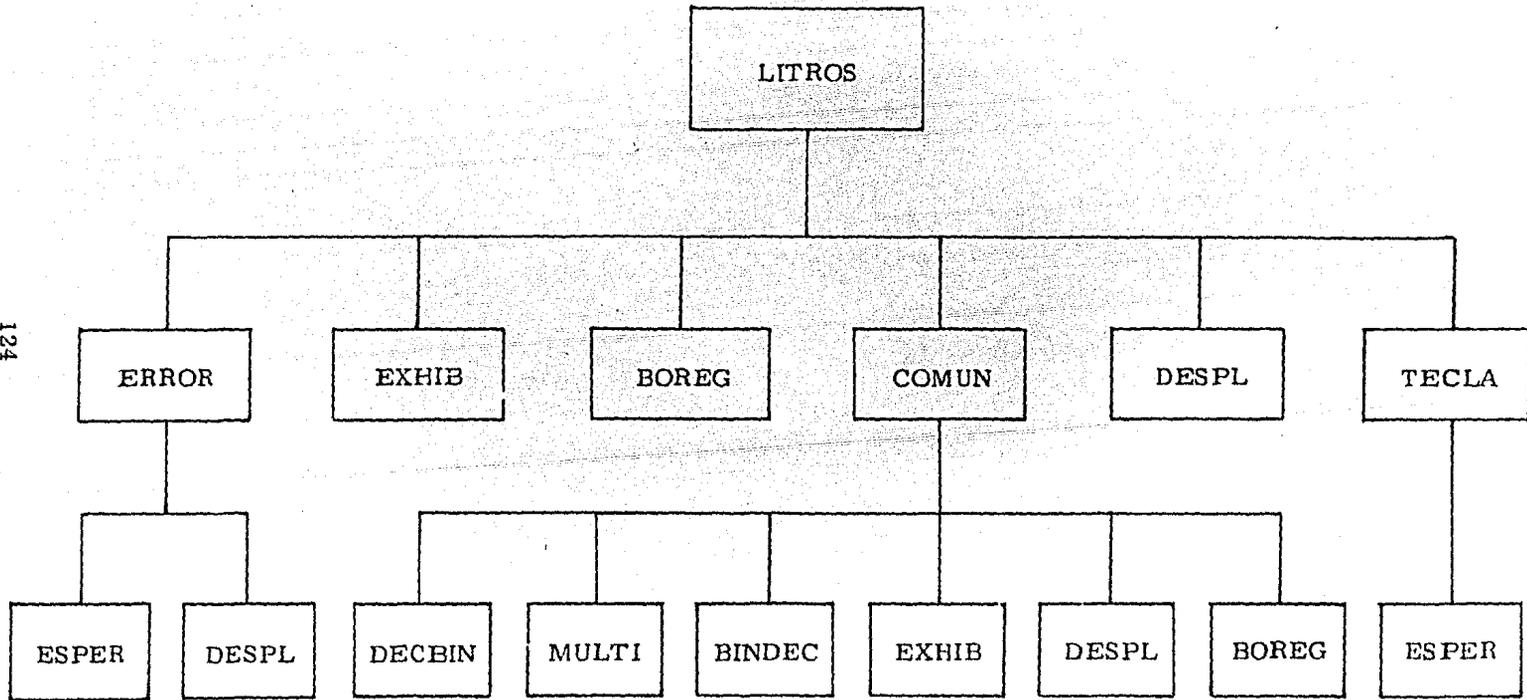
Esta subrutina es llamada por el programa principal (MASTER) y se ejecuta cuando se ha oprimido el comando de programación de caudal (tecla B), sirviendo para que el

usuario pueda programar el número de litros que desea.

Emplea las subrutinas de apoyo para obtener datos (TECLA), indicar condiciones (ERROR) y multiplicación decimal (COMUN); además emplea las subrutinas de utilidad borrar registros (BOREG), despliegue de resultados (EXHIB), y despliegue de mensajes (DESPL).

Las funciones que realiza son:

- 1.- Borrar registros de datos y resultados empleando la subrutina para borrado de registros (BOREG).
- 2.- Comprobación de secuencia analizando el indicador número 0 del registro de condiciones (BANDER).
- 3.- Despliega el mensaje "LLLLL", "-----", que indica listo para recibir datos e interpretarlos (como litros), en caso de que la condición número 0 del registro de condiciones (BANDER) sea la correcta.
- 4.- Desplegar el mensaje "EEEEEE", "EEEEEE", el cual indica error de comando si no está inicializado el sistema.
- 5.- Espera cuatro números que deben ser introducidos; en caso de introducir un carácter erróneo aparece "?????", "?????", y espera a que se introduzca nuevamente un número.
- 6.- Transforma los datos a binario y multiplica por la constante de costo para obtener el valor de la venta equivalente, y en caso de un costo mayor al máximo permitido se indicará ")))", ")))", capacidad sobrepasada y los datos no serán registrados; finalmente transforma los datos a decimal. Empleando la subrutina de multiplicación decimal (COMUN) para realizar estas acciones.
- 7.- Muestra los datos en los exhibidores de litros y dinero empleando la subrutina despliegue de resultados (EXHIB), y regresa el control al programa principal (MASTER).



SUBROUTINA PARA PROGRAMACION DE CAUDAL

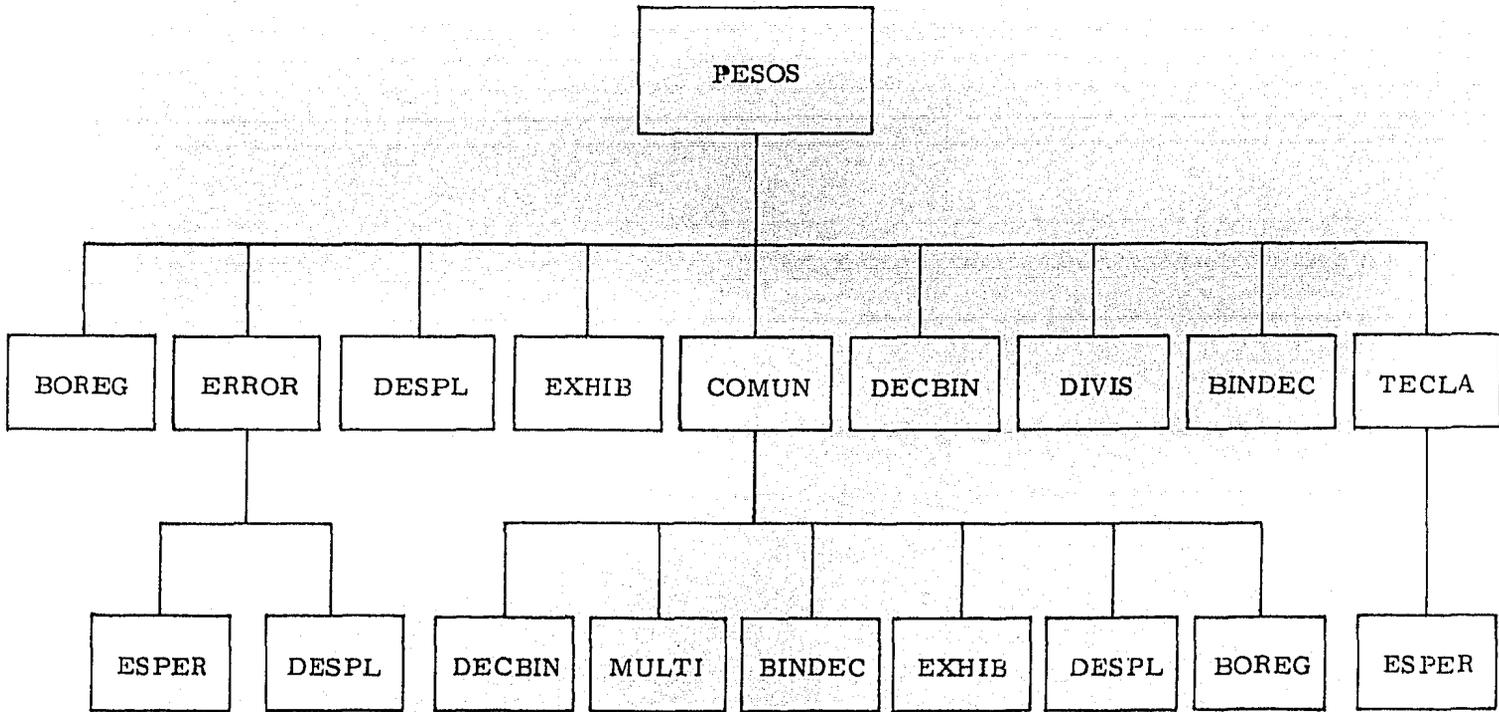
## Subrutina de programación de costo (DINERO)

Esta subrutina es llamada por el programa principal (MASTER) y se ejecuta cuando se ha oprimido el comando de programación de costo (tecla C), sirviendo para que el usuario pueda programar el costo en pesos que desea realizar para que le sean suministrados los litros de combustible correspondiente.

Emplea las subrutinas de apoyo para obtener datos (TECLA), indicar condiciones (ERROR), y multiplicación decimal (COMUN); además emplea las subrutinas de utilería, borrar registros (BOREG), despliegue de mensajes (DESPL), despliegue de resultados (EXHIB), transformación a hexadecimal (DECBIN), división hexadecimal (DIVIS) y transformación a decimal (BINDEC).

Las funciones que realiza son:

- 1.- Borrar registros de datos y resultados empleando para ello la subrutina para borrado de registros (BOREG).
- 2.- Comprobación de secuencia analizando el indicador número 0 del registro de condiciones (BANDER).
- 3.- Despliega el mensaje "\_\_\_\_\_", "PPPPP", que indica listo para recibir datos (e interpretarlos como dinero), en caso de que la condición número 0 del registro de condiciones (BANDER) sea la correcta.
- 4.- Desplegar el mensaje "EEEE", "EEEE" el cual indica error de comando, transfiriendo el control al programa principal (MASTER).
- 5.- Espera cuatro números que deben ser introducidos y en caso de introducir un carácter erróneo aparecerá el mensaje "????", "????", esperando el sistema la reintroducción del número adecuado.



SUBROUTINA PARA PROGRAMACION DE COSTO

- 6.- Transforma los datos a binario para que el procesador pueda operar con ellos.
- 7.- Divide por la constante de costo para obtener los litros correspondientes y aproxima al valor entero inmediato.
- 8.- Ajusta el valor correspondiente de costo empleando para ello la subrutina de multiplicación decimal (COMUN).
- 9.- Muestra los datos en los exhibidores de caudal y costo, empleando la subrutina despliegue de resultados (EXHIB), y devuelve el control al programa principal (MASTER).

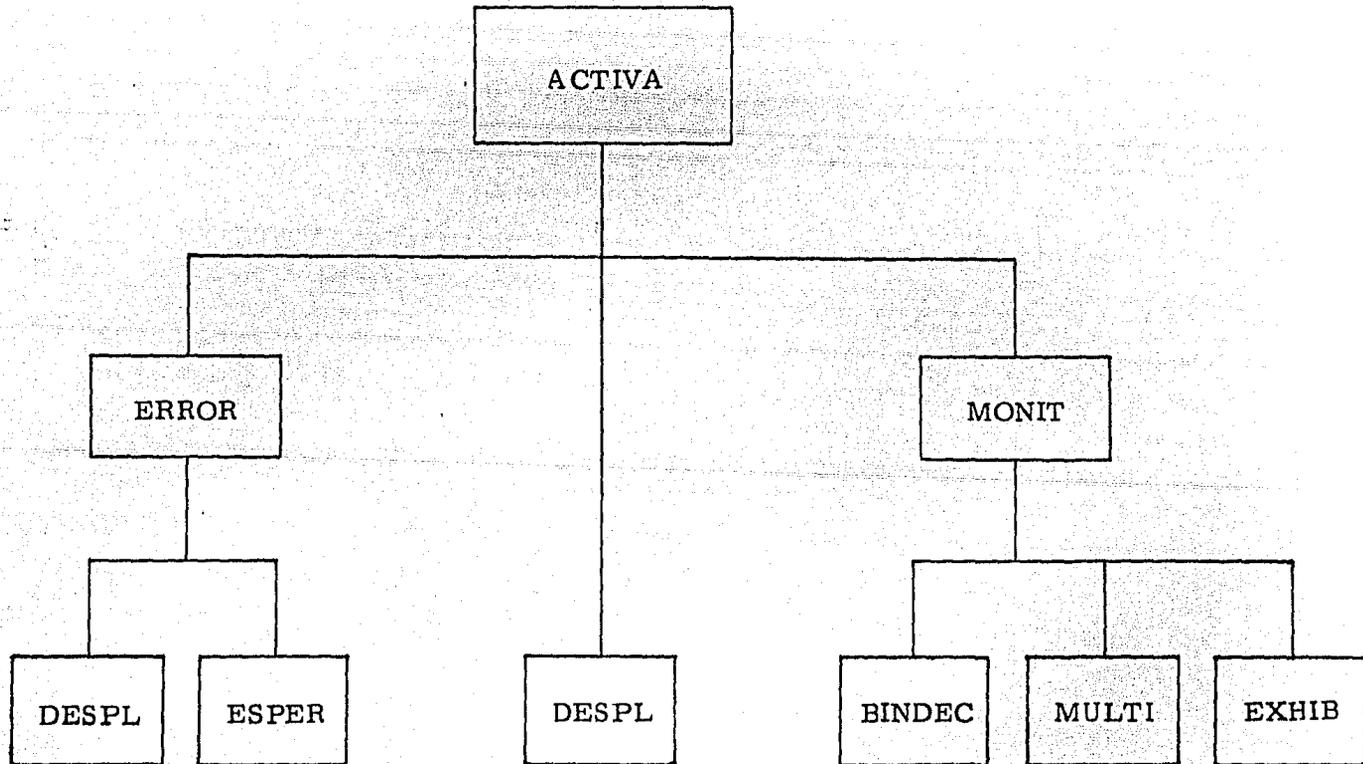
#### Subrutina de activación y monitoreo (ACTIVA)

Esta subrutina activa, supervisa y calcula el consumo de combustible, actualizándolo constantemente y mostrándolo al usuario, tanto en caudal como en costo. Al finalizar o ser bloqueado el sistema procede a desconectar el motor finalizando el proceso.

Hace uso de las siguientes subrutinas de apoyo: control de operación (MONIT), indicar condiciones (ERROR); además emplea la subrutina de utilería para despliegue de mensajes (DESPL).

La función de esta subrutina es:

- 1.- Comprobación de inicio con el análisis del código número 0 del registro de condiciones (BANDER).
- 2.- Muestra el mensaje "00000", "00000", o bien el mensaje "EEEEEE", "EEEEEE", indicando error de comando, devolviendo el control al programa principal (MASTER).



SUBROUTINA DE ACTIVACION Y MONITOREO

- 3.- Bloquear inicialización con el código número 0 de (BANDER).
- 4.- Calcular número de interrupciones a contar.
- 5.- Activar el motor.
- 6.- Esperar interrupción; en caso contrario calcular el consumo y regresar al punto 6 hasta finalizar.
- 7.- Apagar el motor. (Los puntos 5, 6 y 7, se realizan con la subrutina para monitorear y controlar operación (MONIT)).
- 8.- Regresar al programa principal (MASTER).

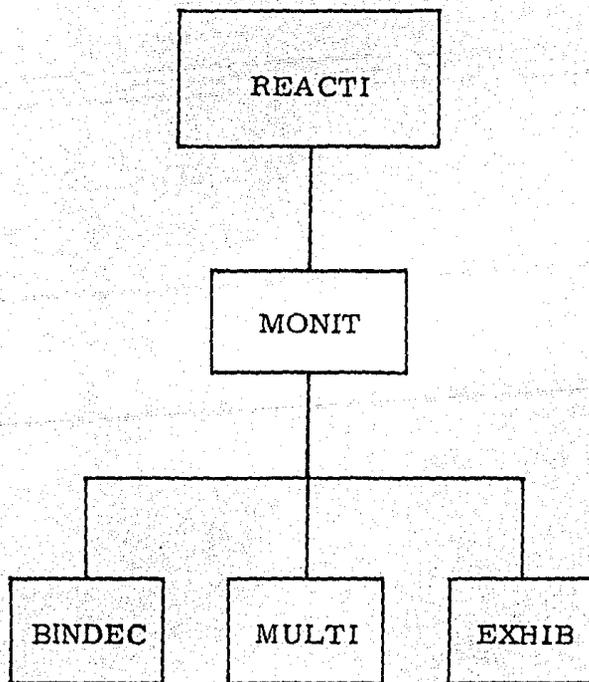
#### Subrutina de reactivación y monitoreo (REACTI)

Esta subrutina tiene por objeto permitir una reactivación del sistema de suministro de gasolina después de haber sido cortado el suministro.

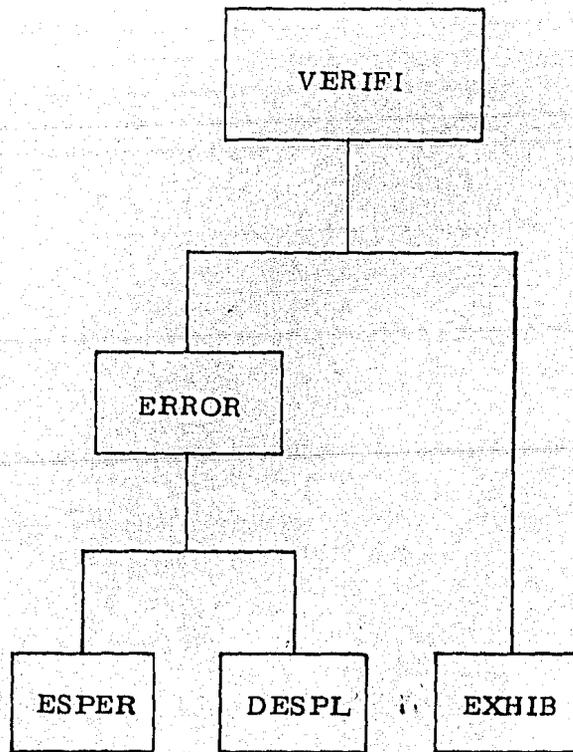
Al reactivar el sistema no se pierden los resultados almacenados en la memoria, pero no es posible hacer una nueva programación de datos operando el sistema en modo directo únicamente y hace uso de la subrutina de monitoreo y control de operación (MONIT).

La secuencia de operación es:

- 1.- Carga el contador de interrupciones para que el sistema opere en modo directo.
- 2.- Activa el motor.
- 3.- Espera interrupción; si no la hay calcula el consumo y regresa al punto 3 hasta finalizar.
- 4.- Apaga el motor y calcula el consumo. Las acciones de la secuencia 2, 3 y 4 son realizadas por la subrutina de monitoreo y control de operación (MONIT).



SUBROUTINA PARA REACTIVACION Y MONITOREO



SUBROUTINA PARA VERIFICACION DE REGISTROS

5.- Regresa el control al programa principal (MASTER).

### Subrutina de verificación de registros (VERIFI)

Esta subrutina permite al usuario la consulta de los registros donde están almacenados los datos programados y los resultados obtenidos hasta el momento de la consulta.

Hace uso de la subrutina de apoyo para indicar condiciones (ERROR), y de la subrutina de utilería para despliegue de resultados (EXHIB).

Las acciones que realiza son:

- 1.- Despliega el mensaje "dddd", "dddd", para indicar que será exhibido el contenido de los registros de datos programados.
- 2.- Muestra los datos programados.
- 3.- Despliega el mensaje "rrrrr", "rrrrr", para indicar que será exhibido el contenido de los registros de resultados obtenidos.
- 4.- Muestra los resultados obtenidos en el suministro de gasolina.
- 5.- Regresa el control al programa principal (MASTER).

### Subrutinas de apoyo

Las subrutinas de apoyo son empleadas por las subrutinas principales para realizar una función específica y son las siguientes:

- 1.- Subrutina para obtener datos (TECLA)
- 2.- Subrutina para indicar condiciones (ERROR)
- 3.- Subrutina de multiplicación decimal (COMUN)
- 4.- Subrutina para monitoreo y control de operación (MONIT)

### Subrutina para obtener datos (TECLA)

Esta subrutina tiene la función de analizar el estado en que se encuentra el teclado del nanocomputador "Z80 Starter Kit" y reconocer cuando es oprimida una de las teclas por parte del usuario.

Las acciones que realiza esta subrutina son:

- 1.- Fija la matriz del teclado en nivel lógico bajo.
- 2.- Analiza inmediatamente el estado del teclado para ver si una tecla ha sido oprimida; si no lo ha sido regresa al punto 1.
- 3.- Analiza fila por fila el teclado para definir la posición de la tecla oprimida.
- 4.- Espera que el usuario suelte la tecla para continuar, empleando la subrutina de utilería para tiempo de espera (ESPER).
- 5.- Compara con una tabla para encontrar el código hexadecimal correspondiente a la tecla.
- 6.- Regresa el control a la subrutina solicitante guardando el código obtenido en el acumulador.

### Subrutina para indicar condiciones (ERROR)

Esta subrutina transmite una cierta información y la mantiene por un cierto tiempo en el Sistema de despliegue de información.

Hace uso de las subrutinas de utilería para tiempo de espera ( ESPER ), y despliegue de mensajes (DESPL).

El código a desplegar debe ser cargado previamente en el registro C del microprocesador.

Las acciones que realiza esta subrutina son:

- 1.- Carga el código de despliegue para ser mostrado en los exhibidores de costo y caudal.
- 2.- Despliega el código llamando a la subrutina de despliegue de información (DESPL).
- 3.- Espera un cierto tiempo y regresa el control a la subrutina solicitante.

### Subrutina para multiplicación decimal (COMUN)

Tiene como función realizar una multiplicación decimal entre un valor que deberá estar cargado previamente en memoria y una constante hexadecimal, también almacenada en memoria; la multiplicación se realiza en forma hexadecimal y el resultado se transforma a decimal nuevamente.

Utiliza las rutinas de utilidad transformación a hexadecimal (DECBIN), multiplicación hexadecimal (MULTI), transformación a decimal (BINDEC), despliegue de resultados (EXHIB), borrado de registros (BOREG), y despliegue de mensajes (DESPL).

Las acciones que realiza esta subrutina son:

- 1.- Transforma el dato a hexadecimal (este dato debe ser cargado previamente).
- 2.- Multiplica por la constante de costo de gasolina (en hexadecimal).
- 3.- Analiza si la multiplicación no excede la capacidad de la memoria, e indica error y borra registros en caso de suceder esto, y pasa (siendo necesario volver a programar el sistema con un valor menor) al punto 5 directamente.
- 4.- Transforma el resultado a decimal cargándolo en la memoria de costo.
- 5.- Regresa el control a la subrutina solicitante.

## Subrutina para controlar operación (MONIT)

Esta subrutina tiene que controlar la operación del sistema de suministro de gasolina, activando y desactivando el motor eléctrico, monitoreando el flujo de caudal y probando si no hay una solicitud de interrupción de suministro.

Utiliza las subrutinas de utilería, multiplicación hexadecimal (MULTI), transformación a decimal (BINDEC), y despliegue de resultados (EXHIB).

Las acciones que realiza son:

- 1.- Programa el contador de eventos con el número de ciclos a realizar.
- 2.- Activa el motor eléctrico.
- 3.- Monitorea el sistema y analiza si:
  - a) Existe interrupción generada por el Z80 CTC y procede a calcular consumo.
  - b) Existe solicitud de corte de suministro.
  - c) Se ha bloqueado el suministro de gasolina.
- 4.- Si la condición (a) se cumple, calcula consumo y decrementa contador de ciclos y regresa al principio del punto 3, pero si se han terminado los ciclos de conteo apaga el motor eléctrico y finaliza.
- 5.- Si se solicita interrupción o se bloquea el suministro (a nivel de la boquilla de la manguera), calcula consumo y apaga el motor eléctrico.
- 6.- Regresa el control a la subrutina solicitante.

## Subrutinas de utilería

Las subrutinas de utilería que son utilizadas son las siguientes:

- 1.- Subrutina de tiempo de espera (ESPER)
- 2.- Rutina de servicio a interrupción (INTER)
- 3.- Subrutina para despliegue de mensajes (DESPL)
- 4.- Subrutina para borrado de registros (BOREG)
- 5.- Subrutina de transformación a hexadecimal (DECBIN)
- 6.- Subrutina de multiplicación hexadecimal (MULTI)
- 7.- Subrutina de transformación a decimal (BINDEC)
- 8.- Subrutina para despliegue de registros (EXHIB)
- 9.- Subrutina de división hexadecimal (DIVIS)

### Subrutina para tiempo de espera (ESPER)

Esta subrutina genera un retardo de 20 ms en la ejecución de un programa. Se afectan los registros H y L al ser utilizada y realiza las siguientes acciones:

- 1.- Carga al registro HL con un número a partir del cual se efectuará una cuenta descendente.
- 2.- Decrementa hasta cero al registro HL.
- 3.- Si HL no es cero regresa al punto 2;  
Si HL es igual a cero continua.
- 4.- Regresa el control a la subrutina solicitante.

### Rutina de servicio a interrupción (INTER)

Esta rutina es llamada cada vez que el Z80 CTC genera una interrupción, y tiene como única función fijar el bit número cinco del registro de condiciones (BANDER) a uno, cada vez que es utilizada.

Esta rutina modifica únicamente al acumulador, y realiza lo siguiente:

- 1.- Carga el acumulador con el registro de banderas (BANDER).
- 2.- Fija el bit número cinco del acumulador a uno.
- 3.- Almacena el contenido del acumulador en el registro de banderas (BANDER).
- 4.- Regresa al programa en ejecución al ocurrir la interrupción empleando la instrucción RETI.

### Subrutina para despliegue de mensajes (DESPL)

Esta subrutina tiene por objeto enviar un mensaje al usuario que deberá aparecer en los exhibidores de caudal y costo.

El código del mensaje a desplegar en el exhibidor de caudal, deberá cargarse previamente en el registro C del microprocesador y en el registro D el código a desplegar en el exhibidor de costo del Sistema de despliegue de información.

Esta subrutina modifica los registros B, H y L del microprocesador.

Las acciones de la subrutina son:

- 1.- Carga contador de exhibidores con cinco.
- 2.- HL indicará la localidad del primer exhibidor de caudal.
- 3.- Se cargará en la localidad del exhibidor indicado por HL el código a desplegar (almacenado en el registro C).
- 4.- Se incrementará en uno el contenido del registro par HL, para así apuntar al siguiente exhibidor y se decrementará en uno el contador de exhibidores.

- 5.- Si se terminó con el quinto exhibidor continuar; en caso contrario, regresar al punto 3.
- 6.- HL indicará la localidad del primer exhibidor de costo.
- 7.- Se cargará el contador de exhibidores con el número cinco.
- 8.- Se cargará en la localidad del exhibidor indicado por HL el código a desplegar que está almacenado en el registro D.
- 9.- Se indicará el siguiente exhibidor y se decrementará en uno el contador de exhibidores.
- 10.- Si se terminó de cargar el quinto exhibidor continuar; en caso contrario regresar al punto 8.
- 11.- Regresar el control a la subrutina solicitante.

#### Subrutina de borrado de registros (BOREG)

Esta subrutina tiene por objeto borrar el contenido de las memorias (esto es, poner en cero), donde se encuentran almacenados los datos programados por el usuario y los resultados obtenidos en el suministro de gasolina.

Al utilizar esta subrutina se modifica el contenido de los registros A, B, D, y E; las acciones que se realizan son:

- 1.- Se coloca un apuntador (el DE) en la primera localidad a ser borrada.
- 2.- Se carga un contador de eventos con el número de localidades a ser borradas.

- 3.- Se pone el valor de cero en el acumulador.
- 4.- Se carga el contenido del acumulador en la memoria indicada por DE.
- 5.- Se incrementa en uno al apuntador DE.
- 6.- Se decrementa al contador B; si no ha terminado de contar regresa al punto 4; en caso contrario continua.
- 7.- Regresa el control a la subrutina solicitante.

#### Subrutina para transformación a hexadecimal (DECBIN)

Esta subrutina tiene por objeto transformar un número decimal de cinco dígitos a su equivalente hexadecimal, siendo necesario que los datos sean indicados previamente por IX+5, IX+4, IX+3, IX+2, IX+1; el resultado queda cargado en el registro par HL. (El número decimal a transformar deberá ser menor de 65 536; las subrutinas que emplean esta conversión solo transforman números en el rango de 0 a 9999.)

Se modifica el contenido de HL, BC, DE e IX en esta subrutina. Realiza las acciones siguientes:

- 1.- Carga contador de números decimales a transformar.
- 2.- Borra registro HL.
- 3.- Multiplica el resultado por diez.
- 4.- Carga el número en el registro par DE.
- 5.- Suma DE y HL.
- 6.- El registro IX apunta al siguiente número decimal a ser transformado, esto es, el registro IX es decrementado.

- 7.- Decrementa contador de números a transformar (decimales) y si no es cero regresa al punto 3; en caso contrario continua.
- 8.- Regresa el control a la subrutina solicitante.

### Subrutina de multiplicación hexadecimal (MULTI)

Esta subrutina realiza una multiplicación hexadecimal entre dos números de 2 bytes cada uno, dejando un resultado de hasta 4 bytes en los registros D, E, H, L.

El multiplicando deberá almacenarse previamente en el par de registros DE y el multiplicador también deberá ser cargado previamente en el registro par BC.

Esta subrutina modifica los registros BC, DE, HL y A y realiza las siguientes acciones:

- 1.- Coloca un contador de 16 iteraciones implementado en el acumulador.
- 2.- Coloca ceros en el registro par HL.
- 3.- Se realiza un corrimiento de DE HL siendo el registro DE el más significativo una localidad binaria a la izquierda, guardando el estado de la bandera de acarreo en ambos corrimientos.
- 4.- Si existe acarreo en HL pasar al punto 5; y si no existe analizar si hubo acarreo en DE. Si no existe acarreo en DE pasar al punto 6, y si existe sumar HL, y BC: si esta suma tiene acarreo incrementar DE y pasar al punto 6; si la suma no tiene acarreo pasar a 6 directamente.
- 5.- Si existe acarreo en HL se incrementa en uno el registro par DE y se procede a revisar si hubo acarreo en DE durante la secuencia 3; si no existió acarreo en DE pasar al punto 6; y si existió, sumar HL y BC; si esta suma tiene acarreo incrementar DE y pasar al punto 6; si la suma no tiene acarreo pasar al punto 6 directamente.

- 6.- Decrementar contador de iteraciones y ver si es cero; si lo es regresar el control a la subrutina solicitante; en caso contrario pasar al punto 3.

### Subrutina para transformación a decimal (BINDEC)

Esta subrutina tiene por objeto transformar un número hexadecimal de cuatro bytes a su equivalente decimal; el número hexadecimal debe ser cargado previamente en el registro HL y el resultado es almacenado en cinco localidades de memoria, apuntadas por IX+5, IX+4, IX+3, IX+2 e IX+1.

Esta subrutina utiliza una tabla de potencias de diez y se realiza una substracción sucesiva para analizar cuantas potencias de diez se requieren para igualar el residuo, con lo cual se encuentra el dígito decimal correspondiente al número hexadecimal en proceso.

Las acciones que realiza esta rutina son:

- 1.- Indicar tabla de potencias de diez con el apuntador IY.
- 2.- Colocar contador de dígito en cero.
- 3.- Cargar potencia de diez apuntada por IY+1 e IY+0.
- 4.- Borrar indicador de acarreo.
- 5.- Restar la potencia al número hexadecimal; en caso de ser menor la potencia de 10 incrementa el contador de dígitos y pasa al punto 4 nuevamente; en caso de ser mayor la potencia de diez prosigue.
- 6.- Sumar potencia de diez para restaurar el número hexadecimal.
- 7.- Almacenar en memoria de resultados.

- 8.- Apuntar a la siguiente potencia de diez incrementando en 2 el contador IV.
- 9.- Apuntar a la localidad de memoria para resultados inmediata inferior, decrementando apuntador IX.
- 10.- Si se ha terminado, regresar a subrutina solicitante, en caso contrario regresar al punto 2.

### Subrutina de despliegue de resultados (EXHIB)

Esta subrutina tiene como función transferir los datos y resultados almacenados en las memorias del sistema, a los exhibidores luminosos de caudal y costo.

Esta subrutina deberá ser cargada previamente con el código 74H para exhibir los registros de datos, o con el código 82H para mostrar los registros de resultados; estos códigos deberán ser cargados en el acumulador.

Modifica y utiliza los registros A, BC, DE y HL y realiza las siguientes funciones:

- 1.- Fija el apuntador DE, que indicará el primer exhibidor luminoso de caudal.
- 2.- Carga el código de operación para despliegue de datos o resultados.
- 3.- Fija un contador de iteraciones en la memoria auxiliar (MEMAUX).
- 4.- Fija el apuntador HL en la tabla de códigos para despliegue en exhibidores.
- 5.- Se carga el registro C con el número a desplegar y se le suma al apuntador de la tabla de códigos HL.

- 6.- Se transfiere el código almacenado en HL a la memoria de control del exhibidor seleccionado.
- 7.- Se apunta al siguiente exhibidor y se decrementa el contador de iteraciones localizado en la memoria auxiliar (MEMAUX).
- 8.- Si el contador de iteraciones es cero regresar el control a la subrutina solicitante; en caso contrario continuar.
- 9.- Si el contador de iteraciones no ha tomado el valor de 5 pasar al punto 11.
- 10.- Si el contador de iteraciones es cinco modificar el apuntador DE para que indique el primer exhibidor de costo y continuar.
- 11.- Regresar al punto 4.

#### Subrutina de división hexadecimal (DIVIS)

Esta subrutina realiza una división hexadecimal entre un dividendo de cuatro bytes y un divisor de dos bytes.

El dividendo deberá ser almacenado previamente en los registros H, L, D, E, y el divisor en el registro doble BC. Al terminar de ejecutarse la división el cociente de dos bytes quedará almacenado en los registros DE y el residuo en los registros HL. (Si el cociente no se puede almacenar en dos bytes el resultado de la división será erróneo; sin embargo en esta aplicación los bytes más significativos del dividendo siempre son cargados con cero, siendo en el peor de los casos, una división entre uno, siempre se tendrá por lo tanto un cociente de dos bytes como máximo).

Esta subrutina modifica y utiliza los registros A, DE, BC, HL, y realiza las siguientes funciones:

- 1.- Coloca un contador para 16 iteraciones implementado en el acumulador.
- 2.- Realiza un corrimiento de los datos del dividendo almacenados en DE y HL una localidad binaria a la izquierda.
- 3.- Si en el corrimiento a la izquierda del par de registros DE, no se genera un acarreo pasar al punto 4; en caso de generarse el acarreo incrementar el contenido del registro HL y continuar (esto es, el bit de más alto orden de DE pasa a la posición de más bajo orden de HL).
- 4.- Borrar bandera de acarreo y sustraer HL y BC, colocando un uno en el bit menos significativo del registro DE, o sea, el cociente.
- 5.- Si la resta de HL y BC es positiva pasar al punto 6 y en caso contrario poner cero en el bit menos significativo del registro E, es decir, se restablece un cero en el cociente y se restaura el dividendo sumando los registros HL y BC.
- 6.- Decrementar el contador de iteraciones y analizar si es cero; si lo es, regresar el control a la subrutina solicitante; en caso contrario regresar al punto 2.

## CODIFICACION

A continuación se realiza la codificación del programa. Se muestra en la primera columna la posición en memoria de la instrucción correspondiente, que aparece en la segunda columna, esta instrucción está codificada en hexadecimal; además se muestra la traducción de la instrucción correspondiente a lenguaje ensamblador, utilizándose cuatro pseudoinstrucciones que son: ORG indicando inicio de codificación, END indicando fin de codificación, DEFB definiendo un espacio en memoria para almacenar tablas o datos y EQU indicando el valor numérico de una etiqueta.

UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES  
CUAUTITLAN

TESIS

DISEÑO DE UN SISTEMA AUTOMATICO DE  
SUMINISTRO DE GASOLINA EMPLEANDO UN  
MICROPROCESADOR Z-80

Este programa fué realizado en código de máquina  
y está preparado para ser empleado en el Z-80  
STARTER SYSTEM.

ORG 2000 ; Inicio de Programa

; Programa MASTER

; Este programa tiene por objeto solicitar un coman-  
do al usuario del sistema, empleando para esto la  
subrutina SUPERV; al recibir este comando pro-  
cede a analizar y transferir el control a la sub-  
rutina elegida por el usuario.

; Este programa no afecta ningún registro y emplea  
las subrutinas SUPERV, INICIA, LITROS, DINERO,  
ACTIVA, REACTI y VERIFI.

2000	CD3B20	ETO	CALL	SUPERV	; subrutina de supervisión
2003	FE0A	CP	0AH		; ¿es comando de inicio?
2005	2816	JR	Z, TA0		; sí, ir a etiqueta TA0
2007	FE0B	CP	0BH		; ¿es comando litros?
2009	2817	JR	Z, TA1		; sí, ir a etiqueta TA1
200B	FE0C	CP	0CH		; ¿es comando pesos?
200D	2818	JR	Z, TA2		; sí, ir a etiqueta TA2
200F	FE0D	CP	0DH		; ¿es comando activa?
2011	2819	JR	Z, TA3		; sí, ir a etiqueta TA3
2013	FE0E	CP	0EH		; ¿es comando reactiva?
2015	281A	JR	Z, TA4		; sí, ir a etiqueta TA4
2017	FE0F	CP	0FH		; ¿es comando verifica?
2019	281B	JR	Z, TA5		; sí, ir a etiqueta TA5
201B	18E3	JR	ETO		; regresa al principio
201D	CD5420	TA0	CALL	INICIA	; subrutina INICIA
2020	18DE	JR	ETO		; regresa al principio

2022	CD6320	TA1	CALL	LITROS	:	subrutina LITROS
2025	18D9		JR	ET0	:	regresa al principio
2027	CDA A20	TA2	CALL	PESOS	:	subrutina DINERO
202A	18D4		JR	ET0	:	regresa al principio
202C	CD1121	TA3	CALL	ACTIVA	:	subrutina ACTIVA
202F	18CF		JR	ET0	:	regresa al principio
2031	CD3E21	TA4	CALL	REACTI	:	subrutina REACTI
2034	18CA		JR	ET0	:	regresa al principio
2036	CD4C21	TA5	CALL	VERIFI	:	subrutina VERIFI
2039	18C5		JR	ET0	:	regresa al principio

;

; Subrutina SUPERV

; Esta subrutina tiene como función

; solicitar un dato al usuario empleando

; para ello a la subrutina de apoyo

; TECLA, y comprobar que este dato

; sea un comando (número hexadecimal

; A, B, C, D, E, F) y regresar el con-

; trol a MASTER con el comando alma-

; cenado en el acumulador; en caso de

; no ser un comando el dato obtenido

; indicar error y volver al principio de

; la subrutina.

; Esta Subrutina hace uso de todos los

; registros modificando su contenido.

; Emplea las subrutinas de apoyo TECLA

; y ERROR y la subrutina de utilería

; DESPL

203B	CD6B21	ES1	CALL	TECLA	:	subrutina TECLA
203E	FE10		CP	10H	:	
2040	3005		JR	NC, ES2	:	
2042	FE0A		CP	0AH	:	
2044	3801		JR	C, ES2	:	
2046	C9		RET		:	regresa
2047	0E86		LD	C, 86	:	indicador de error
2049	CDD421	ES2	CALL	ERROR	:	subrutina ERROR
204C	0EBF		LD	C, BFH	:	indicador de
204E	SI		LD	D, C	:	disponibilidad
204F	CDA A22		CALL	DESPL	:	despliega
2052	18E7		JR	ES1	:	regresa al inicio

```

; Subrutina INICIA
; Esta subrutina realiza un borrado de
; los resultados anteriores almacena-
; dos en memoria, pone ceros en la uni-
; dad de despliegue y habilita al sistema
; para funcionar.
; Se emplean y modifican los registros
; BC, DE y HL, además del acumulador.
; Emplea las subrutinas de utilería BOREG
; y DESPL.

```

```

2054  CDBD22  EI1  CALL  BOREG      ; limpia resultados
2057  3E03    LD    A, 03H      ;
2059  327323  LD    (BANDER),A ; habilita comandos
205C  0EC0    LD    C, C0       ; carga ceros
205E                LD    D, C       ; para despliegue
205F  CDAA22  CALL  DESPL      ; despliega
2062  C9      RET          ; regresa

```

```

;
; Subrutina LITROS
; Esta subrutina permite al usuario programar
; la cantidad de litros que desea.
; Primero despliega el mensaje "LLLL"
; para indicar que está listo para recibir 4 da-
; tos, los procesa e indica el costo de la carga
; además de memorizar estos valores.
; Si se introduce un dato equivocado el sistema
; enviará el mensaje "?????" y esperará la in-
; troducción correcta del dato.
; Esta subrutina emplea la subrutinas de apoyo
; TECLA , COMUN , ERROR, y las subrutinas
; de utilería BOREG, EXHIB y DESPL.
; Se modifican los registros A, B, C, D, E, H y L.
;

```

```

2063  CDBD22  CALL  BOREG      ; limpia resultados
2066  3A7323  LD    A, (BANDER) ; comprueba -
2069  CB47    BIT    0, A      ; - inicialización
206B  282B    JR    Z, EL3
206D  0EC7    LD    C, C7H      ; código "L"
206F  16BF    LD    D, BFH      ; código "-"
2071  CDAA22  CALL  DESPL      ; envía mensaje
2074  0604    LD    B, 04H      ; contador en 4
2076  C5      EL1  PUSH  BC      ; proteger contador

```

2077	CD6B21	EL2	CALL	TECLA	; solicita número
207A	FE0A		CP	0AH	; ¿Es número?
207C	3020		JR	NC, EL4	; no, error
207E	117823		LD	DE, 2378	; sí, procesa -
2081	217723		LD	HL, 2377	; -esta información
2084	010400		LD	BC, 04H	
2087	EDB8		LDDR		
2089	327423		LD	(2374), A	
208C	3E74		LD	A, 74H	; despliega -
208E	CD2323		CALL	EXHIB	; - información
2091	C1		POP	BC	; restablece B
2092	10E2		DJNZ	EL1	; continua
2094	CDE021		CALL	COMUN	; calcula
2097	C9		RET		; regresa
2098	0E86	EL3	LD	C, 86H	; indicador -
209A	CDD421		CALL	ERROR	; - de bloqueo
209D	C9		RET		
209E	0EAC	EL4	LD	C, ACH	; indicador -
20A0	CDD421		CALL	ERROR	; - de error
20A3	3E74		LD	A, 74H	; restablecer
20A5	CD2323		CALL	EXHIB	;
20A8	18CD		JR	EL2	; repetir

;
  
; Subrutina DINERO
  
; Esta subrutina permite al usuario programar
  
; la cantidad de pesos que desea. Esta subru-
  
; tina despliega al principio el mensaje "PPPP"
  
; indicando que espera la cantidad de dinero que
  
; el usuario desea programar; si se introduce
  
; un comando en lugar del dato numérico el sis-
  
; tema responde "???" y espera la introducción
  
; correcta.
  
; Esta subrutina emplea las subrutinas de apoyo
  
; TECLA, COMUN, ERROR y las subrutinas de
  
; utilidad BOREG, DIVIS, EXHIB y DESPL.
  
; Se modifican los registros A, B, C, D, E, H y L.
  
;

20AA	CDBD22		CALL	BOREG	; limpia resultados
20AD	3A7323		LD	A, (BANDER)	; comprueba -
20B0	CB4B		BIT	0, A	; - inicialización
20B2	282B		JR	Z, EP3	;
20B4	0EBF		LD	C, BFH	; código "-"
20B6	168C		LD	D, 8CH	; código "P"

20B8	CDAA22		CALL	DESPL	; envía mensaje
20B9	0604		LD	B, 04H	; contador en 4
20BD	CS	EP1	PUSH	BC	; proteger contador
20BE	CD6B21	EP2	CALL	TECLA	; solicita número
20C1	FE0A		CP	0AH	; ¿es número?
20C3	3040		JR	NC, EP4	; no, error
20C5	117D23		LD	DE, 237D	; si, procesa -
20C8	217C23		LD	HL, 237C	; - información
20CB	010400		LD	BC, 04H	; ;
20CE	EDB8		LDDR		; ;
20D0	327923		LD	(2379), A	; ;
20D3	3E74		LD	A, 74), H	; despliega -
20D5	CD2323		CALL	EXHIB	; -información
20D8	C1		POP	BC	; restablece B
20D9	10E2		DJNZ	EP1	; decrementa B
20DB	DD217D23		LD	1X, 237D	; MEHDIN + S
20DF	CDC822		CALL	DECBIN	; a binario
20E2	228023		LD	(2380), HL	; almacena
20E5	EB		EX	DE, HL	; datos para -
20F6	210000		LD	HL, 00H	; - dividir
20E9	ED4B6F23		LD	BC, (6ASOL)	; costo gasolina
20ED	CD5823		CALL	DIVIS	; divide
21F0	EB		EX	DE, HL	; ;
21F1	227E23		LD	(237E)	; ;
21F4	DD217823		LD	1X, 2378	; ;
21F8	CDF622		CALL	BINDEC	; a decimal
21FB	CDE021		CALL	COMUN	; recalcula
21FE	C9		RET		; regresa
21FF	0E86	EP3	LD	C, 86H	; indicador -
2101	CDD421		CALL	ERROR	; - de bloqueo
2104	C9		RET		; ;
2105	0EAC	EP4	LD	C, ACH	; indicador de
2107	CDD421		CALL	ERROR	; - error
210A	3E74		LD	A, 74H	; ;
210C	CD2323		CALL	EXHIB	; restablece
210F	18AD		JR	EP2	; repetir

; ;  
; Subrutina ACTIVA  
; Esta subrutina tiene por objeto activar al  
; sistema de suministro de gasolina, para  
; esto emplea la subrutina de apoyo MONIT.  
; Además decide de acuerdo a los datos pro-  
; gramados si el sistema actua en modo di-  
; recto o modo programado, tiene además  
; protección para evitar su uso sin haber  
; sido inicializado el sistema.

```

;
; Esta subrutina hace uso de las sub-
; rutinas de apoyo MONIT y ERROR
; además emplea la subrutina de uti-
; leria DESPL.
; Se modifica el acumulador y los re-
; gistros BC, DE y HL
;

```

```

2111 3A7323          LD    A, (BANDER) ; comprueba
2114 CB47           BIT    0,A      ; inicialización
2116 2815           JR     Z,EA1     ;
2118 CB87           RES    0,A      ; bloquea comandos
211A 327323        LD    A, (BANDER) ;
211D 2A7E23        LD    HL, (237E) ; MEMLITH en HL
2120 AF            XOR    A      ; borra acarreo
2121 47             LD    B,A      ; carga 0 en B
2122 4F             LD    C,A      ; carga 0 en C
2123 E5             PUSH  HL      ; almacena HL
2124 ED42           SBC   HL, BC    ; resta
2126 D1             POP    DE      ; restablece HL en DE
2127 2810           JR     Z,EA2     ; salta a EA2
2129 CD1322        EA0   CALL  MONIT ; activa sistema
212C C9             RET          ; regresa
212D 0E86          EA1   LD    C, 86H ; indica -
212F CDD421        CALL  ERROR ; - condición
2132 0EBF          LD    C, BFH   ; de bloqueo
2134 51             LD    D, C    ; y poner
2135 CDAA22        CALL  DESPL  ; "----"
2138 C9             RET          ;
2139 11FFFF        EA2   LD    DE, FFFFH ; operación libre
213C 18EB          JR     EA0     ; activa

```

```

;
; Subrutina REACTI
; Esta subrutina tiene como función
; rehabilitar al sistema despues de
; haber operado ya sea en modo
; directo o en modo programado.
; Esta subrutina hace uso de las
; subrutinas de apoyo MONIT
; Modifica los registros A, DE y HL
;

```

```

213E 2A7E23        LD    HL, (237E) ; complementar
2141 7C            LD    A,H      ; el contador
2142 2F            CPL          ; de interrupciones
2143 57             LD    D,A      ;

```

```

2144 7D          LD    A,L          ;
2145 2F          CPL          ;
2146 5F          LD    E,A          ;
2147 13          INC    DE          ; incrementa en 1
2148 CD1322     CALL MONIT        ; activa
214B C9          RET          ; regresa

```

```

;
; subrutina VERIFI
; Esta subrutina tiene por objeto
; mostrar el contenido de las me-
; morias donde se encuentran los
; datos programados por el usuario
; además los resultados registrados
; por el sistema
; Esta subrutina emplea las subruti-
; nas ERROR y EXHIB
; Se modifica el valor de los regis-
; tros BC, DE, HL
;

```

```

214C 0EA1        LD    C,A1          ; Código "d"
214E CDD421     CALL ERROR        ;
2141 3E74        LD    A,74H          ;
2143 060A        LD    B,0A          ; Contador
2145 C5          EV1    PUSH BC
2146 CD2323     CALL EXHIB
2149 C1          POP    BC
214A 10F9        DJNZ  EV1
214C 0EAF        LD    C,A'F          ; Código "r"
214E CDD421     CALL ERROR
2151 3E82        LD    A,82H
2153 CD2323     CALL EXHIB
2156 C9          RET
                END

```

```

;
; termina la codificación del pro-
; grama principal y sus siete,
; subrutinas básicas.
; A continuación serán codificadas
; las subrutinas de apoyo que son
; empleadas por las subrutinas prin-
; cipales
;

```

```

                OR6    216B          ; empieza codificación
; Subrutinas de Apoyo

```

```

; Subrutina TECLA
; Esta subrutina espera a que el usuario oprima una tecla y en el momento en que lo hace se procede a decodificar el valor correspondiente, comparando código por código e incrementando un contador que al finalizar el proceso indicará el valor numérico de la tecla oprimida.
; Esta subrutina emplea a subrutina de utilidad ESPER y modifica los registros A, BC, HL; al final la información queda en A.
;

```

```

216B 3E3F RT0 LD A,3FH
216D D38C OUT (8C),A ; fijar filas a nivel bajo
216F DB90 IN A,(90)
2171 E61F AND IFH ; ¿dato de entrada?
2173 FE1F CP IFH ; no, regresar al-
2175 28F4 JR Z,RT0 ; -inicio de la rutina
2177 CD9422 CALL ESPER ; tiempo de retraso
217A 0E8C LD C,8CH
217C 0601 LD B,01H ; fijar la fila seleccionada
217E ED41 RT1 OUT (C),B ; a nivel bajo
2180 DB90 IN A,(90) ; recibir información-
2182 E61F AND IFH ; -de columna
2184 FE1F CP IFH ; ¿hay información?
2186 2009 JR NZ,RT2 ; sí, ir a decodificación
2188 CB20 SLA B ; no, seleccionar otra -
218A 3E40 LD A,40H ; -columna
218C B8 CP B ; se terminó de buscar
218D 20EF JR NZ,RT1 ; no, continuar proceso
218F 18DA JR RT0 ; reiniciar rutina
2191 0E00 RT2 LD C,00H ; proceso de decodificación
2193 0D RT3 DEC C
2194 CB38 SRL B
2196 20FB JR NZ,RT3 ; cuando B=0 continuar
2198 CB21 SLA C ; corrimiento a la izquierda
219A CB21 SLA C ; corrimiento a la izquierda
219C CB21 SLA C ; corrimiento a la izquierda
219E CB21 SLA C ; corrimiento a la izquierda
21A0 81 ADD C ; sumar C con A
21A1 21B321 LD HL,RT6 ; fijar apuntador de tabla
21A4 BE RT4 CP (HL) ; comparación con tabla
21A5 2804 JR Z,RT5 ; si se encontró, terminar

```

21A7	23		INC	HL	; incrementar apuntador
21A8	04		INC	B	; incrementar contador
21A9	18F9		JR	RT4	; regreso a comparación
21AB	DB90	RT5	IN	A,(90H)	
21AD	E61F		AND	IFH	
21AF	FE1F		CP	IFH	
21B1	20F8		JR	NZ,RT5	; esperar a que todas-
21B3	CD9422		CALL	ESPER	; -las teclas estén inactivas
21B6	78		LD	A,B	; transferir contador B a A
21B7	C9		RET		; regresar
21B8	FF	RT6	DEFB	FF	; tabla de códigos; "0"
21B9	EF		DEFB	EF	; código "1"
21BA	F7		DEFB	F7	; código "2"
21BB	FB		DEFB	FB	; código "3"
21BC	DF		DEFB	DF	; código "4"
21BD	E7		DEFB	E7	; código "5"
21BE	EB		DEFB	EB	; código "6"
21BF	CF		DEFB	CF	; código "7"
21C0	D7		DEFB	D7	; código "8"
21C1	DB		DEFB	DB	; código "9"
21C2	DD		DEFB	DD	; código "A"
21C3	ED		DEFB	ED	; código "B"
21C4	FD		DEFB	FD	; código "C"
21C5	0D		DEFB	0D	; código "D"
21C6	0B		DEFB	0B	; código "E"
21C7	07		DEFB	07	; código "F"
21C8	0E		DEFB	0E	
21C9	FE		DEFB	FE	
21CA	EE		DEFB	EE	
21CB	DE		DEFB	DE	
21CC	CD		DEFB	CD	
21CD	CB		DEFB	CB	
21CE	C7		DEFB	C7	
21CF	BF		DEFB	BF	
21D0	BD		DEFB	BD	
21D1	BB		DEFB	BB	
21D2	B7		DEFB	B7	
21D3	AF		DEFB	AF	

; Subrutina ERROR.  
; Esta subrutina tiene por objeto mostrar  
; un mensaje que generalmente es una condi-  
; ción errónea introducida por el usuario.  
; Hace uso de las subrutinas de utilería  
; ESPER y DESPL y modifica los registros

; BC, DE y HL; el código del mensaje  
 ; debe ser cargado previamente en el  
 ; registro C.

```

21D4 51 RE0 LD D,C ; repetir código error-
21D5 CDAA22 CALL DESPL ; -en D y desplegarlo
21D8 0625 LD B,25H ; repetir 25 veces
21DA CD9422 RE1 CALL ESPER ; el retardo
21DD 10FB DJNZ REI ;
21DF C9 RET ; regresar

```

; Subrutina COMUN.

; Esta subrutina toma un valor decimal,  
 ; lo transforma a hexadecimal, lo multi-  
 ; plica por el costo de la gasolina, lo  
 ; vuelve a su forma decimal y lo almace-  
 ; na en las localidades de memoria dis-  
 ; puestas para ello.  
 ; Se emplean las subrutinas de utilería  
 ; DECBIN, MULTI, BINDEC, EXHIB,  
 ; DESPL y BOREG. son utilizados y modi-  
 ; ficados los registros A, BC, DE, HL,  
 ; e IX. El dato a transformar deberá es-  
 ; tar cargado previamente.

```

21E0 DD217823 RC0 LD IX, MEMDLITH+5; apuntar al dato
21E4 CDC822 CALL DECBIN ; pasarlo a binario
21E7 227E23 LD (MEMDLITH),HL; almacenarlo
21EA EB EX DE, HL
21EB ED4B6F23 LD BC, (GASOL) ; multiplicarlo-
21EF CDDE22 CALL MULTI ; -por costo gasolina
21F2 228023 LD (MEMDLINH),HL; guardar resultado
21F5 AF XOR A
21F6 BB CP E
21F7 2010 JR NZ, RC1 ; indicar exceso
21F9 DD217D23 LD IX, MEMDLIN+5; apuntar resultado
21FD 2A3023 LD HL, (MEMDLINH); cargar dato
2200 CDF622 CALL BINDEC ; transformarlo a decimal
2203 3E74 LD A, 74 ; apuntador memoria
2205 CD2323 CALL EXHIB ; mostrar al usuario
2208 C9 RET ; regreso
2209 0EF0 RC1 LD C, FOH
220B 51 LD D, C ; indicar exceso

```

220C	CDA A22	CALL	DESPL		
220F	CDBD22	CALL	BOREG		; limpiar registros
2212	C9	RET			; regreso
					; Subrutina MONIT.
					; Esta subrutina tiene como objeto activar
					; al motor eléctrico que acciona la bomba
					; de gasolina; además deberá monitorear
					; al sistema y calcular el consumo en cada
					; ciclo, controlará además el suministro de
					; gasolina de acuerdo a las condiciones pro-
					; gramadas por el usuario. Además deberá
					; desactivar al sistema cuando se presenta
					; una interrupción.
					; Esta subrutina utiliza a las subrutinas de
					; utilería MULTL, BINDEC, EXHIBE y mo-
					; difica los registros BC, DE, HL, IX ade-
					; más del acumulador.
2213	3E22	RM0	LD	A, 22H	; fijar dirección-
2215	ED47		LD	I, A	; -de interrupción
2217	3EA8		LD	A, A8H	
2219	D334		OUT	(84), A	
221B	3E05		LD	A, 05	; Z80 CTC como-
221D	D384		OUT	(84), A	; -contador
221F	3A7123		LD	A, (CONTSC)	; constante de-
2222	D384		OUT	(84), A	; -tiempo
2224	ED5E	RM1	IM	2	; fijar modo 2 de int.
2226	3E01		LD	A, 01H	
2228	D394		OUT	(84), A	; activar motor
222A	91F030		LD	BC, 8010	; contador de no operación
222D	FB	RM2	EI		; permite interrupción
222E	3A7323		LD	A, (BANDER)	; localidad 2373
2231	CB6F		BIT	5, A	; existió interrupción
2233	2316		JR	Z, RM6	; calcular consumo
2235	3E3F	RM3	LD	A, 3F	; ver si se -
2237	D38C		OUT	(8C), A	; oprime comando -
2239	DB90		IN	A, (90)	; de interrupción
223B	E61F		AND	IFH	
223D	FE17		CP	A, 17H	
223F	2806		JR	Z, RM5	; sí, calcula y termina
2241	10EA		DJNZ	RM2	; no, regresa
2243	F3	RM4	DI		; desacopla interrupción
2244	0D		DEC	C	; decrementa contador

2245	20E6		JR	NZ, RM2	
2247	110100	RM5	LD	DE, 01H	; contador de int.-
224A	F3	RM6	D1		; -en uno
224B	D5		PUSH	DE	; proteger contador-
224C	3A7323		LD	A, (BANDER)	; -de int.
224F	CRAF		RES	5, A	
2251	327323		LD	(BANDER), A	; ver si hubo inte-
2254	ED5B823		LD	DE, (RESLITH)	; -rrupción
2258	210100		LD	HL, (0001)	
225B	19		ADD	HL, DE	; sumar 1 a litros
225C	228C23		LD	(RESLITH), HL	
225F	EB		EX	DE, HL	
2260	ED4B6F23		LD	BC, (GASOL)	
2264	CDDE22		CALL	MULTI	; obtener nuevo total
2267	228E23		LD	(RESDINH), HL	
226A	DD218B23		LD	IX, RESDIN+5	
226E	CDF622	RM7	CALL	BINDEC	
2271	DD218623		LD	IX, RESLIT+5	
2275	2A8C23		LD	HL, (RESLITH)	
2278	CDF622	RM8	CALL	BINDEC	
227B	3E82		LD	A, 82	; apuntador resultados
227D	CD2323		CALL	EXHIB	; muestra totales
2280	D1		POP	DE	; restablece cont. int.
2281	1B		DEC	DE	; decrementarlo
2282	20A9		JR	NZ, RM2	
2284	3E00		LD	A, 00	
2286	D394	RM9	OUT	(94), A	; apaga motor
2288	3E03		LD	A, 03H	
228A	D384		OUT	(84), A	; desactiva Z80CTC
228C	C9		RET		; regresa
			END		; fin de codificación

;
  
; A continuación serán codificadas las
  
; subrutinas de utilería que son emplea-
  
; das por las subrutinas de apoyo, y las
  
; subrutinas principales.

;
  
ORG 2294 ; reinicio de codifi-
  
; Subrutinas de utilería ; -cación
  
; Subrutina ESPER
  
; Esta subrutina tiene por objeto generar
  
; un retraso de 20 ms en la ejecución
  
; del programa; se modifica el contenido
  
; del registro HL

2294	21FF08	AE0	LD	HL,08FFH	
2297	2D	AE1	DEC	L	
2298	20FD		JR	NZ,AE1	
229A	25		DEC	H	
229B	20FA	AE2	JR	NZ,AE1	
229D	C9		RET		

; Rutina INTER  
; Esta rutina tiene como objeto fijar  
; un indicador para indicar que el sis-  
; tema fue interrumpido, modificando  
; solamente al acumulador A.  
; En las localidades 22A8H y 22A9H  
; se almacena la dirección del salto  
; vectorizado a la propia subrutina  
; INTER (dirección 229EH)

229E	3A7323		LD	A,(BANDER)	
22A1	CBEF		SET	5,A	; fija condición
22A3	327323		LD	(BANDER),A	; almacena
22A6	ED4D		RETI		; regresa de interrupción
22A8	9E		DEFB	9E	; tabla de salto
2249	A9		DEFB	22	; a INTER

; Subrutina DESPL.  
; Esta subrutina tiene por objeto en-  
; viar un mensaje al usuario a través  
; del Sistema de despliegue de infor-  
; mación.  
; El código del mensaje deberá car-  
; garse previamente en el registro C  
; para que aparezca en el exhibidor de  
; caudal y en el registro D, para que  
; aparezca en el exhibidor de costo;  
; esta subrutina modifica los registros  
; B, H y L.

22AA	0605		LD	B,5H	; contador de ciclos
22AC	2100FE		LD	HL,FE00	; apuntar a exhibidor
22AF	71	AD0	LD	(HL),C	; cargar en exhibidor
22B0	23		INC	HL	; siguiente exhibidor
22B1	10FC		DJNZ	AD0	
22B3	2110FE		LD	HL,FE10	; exhibidor costo
22B6	0605		LD	B,5H	; contador de ciclos
22B8	72	AD1	LD	(HL),D	; cargar en exhibidor

```

22B9 23          INC    HL          ; siguiente exhibidor
22BA 10FC       DJNZ   AD1         ; repetir hasta B=0
22BC C9         RET     .          ; regresa

```

```

;
; Subrutina BOREG
; Esta subrutina tiene por objeto colo-
; car cero en todos los registros donde
; se almacenan los datos de un usuario
; y los resultados de la operación del
; sistema; esta subrutina modifica los
; registros A, B, D y E.

```

```

22BD 117423     AB0    LD     DE,2374H    ; apunta a registros
22C0 061E       LD     B,1EH      ; contador de ciclos
22C2 AF        XOR    A          ; poner cero en A
22C3 12        AB1    LD     (DE),A      ; cargar cero
22C4 13        INC    DE          ; nuevo registro
22C5 10FC       DJNZ   AB1         ; repetir hasta B=0
22C7 C9        RET     .          ; regresa

```

```

;
; Subrutina DECBIN
; Esta subrutina tiene por objeto trans-
; formar un número decimal de cinco dí-
; gitos a hexadecimal dejando al final el
; número en el registro par HL; los datos
; deben estar almacenados en cinco loca-
; lidades de memoria y el apuntador IX de-
; berá indicar al dígito de más alto orden;
; se afectan los registros IX, HL, DE y BC

```

```

22C8 0605       LD     B,5H        ; números decimales a trsf.
22CA 210000     LD     HL,00H      ; limpia reg. de resultados
22CD 29        AX1   ADD    HL,HL      ; resultado por dos
22CE E5        PUSH   HL          ; almacenar
22CF 29        ADD    HL,HL      ; resultado por cuatro
22D0 29        ADD    HL,HL      ; resultado por ocho
22D1 D1        POP    DE          ; restablece res.de AX1
22D2 19        ADD    HL,DE      ; resultado por diez
22D3 DD5E00     LD     E,(IX)      ; carga núm. en E
22D6 1600       LD     D,00H      ; limpia D
22D8 19        ADD    HL,DE      ; sumar
22D9 DD2B       DEC    IX          ; apunta nuevo número
22DB 10F0       DJNZ   AX1         ; si E#0 ir a AX1
22DD C9        RET     .          ; regresa

```

; Subrutina MULTI.  
 ; Esta subrutina realiza una multiplicación en hexadecimal entre dos números de dos bytes cada uno, dejando el resultado en los registros DEHL.  
 ; En DE se carga previamente el multiplicando y en BC se carga el multiplicador; se modifican y utilizan los registros A, BC, DE, y HL.

22DE	3E10		LD	A,10H	; contador para 16 ciclos
22E0	210000		LD	HL,00H	; cero en HL
22E3	EB	AM1	EX	DE,HL	; corrimiento a-
22E4	29		ADD	HL,HL	; la izquierda
22E5	F5		PUSH	AF	; guarda acarreo
22E6	EB		EX	DE,HL	
22E7	29		ADD	HL,HL	; corrimiento de HL
22E8	3001		JR	NC,AM2	
22EA	13		INC	DE	; si hay acarreo, increm.
22EB	F1	AM2	POP	AF	; restablece acarreo
22EC	3004		JR	NC,AM4	; si no existe, ir a AM4
22EE	09		ADD	HL,BC	; sumar HL y BC
22EF	3001	AM3	JR	NC,AM4	; si hay acarreo-
22F1	13		INC	DE	; incrementa DE
22F2	3D	AM4	DEC	A	; decrementa contador-
22F3	C8		RET	Z	; y regresa cuando A=0
22F4	18ED		JR	AM1	; realiza otro ciclo

; Subrutina BINDEC.

; Esta subrutina transforma un número hexadecimal de cuatro bytes a su equivalente decimal; el número a transformar deberá ser cargado previamente en HL y los resultados serán almacenados en las memorias apuntadas por IX para el número más significativo. Son modificados los registros IX, IY, HL, DE y A

22F6	FD211923		LD	IY,AT4	; apuntador potencias
22FA	AF	AT0	XOR	A	; cero en contador
22FB	FD5601	AT1	LD	D,(IY+1)	; cargar potencia-
22FE	FD5E00		LD	E,(IY+0)	; de 10

2301	B7	AT2	OR	A	; bandera de acarreo =0
2302	ED52		SBC	HL,DE	; restar potencia
2304	3803		JR	C,AT3	;
2306	3C		INC	A	; incrementar cont. dígito
2307	18F8		JR	AT2	; seguir restando
2309	19	AT3	ADD	HL,DE	; restablecer última comp.
230A	DD7700		LD	(IX+0),A	; almacenar cont.dígito
230D	DD2B		DEC	IX	; indicar siguiente dígito
230F	FD23		INC	IY	; cargar siguiente-
2311	FD23		INC	IY	; potencia
2313	7B		LD	A,E	; comprueba si se-
2314	FE01		CP	01H	; terminó
2316	20E2		JR	NZ,AT0	; prosigue
2318	C9		RET		; regresa
2319	10	AT4	DEFB	10H	
231A	27		DEFB	27H	; diez mil
231B	E8		DEFB	E8H	
231C	03		DEFB	03H	; mil
231D	64		DEFB	64H	
231E	00		DEFB	00H	; cien
231F	0A		DEFB	0AH	
2320	00		DEFB	00H	; diez
2321	01		DEFB	01H	
2322	00		DEFB	00H	; uno

;

; Subrutina EXHIB.

; Su función es transferir los datos y resultados almacenados en las memorias del sistema a los exhibidores de caudal y costo. Deberá ser indicado con el código 74H cargado en el acumulador, si se desea transferir datos programados y con el código 82H si son los resultados del suministro los que se quieren mostrar. Se modifican los registros A, BC, DE y HL.

2323	1100FE	ARO	LD	DE,FE00H	; primer exhibidor
2326	0623		LD	B,23H	
2328	4F		LD	C,A	; apuntador datos o result.
2329	3E0A		LD	A,AH	; fijar contador
232B	327223		LD	(MEMAUX),A	; de eventos

232E	214E23	AR1	LD	HL, AR2	; HL indica códigos
2331	0A		LD	A, (BC)	; carga número-
2332	C5		PUSH	BC	; - a desplegar -
2333	0600		LD	B, 00H	; en BC y sumarlo-
2335	4F		LD	C, A	; -al apuntador de -
2336	09		ADD	HL, BC	; -códigos
2337	EDA0		LDI		; transfiere (HL) a (DE)
2339	C1		POP	BC	
233A	03		INC	BC	; incrementar apuntador
233B	3A7223		LD	A, (MEMAUX)	
233E	3D		DEC	A	
233F	327223		LD	(MEMAUX), A	; decrementar contador
2342	FE00		CP	00H	; analizar si se-
2344	C8		RET	Z	; finalizó y regresar
2345	FE05		CP	05H	; compara con cinco
2347	20E5		JR	NZ, AR1	; continua proceso
2349	1110FE		LD	DE, FE10	; exhibidor costo
234C	18E0		JR	AR1	; continua proceso
234E	C0	AR2	DEFB	C0	; código 0
234F	F9		DEFB	F9	; código 1
2350	A4		DEFB	A4	; código 2
2351	B0		DEFB	B0	; código 3
2352	99		DEFB	99	; código 4
2353	92		DEFB	92	; código 5
2354	82		DEFB	82	; código 6
2355	F8		DEFB	F8	; código 7
2356	80		DEFB	80	; código 8
2357	90		DEFB	90	; código 9

; Subrutina DIVIS.  
; Esta subrutina efectúa una división hexa-  
; decimal entre un dividendo de cuatro by-  
; tes almacenados en HLDE y un divisor  
; de dos bytes almacenado en BC. Al ter-  
; minar la división el cociente de dos by-  
; tes estará en DE y el residuo en HL. Se  
; modifican y utilizan los registros A, DE,  
; BC, HL.

2358	3E10		LD	A, 10H	; contador de ciclos
235A	29	AV0	ADD	HL, HL	; corrimiento de HL
235B	EB		EX	DE, HL	
235C	29		ADD	HL, HL	; corrimiento de -
235D	EB		EX	DE, HL	; - DE

235E	30J1		JR	NC,AV1	; si hay acarreo-
2360	23		INC	HL	; -incrementa HL
2361	B7	AV1	OR	A	; borrar acarreo
2362	ED42		SBC	HL, BC	; restar divisor
2364	13		INC	DE	; incrementar cociente
2365	F26B23		JP	P,AV2	; con signo negativo-
2368	09		ADD	HL, BC	; restablece divisor
2369	CB83		RES	0,E	; restablece cociente
236B	3D	AV2	DEC	A	; decrementa ind.ciclos
236C	20EC		JR	NZ,AV0	; continua proceso
236E	C9		RET		; regresar

;
  
; Las etiquetas utilizadas en los programas
  
; anteriores fueron las siguientes:

;
  
MEMAUX EQU 2372
  
BANDER EQU 2373
  
GASOL EQU 236F
  
MEMMLIT EQU 2374
  
MEMMLITH EQU 237E
  
MEMDIN EQU 2379
  
MEMDINH EQU 2380
  
RESLIT EQU 2382
  
RESLITH EQU 238C
  
RESDIN EQU 2387
  
RESDINH EQU 238C
  
CONSC EQU 2371
  
END

## MAPA DE MEMORIA

A continuación se muestra la colocación de cada subrutina en la memoria del nanocomputador, y también las posiciones que ocupan los registros de datos, resultados, condiciones, etc.

### Programa y subrutinas

- 1.- Programa principal (MASTER) ocupa las posiciones 2000 a 203A.
- 2.- Subrutina de supervisión (SUPERV) ocupa las posiciones 203B a 2053.
- 3.- Subrutina de inicialización (INICIA) ocupa las posiciones 2054 a 2062.
- 4.- Subrutina de programación de caudal (LITROS) ocupa las posiciones 2063 a 20A9.
- 5.- Subrutina de programación de costo (DINERO) ocupa las posiciones 20AA a 2110.
- 6.- Subrutina de activación y monitoreo (ACTIVA) ocupa las posiciones 2111 a 213D.
- 7.- Subrutina de reactivación y monitoreo (REACTI) ocupa las posiciones 213E a 214B.
- 8.- Subrutina de verificación de registros (VERIFI) ocupa las posiciones 214C a 2156.
- 9.- Primera zona libre de programa ocupa las posiciones 2157 a 216A.
- 10.- Subrutina para obtener datos (TECLA) ocupa las posiciones 216B a 21D3.
- 11.- Subrutina para indicar condiciones (ERROR) ocupa las posiciones 21D4 a 21DF.
- 12.- Subrutina de multiplicación decimal (COMUN) ocupa las posiciones 21E0 a 2212.

- 13.- Subrutina para monitoreo y control de operación (MONIT) ocupa las posiciones 2213 a 228C.
- 14.- Segunda zona libre de programa ocupa las posiciones 228D a 2293.
- 15.- Subrutina de tiempo de espera (ESPER) ocupa las posiciones 2294 a 229D.
- 16.- Rutina de servicio a interrupción (INTER) ocupa las posiciones 229E a 22A9.
- 17.- Subrutina para despliegue de mensajes (DESPL) ocupa las posiciones 22AA a 22BC.
- 18 - Subrutina para borrado de registros (BOREG) ocupa las posiciones 22BD a 22C7.
- 19.- Subrutina de transformación a hexadecimal (DECBIN) ocupa las posiciones 22C8 a 22DD.
- 20.- Subrutina de multiplicación hexadecimal (MULTI) ocupa las posiciones 22DE a 22F5.
- 21.- Subrutina de transformación a decimal (BINDEC) ocupa las posiciones 22F6 a 2322.
- 22.- Subrutina para despliegue de registros (EXHIB) ocupa las posiciones 2323 a 2357.
- 23.- Subrutina de división hexadecimal (DIVIS) ocupa las posiciones 2358 a 236E.

### Registros y memoria para datos y resultados

- 1.- Costo de la gasolina (GASOL) ocupa las posiciones 236F y 2370.
- 2.- Constante del medidor de caudal (CONSC) ocupa la posición 2371.

- 3.- Memoria auxiliar (MEMAUX) ocupa la posición 2372.
- 4.- Registro de condiciones (BANDER) ocupa la posición 2373.
- 5.- Datos de caudal en decimal (MEMLIT) ocupan las posiciones 2374 a 2378.
- 6.- Datos de costo en decimal (MEMDIN) ocupan las posiciones 2379 a 237D.
- 7.- Datos de caudal en hexadecimal (MEMLITH) ocupan las posiciones 237E y 237F.
- 8.- Datos de costo en hexadecimal (MEMDINH) ocupan las posiciones 2380 y 2381.
- 9.- Resultados de caudal en decimal (RESLIT) ocupan las posiciones 2382 a 2386.
- 10.- Resultados de costo en decimal (RESDIN) ocupan las posiciones 2387 a 238B.
- 11.- Resultados de caudal en hexadecimal (RESLITH) ocupan las posiciones 238C y 238D.
- 12.- Resultados de costo en hexadecimal (RESDINH) ocupan las posiciones 238E y 238F.

El equipo de control para el sistema de suministro de gasolina está constituido por los siguientes elementos.

- 1.- Subsistema de microcomputadora.
- 2.- Subsistema de despliegue de información.
- 3.- Subsistema del transductor de caudal.
- 4.- Subsistema de control del motor.
- 5.- Subsistema de alimentación del equipo.

### Subsistema de microcomputadora

Está constituido por el nanocomputador Z80 Starter Kit, el cual consta de los siguientes elementos: microprocesador Z80 CPU, oscilador, circuito contador temporizador Z80 CTC, circuito de interfase para entrada y salida en paralelo Z80 PIO, memorias de lectura y escritura (RAM), memoria de lectura (ROM) para control del nanocomputador, sistema de despliegue con seis exhibidores, teclado hexadecimal con funciones de control para el programa monitor, interfase para el bus normalizado S-100, área de conexión externa con señales del sistema, programador de memoria tipo EPROM e interfase para grabadora.

### Subsistema de despliegue de información

Este equipo construido en la Facultad de Estudios Superiores Cuautitlan\*, está compuesto por treinta y dos exhibidores luminosos de siete segmentos y por el control de los exhibidores, el cual permite que sean manejados por el microprocesador Z80 CPU como localidad de memoria.

\* El sistema de despliegue de información fue diseñado en la Facultad de Estudios Superiores Cuautitlan, por el Maestro en Ciencias Juan Antonio Navarro M. y la Ingeniero Mecánico Electricista Lourdes Clares Fuentes, en el año de 1981.

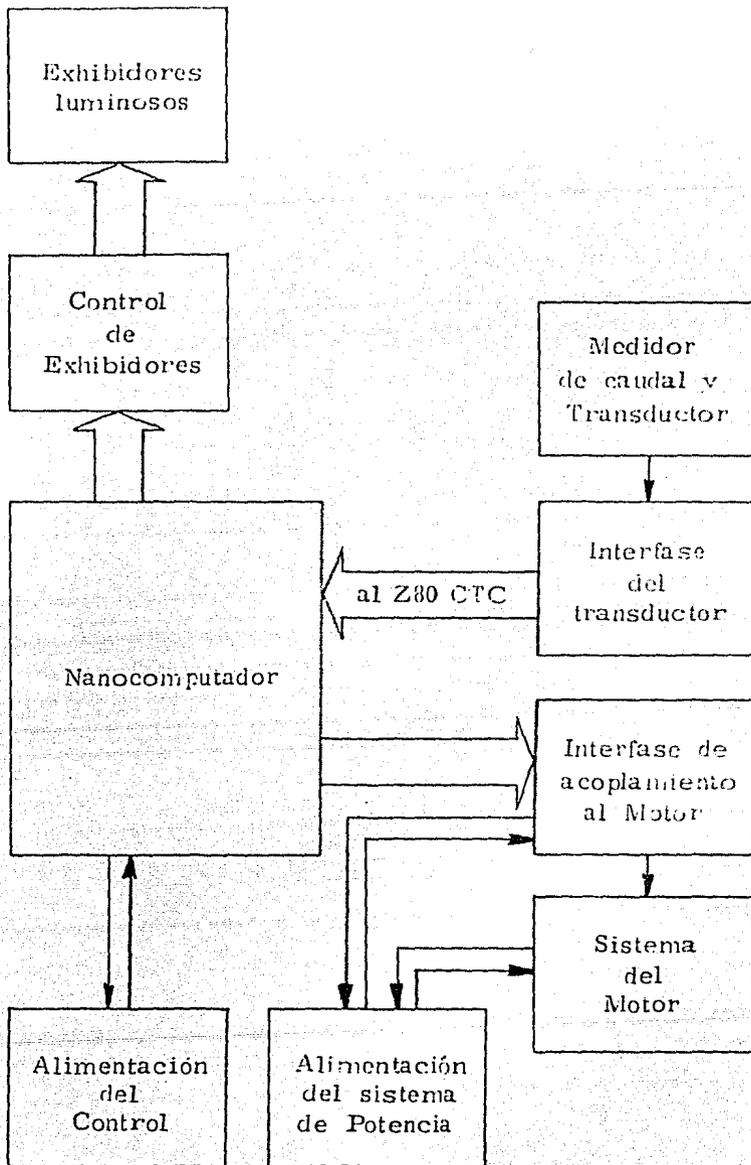


DIAGRAMA A BLOQUES DEL SISTEMA

Es conectado al nanocomputador Z80 Starter Kit a través del bus normalizado S-100; la interfase entre el bus y el nanocomputador Z80 Starter Kit es parte integrante de este último.

### Subsistema del transductor de caudal

Este equipo consta de un medidor de desplazamiento positivo para caudal y un sistema de detección óptico que permite la transmisión de una señal digital al subsistema del nanocomputador Z80 Starter Kit a través del circuito de conteo y temporización Z80 CTC.

El sistema de detección está formado por un arreglo de fotodiodo y fotosensor, además de los circuitos adecuados para transmitir una señal compatible con el sistema del nanocomputador.

### Subsistema de control del motor

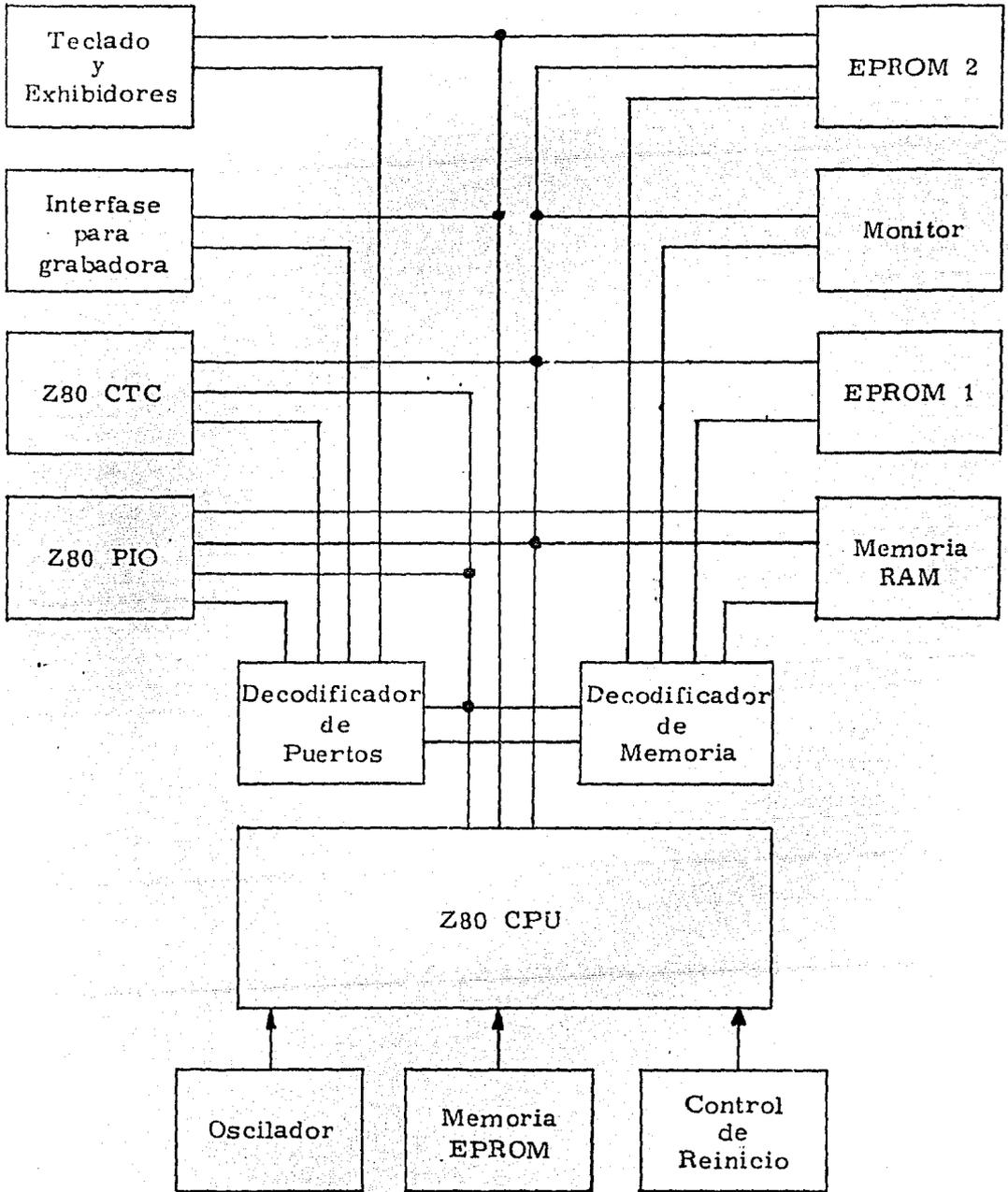
El subsistema de control del motor consiste básicamente en un tiristor bidireccional que permite, o no, el flujo de energía al motor eléctrico; este tiristor está controlado por una señal generada en el nanocomputador Z80 Starter Kit y transferida por medio de un optoacoplador al tiristor bidireccional.

### Subsistema de alimentación del equipo

Este subsistema incluye por una parte la alimentación al nanocomputador Z80 Starter Kit, al subsistema de despliegue de información, al subsistema transductor de caudal; por otro lado incluye la alimentación para el subsistema de control del motor que debe estar físicamente aislado del resto del sistema.

## NANOCOMPUTADOR

El nanocomputador Z80 Starter Kit, es un dispositivo implementado en una tarjeta de circuito impreso y está constituido por los siguientes elementos:



ELEMENTOS DEL NANOCOMPUTADOR

- 1.- Sincronizador o reloj.
- 2.- Microprocesador Z80 CPU.
- 3.- Circuito de conteo y temporización Z80 CTC.
- 4.- Interfase de entrada y salida en paralelo Z80 PIO.
- 5.- Teclado y exhibidores.
- 6.- Memoria del sistema (tipos RAM y ROM).
- 7.- Interfase para utilizar el bus S-100.
- 8.- Area de conexión externa.

Además, el nanocomputador cuenta con otros elementos, como son, la interfase para grabadora y programador de memorias del tipo EPROM.

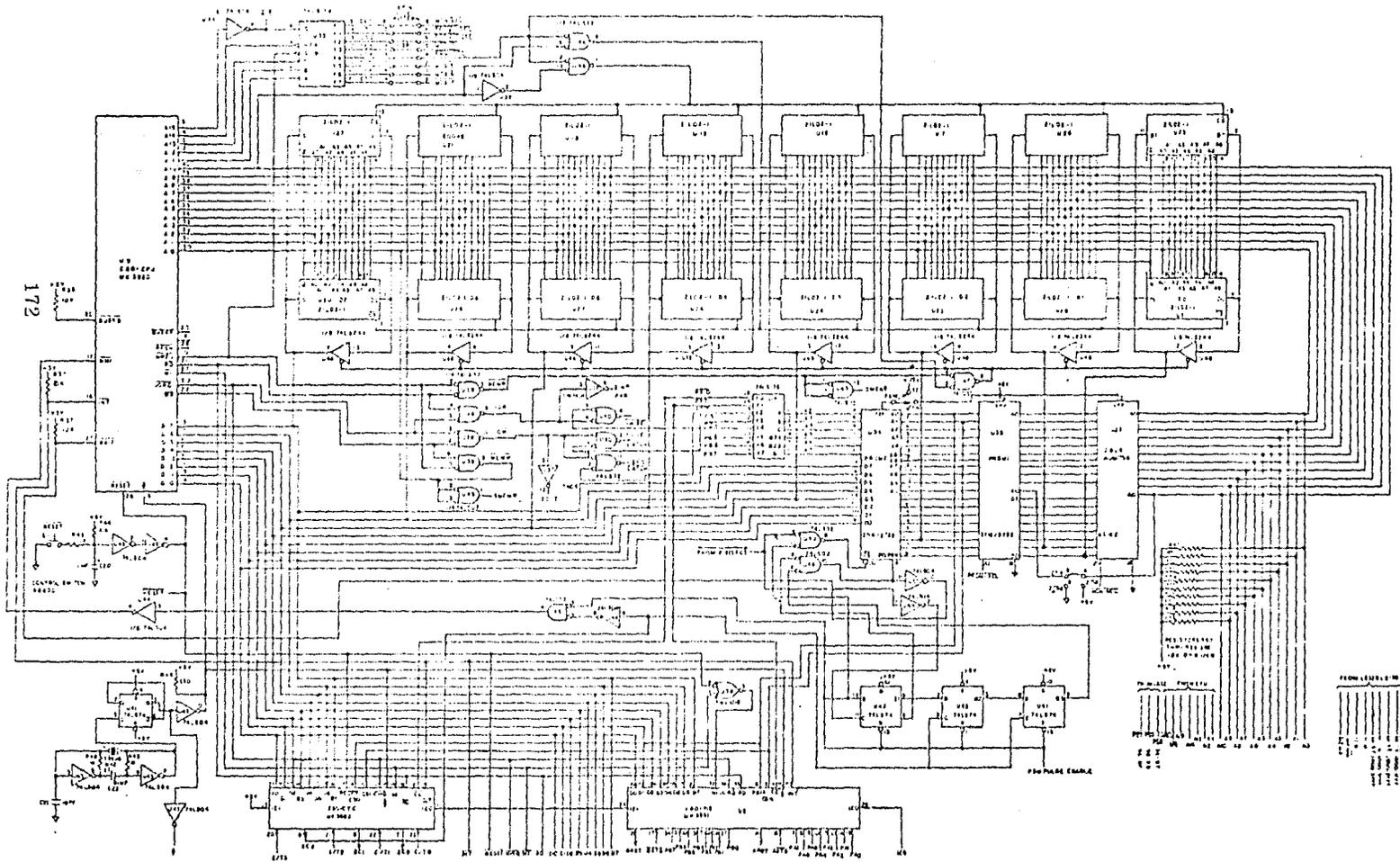
#### Sincronizador o reloj

Está constituido por un oscilador que genera una frecuencia aproximada de 2MHz; esta frecuencia es baja para el microprocesador Z80 CPU, pero tiene la ventaja de permitir la utilización de memorias lentas y una gran compatibilidad con los periféricos de la serie 8080.

#### Microprocesador Z80 CPU

Este microprocesador es el cerebro del sistema y provee las señales necesarias para controlar el teclado y el sistema de despliegue (este último no será empleado en este diseño). Las señales de control del microprocesador, las líneas del bus de datos y las líneas del bus de direcciones, van dirigidas a un área de alambrado externo donde se puede conectar circuitería extra al nanocomputador.





FROM LEVEL 1 IN

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

## Circuito de conteo y temporización Z80 CTC

Este dispositivo consta de cuatro contadores temporizadores independientes; el programa monitor utiliza tres de estos canales para realizar funciones de lectura y escritura en una grabadora y en la programación de memorias EPROM; el canal número cero está disponible siempre para el usuario y es el que se emplea para recibir la información del transductor de caudal.

## Interfase de entrada y salida en paralelo Z80 PIO

Este circuito integrado de la familia Z80 permite transferir información en forma paralela a través de dos puertos de ocho bits; la información puede ser transferida a los puertos, o bien recibida desde ellos.

Este circuito programable tiene una gran cantidad de aplicaciones, sin embargo, no es necesaria su utilización en este diseño.

## Teclado y exhibidores

El teclado del nanocomputador consta de 28 teclas incluyendo los números hexadecimales, y 12 teclas para comunicarse con el programa monitor que controla el sistema; en este diseño no es necesario emplear estas teclas de comando y tampoco serán empleados los seis exhibidores luminosos que tiene el sistema y que permiten obtener información del contenido de los registros del microprocesador, de sus puertos y del contenido de las localidades de memoria.

El teclado es utilizado en este diseño de la siguiente forma:

- 1.- Los números del 0 al 9 se utilizan como tal, ya que son necesarios para introducir los datos del usuario, en sistema decimal.
- 2.- Los números hexadecimales A, B, C, D, E, F, son utilizados como comandos, esto es, no serán interpretados como números hexadecimales sino como órdenes para el sistema.

- 3.- Las teclas de control para el funcionamiento del nanocomputador no serán empleadas, esto es, no serán accedidas por el usuario del sistema de suministro de gasolina.

### Memoria de lectura y escritura

El nanocomputador Z80 Starter Kit cuenta con capacidad para almacenar 2048 bytes de información en memoria de sólo lectura ROM, en la cual se encuentra alojado el programa monitor del sistema (la denominación de esta memoria es 8316 E-ROM-ZBUG); además, cuenta con capacidad para almacenar 1024 bytes de información en memoria de lectura y escritura del tipo 21LO2-1 RAM, en las cuales pueden almacenarse datos y programas. El nanocomputador tiene la posibilidad de alojar más memoria de ambos tipos. A continuación aparecen las localidades de memoria utilizadas por el programa monitor, las áreas para incrementar memoria de sólo lectura, el área para almacenar datos en memoria de lectura y escritura (RAM), y las localidades que se utilizan para transferir información al subsistema de despliegue de información.

### Mapa de la memoria del nanocomputador

- 1.- Área de memoria utilizada por el programa monitor. Ocupa las localidades de memoria 0000 a 07FF.
- 2.- Primer área para memoria programable. Ocupa las posiciones 0800 a 0FFF. (opcional).
- 3.- Segunda área para memoria programable. Ocupa las posiciones 1000 a 17FF. (opcional).
- 4.- Primer área no utilizada. Ocupa las posiciones 1800 a 1FFF.
- 5.- Área de memoria de acceso aleatorio disponible para el usuario. Ocupa las localidades de memoria 2000 a 333F.
- 6.- Área de memoria de acceso aleatorio utilizada por el apilador (stack). Ocupa las localidades de memoria 2390 a 23A0.

- 7.- Area de memoria de acceso aleatorio utilizada para tablas de puntos de ruptura y almacenamiento de registros. Ocupa las posiciones de memoria 2AA9 a 23FF.
- 8.- Area de memoria de acceso aleatorio opcional. Ocupa las posiciones de memoria 2400 a 27FF.
- 9.- Segunda área de memoria no utilizada. Ocupa las posiciones de memoria 2800 a FDFE.
- 10.- Area utilizada por el sistema de despliegue de información (en este proyecto). Ocupa las posiciones de memoria FE00 a FE1F.
- 11.- Tercera área no utilizada. Ocupa las posiciones de memoria FE20 a FFFF.

#### Interfase para el bus S-100

Esta interfase es empleada para permitir la conexión de diferentes dispositivos al sistema del nanocomputador; en este diseño permite comunicarse con el sistema de despliegue de información.

#### Area para conexión externa

El nanocomputador tiene una zona específica para acoplar circuitos integrados, ya sea para expansiones de memoria, interfases de video, etc.. Las señales del Z80 CPU, del Z80 PIO, del Z80 CTC y otras señales, como las selectoras de puerto (PS5, PS6, PS7, activas con nivel lógico bajo), se pueden obtener de puntos cercanos a esta área.

#### EXHIBIDORES

El objetivo de este dispositivo es el mostrar en una serie de exhibidores luminosos de siete segmentos, ciertos datos o resultados obtenidos durante la operación del sistema de suministro de gasolina.

Este dispositivo está constituido por dos tarjetas de circuito impreso y son:

- 1.- Tarjeta de control del sistema de despliegue.
- 2.- Tarjeta de exhibidores luminosos.

### Tarjeta de control del sistema de despliegue

Esta tarjeta tiene la función de controlar a los 32 exhibidores luminosos del dispositivo y lograr que el microprocesador Z80 CPU pueda direccionar a dichos exhibidores como localidades de memoria, comunicándose estos dos subsistemas a través del bus S-100.

Las partes principales de esta tarjeta de control son las siguientes:

- 1.- Oscilador
- 2.- Memoria de acceso aleatorio
- 3.- Contador de estados
- 4.- Decodificador de estados

### Oscilador

El oscilador genera una frecuencia de 80 KHz que permite sincronizar la operación del contador de estados y del decodificador de estados.

### Memoria de acceso aleatorio

La tarjeta de control dispone de 48 bytes de memoria RAM, expandible hasta 64 bytes; estas localidades de memoria son accedidas por el Z80 CPU a través del bus S-100 y contienen la información que se muestra en los exhibidores luminosos, siendo accedida por el procesador únicamente para modificar dicha información.

## Contador de estados

El contador de estados recibe pulsos del oscilador direccionando así, una por una, las dieciseis palabras de cuatro bits de la memoria del sistema y manejando los segmentos luminosos de tres exhibidores simultáneamente.

## Decodificador de estados

Los ánodos de los exhibidores luminosos son activados por el decodificador de estados, permitiendo que los transistores que los alimentan se activen; el decodificador es alimentado por el contador, permitiendo que se puedan encender hasta cuarenta y ocho exhibidores luminosos, de tres en tres.

Cuando es accesado este sistema por el nanocomputador es colocada en el bus de direcciones la localidad del exhibidor seleccionado, transfiriendo a través del bus de datos del bus S-100 la información de los segmentos a activar, procediendo a poner en estado activo la señal sMEMW del bus S-100; las líneas A0-A3 son direcciones para seleccionar una de las 16 localidades de la memoria; las líneas A4-A7 seleccionan un exhibidor (con 1 byte) del total direccionable que es 24.

Las líneas A8-A16 permiten escribir la nueva información en el exhibidor direccionado en el momento de activarse la señal sMEMW.

## Tarjeta de exhibidores luminosos

La tarjeta de exhibidores luminosos está constituida por un arreglo de dos filas de exhibidores de siete segmentos del tipo TIL 312, contando cada fila con un total de dieciseis elementos. En este diseño no se utilizan todos los exhibidores sino únicamente diez, cinco en la fila superior, conformando los exhibidores de caudal y cinco en la fila inferior constituyendo los exhibidores de costo. Aún cuando los demás exhibidores no son empleados, el utilizar esta tarjeta nos permite ampliar la capacidad del sistema, enviando mensajes o manejando números mucho mayores o incluso permitiendo informar al usuario de otras condiciones de operación, todo ello con cambios de programa únicamente.

En este sistema las direcciones de los exhibidores luminosos son las siguientes:

FE00	exhibidor	00
FE0F	exhibidor	15
FE10	exhibidor	16
FE1F	exhibidor	31

Sin embargo, sólo serán utilizados los exhibidores luminosos FE00 a FE04 y FE10 a FE14 pudiéndose alterar esta configuración por programa y permitiendo la utilización de los demás exhibidores luminosos para otro fin en caso de ser necesario; las direcciones en memoria pueden ser modificadas empleando para ello el micro-interruptor que se encuentra localizado en la tarjeta de control.

## TRANSDUCTOR

Este subsistema tiene la función de detectar el flujo de gasolina que circula a través de una tubería y proporcionar una señal eléctrica del tipo digital al subsistema de microcomputadora de acuerdo al flujo de gasolina que esté siendo suministrado.

Este subsistema se compone básicamente de dos elementos, que son:

- 1.- Medidor de caudal
- 2.- Acoplador y convertidor

### Medidor de caudal

El medidor de caudal es un dispositivo cuya función es determinar la cantidad de fluido que circula a través de él.

En un medidor de desplazamiento positivo, el caudal se determina subdividiendo la corriente total del fluido en fracciones de volumen conocido y la medición se efectúa entonces mediante el recuento del número de fracciones en la unidad de tiempo.

De los tres tipos de medidores de desplazamiento positivo de Pistón oscilante, de Disco oscilante y el de Embolos múltiples, el primero es el más utilizado para la medición del caudal de gasolina ya que resiste mejor la corrosión de los componentes de la gasolina; está formado básicamente por una cámara de medida, constituida por un anillo externo que es la pared de la caja y un anillo interno concéntrico con el anillo externo. Funcionalmente está formado por las siguientes partes:

- 1.- Orificio de entrada
- 2.- Cámara anular para la medida del fluido que entra
- 3.- Pistón hueco
- 4.- Diafragma
- 5.- Orificio de salida
- 6.- Cámara anular para la medida del fluido que sale
- 7.- Eje del pistón
- 8.- Rodillo central

El contador de pistón oscilante tiene una cámara de medida a la cual todo el fluido penetra a través del orificio de entrada, pasando alrededor del espacio anular entre los anillos externo e interno y continuando hacia el orificio de descarga. Parte del fluido recorre esta distancia por el espacio existente entre el pistón y el anillo externo y parte circula por el espacio entre la pared del pistón hueco y el anillo interior.

El pistón del medidor va guiado por un eje el cual sigue una trayectoria circular entre el anillo interno y un rodillo central. Una prolongación del eje que atraviesa la cubierta de la cámara de medida, comunica movimiento a una serie de ruedas dentadas que registran sobre una esfera graduada el volumen total del fluido que ha circulado por el contador a partir del momento en que se ajustó a cero.

Las características de un medidor de pistón oscilante típico para la medición del caudal de gasolina son las siguientes:

Capacidad de medición entre 40 y 75 litros por minuto

Diámetro de tubería en carga y descarga  $3/4$  de pulgada y 1 pulgada.

Estos dispositivos vienen integrados con un contador mecánico a base de engranes, sin embargo, para propósitos de este diseño

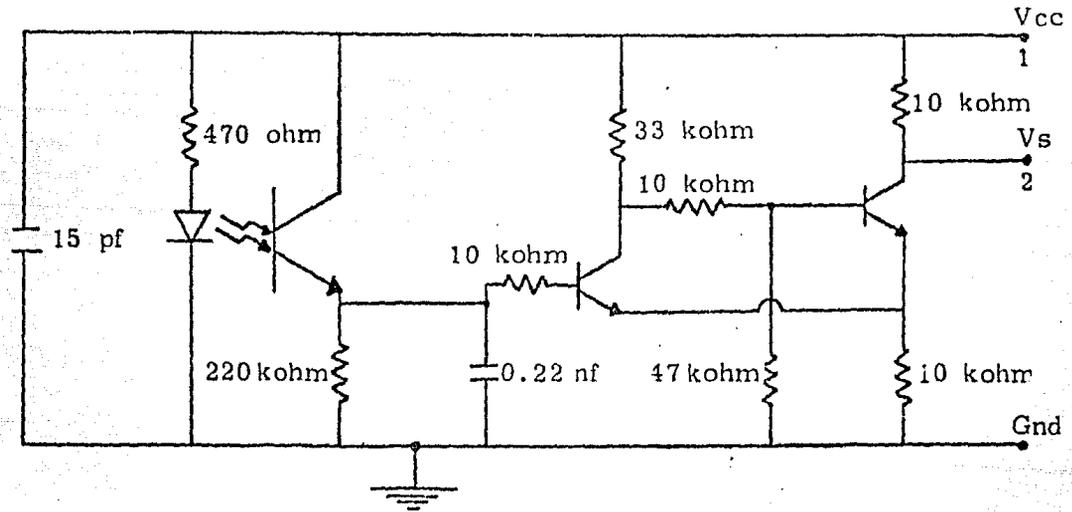


DIAGRAMA DE CIRCUITO DEL DETECTOR

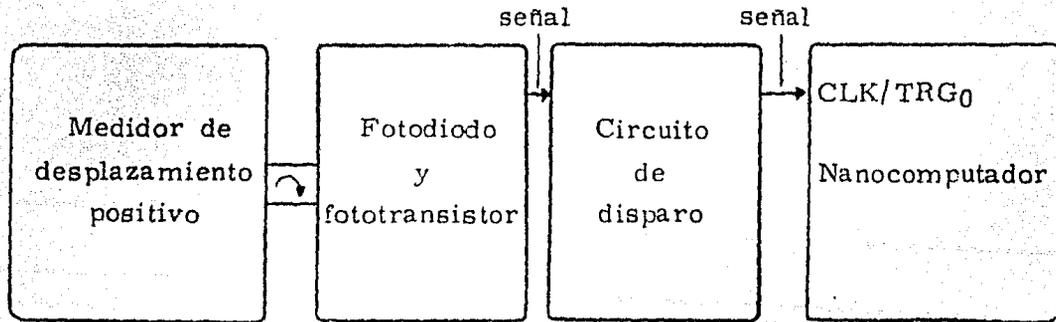


DIAGRAMA A BLOQUES

debe ser sustituido por un disco acoplado al eje de giro; este disco tendrá uno o varios orificios para dejar pasar la luz de un diodo emisor y con esto estimular un fotodetector que mandará pulsos eléctricos al microcomputador.

Este sistema de suministro de gasolina debe ser capaz de adaptarse a diversos medidores de desplazamiento, los cuales pueden tener diferentes capacidades, por lo que la relación vueltas-caudal será un dato implementado por programación, significando que el sistema electrónico y su programa puede ser utilizado con varios medidores de desplazamiento, ya sean de pistón oscilante, disco oscilante o inclusive con bombas rotatorias utilizadas como contadores de caudal.

Si el caudal del fluido aumenta, entonces la velocidad del eje del medidor también aumentará; este incremento o decremento de la velocidad del eje del contador no afectará el funcionamiento del microcomputador ya que puede recibir los pulsos luminosos a una frecuencia mayor o menor de la especificada como típica.

### Acoplador y convertidor

El acoplamiento entre el medidor de caudal y el nanocomputador Z80 Starter Kit se efectúa por medio de un fotosensor y un disco perforado, que permite el paso de la luz desde un fotodiodo hasta un fotodetector en ciertos puntos del disco cuando este gira y pasa a través del orificio el haz de luz.

El eje del medidor de pistón oscilante está directamente acoplado al disco perforado y así cada vez que el eje dé una revolución, el orificio del disco permitirá el paso de luz a través de él.

En este diseño se utiliza un arreglo de fotodiodo y fototransistor integrados en un solo componente. El utilizar un diodo emisor de luz y un fototransistor integrados en un módulo, tiene la ventaja de evitar problemas de acoplamiento óptico entre ambos elementos y permite simplificar el diseño y en su caso el mantenimiento.

Existen tres tipos básicos de arreglos de fuente y sensor que son el: TIL 138, TIL 143 y TIL 145; el arreglo de sensor y emisor que emplearemos será el TIL 138, debido a que su velocidad de respuesta es mayor que la del TIL 143 y no necesitamos la alta ganancia que nos proporciona el fotosensor en configuración Darlington del TIL 145.

El arreglo del emisor y sensor TIL 138 está formado por un diodo infrarrojo de galio arsénico, tipo TIL 32 y un fototransistor del tipo TIL 78 NPN, montados ambos dispositivos en un soporte termoplástico tipo ABS.

Las características básicas se enumeran a continuación: Voltaje inverso de fuente 2V, Corriente continua inversa de fuente 40 mA, Voltaje colector emisor en sensor 50 V, Voltaje emisor colector en sensor 7 V, Disipación en el sensor a 25°C 50mW, Tiempo de subida y bajada 1.5 ms.

Para acoplar el dispositivo detector al nanocomputador Z80 Starter Kit es utilizado un circuito con un disparador Smith, el cual tiene la función de eliminar el ruido generado por el transductor y el fototransistor, para así transmitir a través de la señal CLK/TRG, (que se encuentra junto al área de alambrado externo), un pulso cuadrado al canal cero del circuito contador temporizador Z80 CTC.

## INTERFASE AL MOTOR

La interfase al motor eléctrico tiene por objeto permitir el envío de un dato binario que tendrá la función de activar o desactivar al motor eléctrico que impulsa a la bomba suministradora de gasolina; esta interfase está constituida básicamente por:

- 1.- Tiristor bidireccional
- 2.- Optoacoplador e interfase.

## Tiristor bidireccional

El tiristor bidireccional o TRIAC tiene la función de permitir el flujo de energía al motor eléctrico de acuerdo al estado de su compuerta.

El tiristor bidireccional nos permite el flujo de corriente alterna sin modificaciones, en comparación con el rectificador de silicio SCR que permite el flujo de corriente solamente en un semiciclo de la señal de corriente alterna; el tiristor bidireccional adecuado para este diseño es el TIC 226D; algunas de sus características son: corriente pico de compuerta 1A, voltaje pico de entrada 400V, corriente pico de entrada 8A; ésto significa que el motor eléctrico que impulsa a la bomba no deberá sobrepasar estos límites de corriente y voltaje.

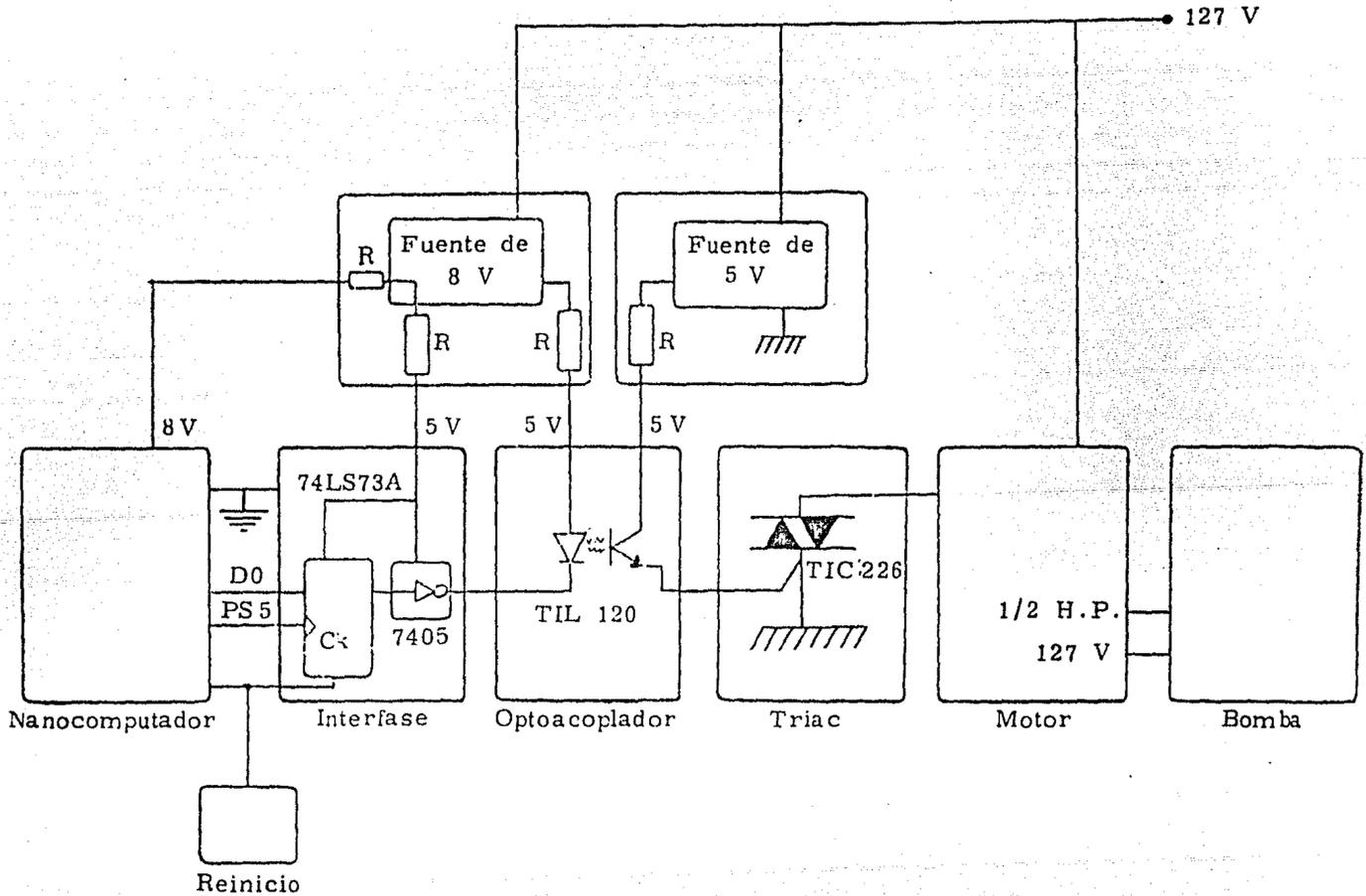
## Optoacoplador

Un dispositivo de este tipo permite aislar físicamente el motor eléctrico del sistema electrónico, evitando así que un voltaje transitorio en el sistema de potencia dañe a los componentes electrónicos que actúan con niveles de voltaje y corriente pequeños; el optoacoplador más adecuado para realizar esta función es el TIL 120, formado por un fotodiodo y un fototransistor; las características básicas de este dispositivo son: voltaje de aislamiento, que va desde +1KV a -1KV, corriente del diodo 40 mA, voltaje inverso en el diodo 3V, voltaje colector emisor en el transistor 35V, voltaje emisor colector 7V y corriente de colector 50 mA.

El fotodiodo es conectado a la salida de un multivibrador biestable tipo D (74LS 73A), utilizando para ello un inversor de colector abierto del tipo 7405; debido a que el multivibrador tiene un voltaje de salida entre 3.4V y 0,4V con una corriente de -400 mA cuando tiene nivel lógico alto, y de 16 mA cuando tiene nivel lógico bajo, valores de corriente que no permitirían operar adecuadamente al fotodiodo que requiere un flujo de corriente de 40 mA.

La forma en que opera la interfase es la siguiente:

El puerto 94H del nanocomputador controla la operación del sistema por medio del bit número cero del puerto; este bit es transferido por la línea de más bajo orden del bus de datos, siendo trans-



INTERFASE AL MOTOR

ferido al biestable D; el biestable es activado por la señal selectora de puerto PS5 (que actúa con nivel lógico bajo); el biestable detecta el cambio de nivel de alto a bajo y transfiere la información de la línea D0 del bus de datos a la salida.

Al transferir un cero lógico, el nivel de voltaje a la salida del biestable es bajo, pero al pasar a través del inversor, se transforma en alto por lo que el fotodiodo no podría encender, ya que la diferencia de voltaje entre la alimentación del fotodiodo y su salida es la misma; al colocar un uno lógico en el biestable, este nivel se transfiere al inversor, apareciendo a la salida de éste, un nivel lógico bajo, encendiendo inmediatamente el fotodiodo debido a la diferencia de potencial.

Al activarse el fotodiodo emite luz, que es detectada por el fototransistor, poniéndose en estado de conducción y activando la compuerta del tiristor; al desactivarse el fotodiodo, se desactiva el fototransistor, llevando un nivel lógico bajo a la compuerta del tiristor y causando su desactivación en el primer cruce por cero de la señal de voltaje alterno.

## ALIMENTACION DEL EQUIPO

La alimentación del sistema debe incluir dos fuentes independientes:

1. - Alimentación del optoacoplador
2. - Alimentación del nanocomputador,  
Sistema de despliegue y transductor

### Alimentación del optoacoplador

La alimentación del optoacoplador incluye solamente al fototransistor ya que es la parte acoplada al sistema de potencia; para este fototransistor requerimos una fuente de voltaje de 5V de directa. Esta fuente de alimentación de 5V se obtiene de la señal de línea de 120V de alterna; a través de un transformador, con una en-

trada en el primario de 120V de alterna y una salida en el secundario de 12V de alterna, y con rectificadores de onda completa del tipo puente; el voltaje de 5V de directa se fijará por medio del circuito integrado 7805 o algún sustituto.

Las ventajas que se obtienen al utilizar un circuito integrado en la construcción de las fuentes reguladas de voltaje son:

- 1.- Se obtiene una regulación eficaz.
- 2.- Se emplea un número pequeño de componentes.
- 3.- Proporciona corrientes altas.

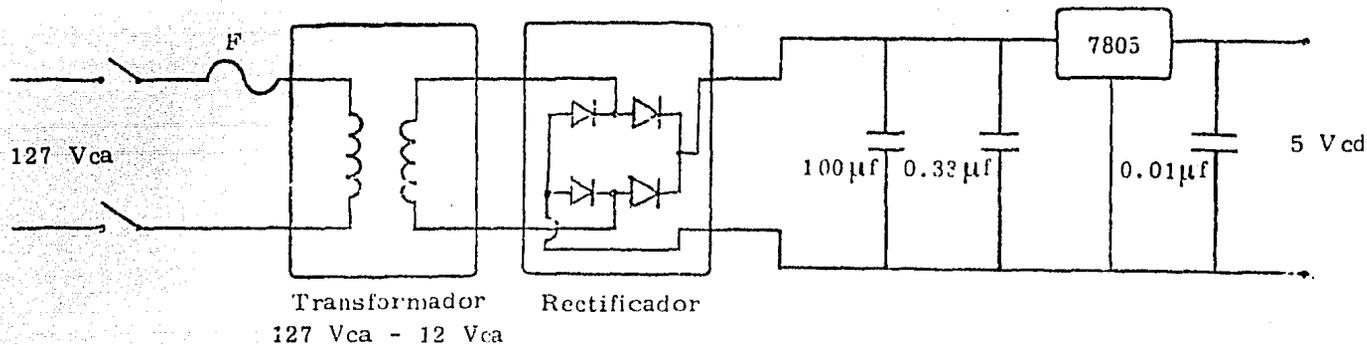
Para poder utilizar el circuito integrado 7805 el voltaje de alimentación al circuito integrado debe ser al menos 2.5V de directa mayor que el valor absoluto del voltaje nominal de salida, además de que la señal debe ser filtrada previamente.

La fuente de alimentación consta de un transformador con una salida en el secundario de 12V de alterna y 1A. En el devanado secundario se implementa un circuito rectificador de onda completa con grupos de 4 diodos o de una cuarteta integrada, obteniendo a la salida una señal de corriente directa pulsante, la cual es filtrada a través de un capacitor C1. Después se conecta el circuito integrado 7805 o algún sustituto junto con un par de capacitores, uno a la entrada de  $0.33\mu\text{F}$  y otro a la salida de  $0.01\mu\text{F}$  los cuales nos servirán para estabilizar la señal del circuito integrado regulador de voltaje.

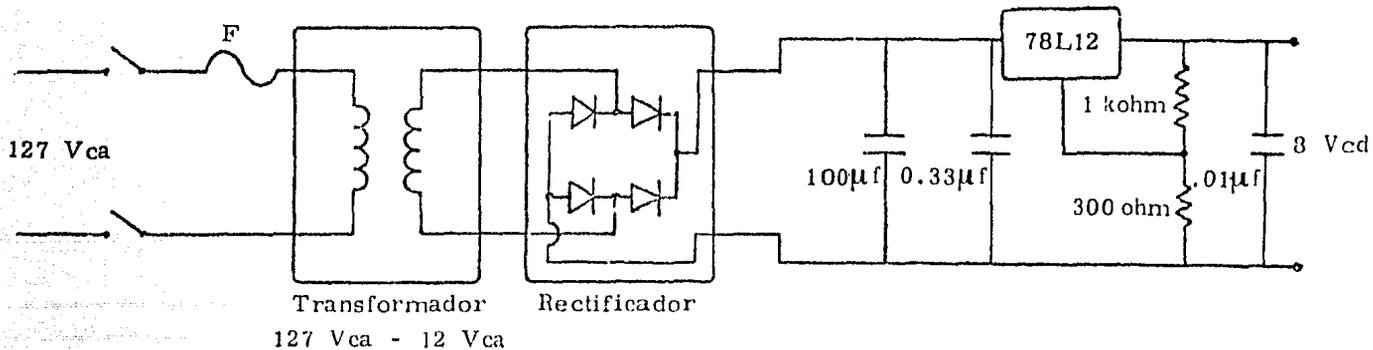
### Alimentación del nanocomputador

La alimentación de 8V, requerida por el nanocomputador, el transductor y el sistema de despliegue, puede ser una sola fuente de 8V de directa o bien el transductor puede ser alimentado por una tercera fuente (independiente del sistema del nanocomputador).

El circuito implementado para generar un voltaje de 8V de directa se basa en el circuito LM 78L12 en lugar del LM 7805, siendo el transformador y el rectificador del mismo tipo que los empleados en la primera fuente; este circuito lleva también un capacitor de entrada de  $0.33\mu\text{F}$  y uno de salida de  $0.01\mu\text{F}$ . Sin embargo es necesario conectar un par de resistencias, una de 1 kohm



ALIMENTACION DEL OPTOACOPLADOR



ALIMENTACION DEL NANOCOMPUTADOR

y otra de 330 ohm entre la salida, la terminal de referencia y tierra, para así estabilizar el voltaje regulado de salida a ocho volts, ya que el circuito regulador en su configuración básica proporciona 12V de directa.

### Protección contra fallas de suministro

Para un mejor funcionamiento del sistema contra interrupciones en el suministro de energía de corriente alterna comercial, es recomendable utilizar un sistema de soporte de energía de C.A. que cuenta con las siguientes características:

- 1.- Se encuentra diseñado para operar conectado siempre entre el equipo y la red comercial.
- 2.- Cuando la tensión de la línea comercial está presente, el sistema de soporte de energía mantiene un banco de baterías en condiciones óptimas de carga y conecta el equipo directamente a la red comercial.
- 3.- Cuando la tensión de la línea comercial baja de un nivel preestablecido o falla por completo, automáticamente el sistema de soporte suministra energía al equipo, para que continúe operando sin interrupción.

El sistema de soporte de energía de C.A. tipo ininterrumpible, cuenta con capacidades de 500 VA, 1500 VA, 2500 VA, 5000 VA, 7500 VA.

El sistema está compuesto por los siguientes elementos constitutivos:

Rectificador - cargador, Inversor, Interruptor de transferencia, Banco de baterías y estante, y cuenta con las siguientes características:

Voltaje de entrada de 127V, Voltaje de salida de 127V, ambos con una tolerancia de cinco por ciento, y una frecuencia de 60 Hz.

## CONCLUSIONES

---

Con la aparición de los semiconductores se genera un vertiginoso avance en la Electrónica y a partir de entonces se han desarrollado dispositivos más y más complejos, los cuales a su vez se han incorporado a nuevos sistemas, que han sustituido ventajosamente a equipos ya existentes. Este avance se ha dado ya en los países industrializados y además, la introducción de nuevos equipos ha sido cosa muy rápida, así que diseñar un equipo nuevo es una labor poco práctica desde el punto de vista económico, ya que existe al menos un equipo similar diseñado en el extranjero, llevando éste en muchos casos a abandonar el proyecto de diseño. Sin embargo la solución de incorporar equipos importados no sólo provoca trastornos económicos, sino que acentúa la brecha tecnológica con el exterior y lleva a depender más y más de éste.

Es por ello necesario realizar diseños propios aún cuando el costo del diseño sea elevado.

El último gran avance en la tecnología de los semiconductores fue la introducción del microprocesador, dispositivo que permite realizar una gran cantidad de funciones que previamente debían ser implementadas con una cantidad considerable de circuitos integrados. El microprocesador transforma el problema de diseñar la circuitería en un problema de diseño del programa, lo cual no implica un costo de material o espacio, sino un costo de desarrollo.

El costo del desarrollo de un programa puede ser minimizado mediante la adecuada estructuración del mismo, lo cual permite reducir errores de programación y de lógica, además de hacerlo fácilmente comprensible a otro programador. También permite modificar alguna función sin alterar el programa en su totalidad, por ejemplo, en el actual diseño con un pequeño cambio podríamos lograr que el sistema manejara dos tipos distintos de combustible, cada uno con su costo respectivo. Sería posible también aumentar la capacidad de

manipulación aritmética del sistema, implementando programas que manejasen números mayores a 65,535 que es la capacidad máxima del sistema, con sólo sustituir los programas aritméticos y sin cambiar en nada la estructuración del programa.

Este diseño utiliza el código de máquina del microprocesador Z80 CPU; la principal ventaja que obtenemos con ésto, radica en el uso de la memoria del sistema de la forma más eficaz y rápida posible, en comparación al empleo de lenguajes de alto nivel como Fortran, Basic, etc.. En este caso se emplea para todo el programa de operación aproximadamente un kilobyte de memoria, pudiéndose cargar en una sola pastilla de memoria tipo EPROM.

Para concluir, podemos decir que el sistema que se diseñó no sólo se puede emplear en un equipo de suministro de gasolina, sino que en una forma más general se puede aplicar al control del flujo de flúidos, área ésta de sumo interés para la Ingeniería.

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

---

- 1.- Barden, William: "The Z80 Microcomputer Handbook".  
Editorial Howard W. Sams.
- 2.- Berenguer, Xavier: "Los Ordenadores". Editorial Salvat.
- 3.- Brown, George G.: "Ingeniería Química". Editorial Marín.
- 4.- Doebelin, Ernest: "Measurement Systems". Editorial Mc.  
Graw Hill.
- 5.- Facultad de Ingeniería, UNAM, División de Educación  
continua: "Introducción a los microprocesadores".
- 6.- García, Octavio: "Microprocesadores Z80 e interfases".  
Editado por ESIME - IPN.
- 7.- Holzbock, W.G.: "Instrumentos para medición y control".  
Editorial CECSA.
- 8.- Libes, Sol y Garetz, Mark: "Interfacing to S-100/IEEE  
696 microcomputers". Editorial Osborne/Mc. Graw Hill.
- 9.- Mataix, Claudio: "Mecánica de fluidos y Máquinas hi-  
dráulicas". Editorial Harper & Row Publishers Inc.
- 10.- Moralejo, M. y Pascual, E.: "La Electrónica".  
Editorial Salvat.

- 11.- Peatman, John B.: "Microcomputer Based Design". Editorial Mc. Graw Hill.
- 12.- Portilla Marcial: "Administración de Proyectos de Software". División de Educación continua, Facultad de Ingeniería, U.N.A.M.
- 13.- Rodney, Zarks: "From Chips to Systems". Editorial Sybex.
- 14.- S.D. Systems: "Operations Manual Z80 Starter Kit". Editado por S.D. Systems.
- 15.- Siskind, Charles: "Electrical Machines". Editorial Mc. Graw Hill.

## ANEXO

## INSTRUCTIVO DE OPERACION

El sistema de suministro de gasolina externamente está compuesto por los elementos que a continuación se indican:

- 1.- Tablero de Control.
- 2.- Almacén para la bomba, medidor, y motor.
- 3.- Manguera de suministro.

El Tablero de Control permite al usuario emplear el equipo de suministro de gasolina. Está constituido por los siguientes elementos:

- 1.- Panel de Comandos.
- 2.- Panel de Programación.
- 3.- Exhibidores luminosos.

### PANEL DE COMANDOS

El Panel de Comandos consta de varias teclas, colocadas en la parte inferior del tablero de control y son las siguientes:

- 1.- A. Iniciar proceso.
- 2.- B. Programación de caudal.
- 3.- C. Programación de costo.
- 4.- D. Activación del sistema.
- 5.- E. Reactivación del sistema.
- 6.- F. Verificación de datos o corte de suministro.

### Comando A (Iniciar Proceso)

Con este comando se prepara al sistema para ser utilizado por el usuario, realizando varias acciones que son; colocar en "cero" los exhibidores luminosos, borrar los registros de datos programados y resultados obtenidos por el usuario anterior, y finalmente habilitar a los demás comandos para ser utilizados por el nuevo usuario.

### Comando B (Programar caudal)

Con este comando se indica al sistema que se desea programar cierta cantidad de litros; al ser accionado despliega el mensaje "LLLLL" "-----" en los exhibidores luminosos para indicarle al usuario que ha reconocido el comando, esperando a continuación la introducción del número de litros (cuatro dígitos) registrando estos datos y mostrando el costo de tal suministro.

### Comando C (Programar costo)

Con este comando se indica al sistema de suministro que se desea obtener una cantidad de gasolina equivalente a una determinada cantidad de dinero. Al ser accionado despliega el mensaje "-----" "PPPPP" en los exhibidores luminosos para indicarle al usuario que ha reconocido el comando, esperando a continuación la introducción del número de pesos que desee emplear el usuario y mostrando los litros que serán suministrados.

### Comando D (Activación del sistema)

Con este comando se colocan nuevamente en cero los exhibidores luminosos y se activa el motor eléctrico que maneja a la bomba de suministro de gasolina, se registran los datos programados (si es que los hay) y monitorea el flujo de gasolina hasta llegar a la condición impuesta por el usuario o hasta que se le indique, desactivando a continuación el motor eléctrico.

### Comando E (Reactivación del sistema)

Con este comando se permite al usuario la activación del sistema de suministro de gasolina después de haber sido desactivado, sin que se pierdan los resultados que se tenían hasta ese momento, quedando inhabilitados los comandos para programar litros o costo.

## Comando F (Verificación de datos o corte de suministro)

Cuando el sistema no está suministrando gasolina este comando permite ver la cantidad y el costo de la gasolina que se ha programado; además muestra el contenido del registro de resultados. Cuando el sistema está suministrando gasolina este comando nos permite finalizar la operación.

## PANEL DE PROGRAMACION

El panel de programación está constituido por un teclado con diez botones que corresponden a los números de cero al nueve, con los cuales el usuario ordena la cantidad de litros o de dinero que desea, debiendo introducir siempre un número entero de cuatro cifras (si es un número pequeño deberá introducir ceros a la derecha para completar las cuatro cifras).

## EXHIBIDORES LUMINOSOS

Dos grupos de exhibidores luminosos permiten ver la cantidad de litros y pesos suministrados al usuario o programados por él.

Cada grupo de exhibidores luminosos consta de cinco dígitos. El exhibidor superior muestra la cantidad de litros, mientras que el inferior muestra la cantidad de pesos.

Los exhibidores luminosos permiten también visualizar diversos mensajes como son: "LLLLL", "-----" para programar litros; "-----", "PPPPP" para programar costo; "EEEE", "EEEE" para indicar error de secuencia "-----", "-----" para indicar error de comando (si en lugar de introducir un comando se introduce un número), etc.

## MODOS DE OPERACION

El sistema de suministro de gasolina puede ser operado de dos formas diferentes que son:

- 1.- Modo directo.
- 2.- Modo programado.

### Modo directo

Al ser operado de esta forma, el sistema de suministro actúa del mismo modo en que operan los actuales sistemas de suministro, esto es, se activa el sistema y se controla el flujo de gasolina manualmente, o bien, se espera a que el tanque de gasolina se llene.

Las acciones que se ejecutan para emplear este modo de operación son:

- 1.- Se retira la manguera del soporte.
- 2.- Se inicializa el sistema oprimiendo el comando "A" con lo cual aparecen ceros en los exhibidores luminosos.
- 3.- Se activa el sistema oprimiendo el comando "D".
- 4.- Se puede cortar el suministro en cualquier momento con el comando "F".
- 5.- Si se desea reactivar una vez más el sistema se oprime el comando "E".
- 6.- Al finalizar se repone la manguera en el soporte.

### Modo Programado

Al ser operado en esta forma, el sistema es programado con un valor de caudal o un costo, dejando de operar precisamente cuando dicho valor programado es alcanzado.

Se puede programar el sistema en dos formas diferentes; una introduciendo el caudal que debe ser suministrado, y otra introduciendo los datos del consumo que se desea realizar.

### Programación de Caudal

Las acciones que se ejecutan para utilizar el modo programado con datos de caudal (litros a suministrar) son:

- 1.- Se retira la manguera del soporte.
- 2.- Se inicializa el sistema oprimiendo el comando "A" con lo cual aparecen ceros en los exhibidores luminosos.
- 3.- Se introduce el comando "B"; el sistema despliega el mensaje "LLLLL", "-----" que indica listo para recibir datos.

Si el sistema no fué inicializado oprimiendo el comando "A" aparece el mensaje de error "EEEEEE", "EEEEEE".

- 4.- Se introduce el número de litros a consumir; es necesaria la introducción de cuatro cifras por lo que en caso necesario deberán introducirse ceros a la derecha para completar cuatro dígitos.

Si se introduce un carácter erróneo el sistema responde "?????", "?????" esperando la introducción correcta.

- 5.- Esperar a que aparezca el costo equivalente a los litros programados; ésto sucede inmediatamente después de la introducción de la cuarta cifra del punto anterior. Si el usuario no está de acuerdo con el costo puede reprogramar el número de litros regresando al punto 3 de la secuencia, o bien si el sistema responde ")))))", ")))))" indicando que el costo excede la capacidad del sistema, también se debe regresar al punto 3.

- 6.- Se activa el sistema oprimiendo el comando "D" (aparecen nuevamente ceros en los exhibidores luminosos).
- 7.- Se puede cortar el suministro en cualquier momento con el comando "F".
- 8.- Si se desea reactivar una vez más el sistema, se oprime el comando "E"; sin embargo la programación realizada por el usuario es eliminada.
- 9.- Al finalizar se repone la manguera en el soporte.

### Programación de Costo.

Las acciones que se ejecutan para utilizar el modo programado con datos de costo (dinero a pagar) son:

- 1.- Se retira la manguera del soporte.
- 2.- Se inicializa el sistema oprimiendo el comando "A" con lo cual aparecen ceros en los exhibidores luminosos.
- 3.- Se introduce el comando "C"; el sistema despliega el mensaje "-----", "PPPPP" que indica listo para recibir datos.

Si el sistema no fue inicializado oprimiendo el comando "A" aparece el mensaje de error "EEEE", "EEEE".

- 4.- Se introduce el número de pesos a pagar; es necesaria la introducción de cuatro cifras, por lo que en caso necesario deberán introducirse ceros a la derecha para completar cuatro dígitos.

Si se introduce un carácter erróneo el sistema responde "?????", "?????" esperando la introducción correcta.

5.- Esperar a que aparezcan los litros equivalentes al costo programado; ésto sucede inmediatamente después de la introducción de la cuarta cifra del punto anterior.

Si el usuario no está de acuerdo con la cantidad de litros equivalentes puede reprogramar el número de litros regresando al punto 3 de la secuencia.

6.- Se activa el sistema oprimiendo el comando "D" (aparecen ceros en los exhibidores luminosos).

7.- Se puede cortar el suministro en cualquier momento con el comando "F".

8.- Si se desea reactivar una vez más el sistema, se oprime el comando "E"; sin embargo la programación realizada por el usuario es eliminada.

9.- Al finalizar se repone la manguera en el soporte.

## VERIFICACION DE REGISTROS

Cuando por alguna razón se pierde la visualización del total consumido o la programación realizada (por ejemplo si se introduce el comando programación de Litros al finalizar el suministro de gasolina), basta con oprimir el comando de verificación de registros para que sea mostrada en primer lugar la cantidad programada por el usuario y a continuación restablece los resultados del suministro de gasolina y el costo de dicho suministro.